International Journal on Applications in Engineering and Technology
Volume 4: Issue 2: May 2018, pp 65 – 70 www.aetsjournal.com ISSN (Online) : 2455 - 0523

UNDERSTANDING SMARTPHONE SENSOR AND APP DATA FOR ENHANCING THE SECURITY OF SECRET QUESTIONS

D.ANBUMOZHI , M.VIJAYANANTHKUMAR , R.VETRIVENTHAN, SENTHIL KUMARAN

Abstract— Many web applications provide secondary authen- tication methods, i.e., secret questions (or password recovery questions), to reset the account password when a user's login fails. However, the answers to many such secret questions can be easily guessed by an acquaintance or exposed to a stranger that has access to public online tools (e.g., online social networks); moreover, a user may forget her/his answers long after creating the secret questions. Today's prevalence of smartphones has granted us new opportunities to observe and understand how the personal data collected by smartphone sensors and apps can help create personalized secret questions without violating the users' privacy concerns. In this paper, we present a Secret- Question based Authentication system, called "Secret-QA", that creates a set of secret questions on basic of people's smartphone usage. We develop a prototype on Android smartphones, and evaluate the security of the secret questions by asking the acquaintance/stranger who participate in our user study to guess the answers with and without the help of online tools; meanwhile, we observe the questions' reliability by asking participants to answer their own questions. Our experimental results reveal that the secret questions related to motion sensors, calendar, app installment, and part of legacy app usage history (e.g., phone calls) have the best memorability for users as well as the highest robustness to attacks.

Keywords— smartphone , Secret-QA system

I. INTRODUCTION

Secret questions (a.k.a password recovery questions) have been widely used by many web applications as the secondary authentication method for resetting the account password when the primary credential is lost [1]. When creating an online account, a user may be required to choose a secret question from a pre-determined list provided by the server, and set answers accordingly. The user can reset his account password by providing the correct answers to the secret questions later. For the ease of setting and memorizing the answers, most secret questions are blank-fillings (a.k.a. fill-in-the-blank, or short-answer questions), and are created based on the long- term knowledge of a user's personal history that may not change over months/years (e.g., "What's the model of your first car?"). However, existing research has revealed that such blank-filling questions created upon the user's long-term history may lead to poor security and reliability.

The "reliability" of a secret question is its memorability the required effort or difficulty of memorizing the correct answer. Without a careful choice of a blank-filling secret question, a user may be declined to log in, because he cannot remember the exact answer that he provided, or he may misspell the input that requires the perfect literally-matching to the correct answer.

The recent prevalence of smartphone has provided a rich source of the user's personal data related to the knowledge of his short-term history, i.e., the data collected by the smart-phone sensors and apps. Is it feasible to use the knowledge of one's short-term personal history (typically within one month) for creating his secret question?

- Intuitively, the short-term personal history is less likely to be exposed to a stranger or acquaintance, because the rapid variations of an event that a person has experienced within a short term will increase the resilience to guess attacks. This implies improved security for such secret questions.
- Moreover, research findings in psychology show that one can easily memorize the details of his short-term activity, if this activity occurs multiple times during a short-term (e.g., calling a friend many times), and/or this activity heavily involves his time and effort in a short time period (e.g., running exercise).

In this paper, we present a Secret-Question based Authentication system, called "Secret-QA", taking advantage of the data of smartphone sensors and apps without violating the user privacy. Meanwhile, we develop a prototype of Secret-QA, and conduct an experimental user study involving 88 volunteers to evaluate the reliability and security of the set of secret question created in the system. Specifically,

- We design a user authentication system with a set of secret questions created based on the data of users' shortterm smartphone usage.
- We evaluated the reliability and security of the three types of secret questions (blank-filling, true/false, and multiple-

D.Anbumozhi , Master of Computer Applications , Meenaakshi Ramasamy Engineering College , Thathanur , Ariyalur Dt , Tamil Nadu .

M.Vijayananthkumar , Assistant professor/MCA , Meenaakshi Ramasamy Engineering College , Thathanur , Ariyalur Dt , Tamil Nadu .

Mr.R.Vetriventhan , BE , MTech , MISTE , Head of the department , Department of MCA , Meenaakshi Ramasamy Engineering College , Thathanur , Ariyalur Dt , Tamil Nadu .

Mr.Senthil Kumaran , ME Phd , Managing Director , Meenaakshi Ramasamy Engineering College , Thathanur , Ariyalur Dt , Tamil Nadu.

choice) with a comprehensive experiment involving 88 participants.

- The experimental results show that the combination of multiple lightweight true-false and multiple choice questions required less input effort with the same strength provided by blank-filling questions.
- We evaluate the usability of the system, and find that the Secret-QA system is easier to use than those existing authentication system with secret questions based on users' long-term historic data.

The rest of this paper is organized as follows: we provide background knowledge in Section II. In Sections III, we give an overview of the system design. We present our approach of creating secret questions in Section IV. In Sections V and VI, we evaluate the system performance over all created secret questions. We conclude the paper in Section VII.

II. BACKGROUND AND RELATED WORK

The blank-filling secret questions are dominant as the main stream authentication solution, especially in web and email authentication systems [1], despite the criticism on its security and reliability.

Guessing attacks by acquaintance and stranger. The security of secret questions for authentication was studied by Zviran and Haga in 1990 , which indicated that the answers of 33% questions can be guessed by the "significant others" who were mainly participants' spouses (77%) and close friends (17%). Another similar study was conducted by Podd et al, which revealed a higher rate of successful guessing (39.5%). A recent study showed that even an open question written by the user himself was still vulnerable to the guessing attacks launched by his acquaintance.

On the other hand, strangers can be more sophisticated than ever to launch the guessing attacks, as they can access the user's personal history through online social networks (OSN) or other public online tools. Therefore, the statistical guessing has become an effective way to compromise a few personal "secret" questions(e.g., "Where were you born?", "What is the name of your high school?").

Poor reliability of secret questions in real world. Regarding the reliability, a secret question should be memory-wise effortless for users . However, today's mainstream secret question methods fail to meet this requirement. A recent study revealed that nearly 20% users of four famous webmail providers forgot their answers within six months. Moreover, dominant blank-filling secret questions with case sensitive answers require the perfect literally matching to the set answer, which also contributes to its poor reliability.

Recent proposals of user authentication systems. To reduce the vulnerability to guessing attacks, Babic et al tried using short-term information such as a user's dynamic Internet activities for creating his secret questions, namely network activities (e.g., browsing history), physical events (e.g., planned meetings, calendar items), and conceptual opinions (e.g., opinions derived from browsing, emails) . They emphasized that frequently changing secret questions will be difficult for attackers to guess the answers. However, this research is based on the data related to a user's Internet activities, while our work leverages the mobile phone sensor and app data that can record a user's physical world activities, for creating secret questions.

For better reliability, one may choose other types of secret questions rather than blank-filling questions to avoid the difficulty in recalling and inputting the perfect literally matching answer. For example, the login to an online social network requires a user to recognize one of his friends in a photo. However, it is feasible that a user fails to recognize if he is not familiar to that particular friend chosen by the authentication server.

Such existing proposals serve as a good start of using one's short-term activities to create secret questions as well as trying other question types. Since the smartphone has become one's most inseparable device of recording his life, this paper presents a user authentication system Secret-QA to study on how one's short-term history almost all types of one's activities sensible to the smartphone can benefit the security and reliability of secret questions. Meanwhile, we evaluate the attack robustness of using a combination of many lightweight questions (true/false, multiple-choice) instead of using the blank-fillings, in order to strike a balanced tradeoff between security(and/or reliability) and usability.

Figure 1. System architecture of Secret-QA, for a typical user scenario of resetting the account password through answering the secret questions.

III. SYSTEM OVERVIEW

The Secret-QA system consists of two major components, namely the user event extraction scheme and the challenge response protocol, which is shown in Figure 1 and will be elaborated next.

A. The User - event Extraction Scheme

Today's smartphones are typically equipped with a plethora of sensors and apps which can capture various events related

to a user's daily activities, e.g., the accelerometer can record the user's sports/motion status without consuming excessive battery .

Selection of sensors/apps. In the user-event extraction scheme, Secret-QA selects a lists of sensors and apps for extracting the user activities, including: (1) the common sensors equipped on the top ten best selling smartphones in 2013, (2) the top-ten downloaded Android apps in 2013, and (3) the legacy apps (Call, Contact, SMS, etc.), as shown in Table I. Because these sensors and apps are already built in for almost all the smartphones, our approach is naturally suitable for smartphone users without introducing any extra hardware costs.

Secret-QA client app. Given the designated sensors and apps for building the authentication system, we develop a Secret-QA client app called "EventLog" to extract the features for question generation. As shown in the block diagram (the step 0 in Figure 1), the client app schedules the feature extraction process periodically, and then features will be recorded in the local databases. For example, we adopt libSVM [15] on Android to detect motion related user events, and we roughly set the minimum duration to 10 minutes for noise removal (details on how to create questions and algorithms for other types of events extraction will be given in Section IV). Note that our extraction of user events are most lazily scheduled using Android Listener [16] to save battery; meanwhile, we will pause the scheduling for some sensors after the screen is locked (e.g., app usage), because no events can happen during screen lock periods.

Secret-QA server. A trusted server is used as the auditor, which can also provide the user authentication service even if the phone is not available. As shown in block diagram of Figure 1, when authentication is needed, users' phone can generate questions with local sanitized data and send the answers/results (e.g., how many questions they answered correctly) to auditors via HTTPS channels.

TABLE -1 TOP TEN CATEGORIES OF SENSORS/APPS SELECTED IN SECRET-QA.

1) GPS	2) Acc. (Accelerometer)
3) Calendar	4) Battery charging
5) Photo-taking	6) Contact
7) App installment	8) Call
9) SMS	10) App usage (mainly OSN apps)

B. A Three-phase Challenge Response Protocol

As shown in Figure 1 (from step 1 – 5), a service provider needs to authenticate the user's identity (typically for resetting the account password) through our trusted server. The service prescribes three phases for authentication.

- Issue: the user issues an authentication request to the service provider (e.g., an OSN website, the step 1 in Figure 1), then the OSN website asks our trusted server for one or more encrypted secret questions and its answers; the questions are finally transferred to the user displaying on the smartphones (the step 2 – 3 in Figure 1). The information at this phase must be sent over a secure channel [17] against the malicious eavesdroppers.

Challenge: the user provides answers to the challenge questions according to his/her short term memory, then sends it back to the OSN website (the step 4 in Figure 1).

- Authentication: the authentication is successful if the user's response conforms to the correct answers; otherwise, a potential attack is detected. If the times of authentication failure exceeds the threshold, our trusted server would deny to provide service for this particular user, as the in the last step in Figure 1.

Note that the interactions with server is also necessary to improve the resilience to some obvious attack vectors in local operation mode. For instance, if a user's mobile phone is stolen/lost (or the user has been followed by a stranger for days), the user can disable EvenLog functionality (or remote lock/swipe out the phone) to eliminate the danger of potential adversary who records the users' recent activities with the help of server.

C. Threat Models

Former studies including [2]–[4] focused on attacks launched by users' significant others or acquaintances, but they ignored malicious guessing attacks from strangers. Moreover, sophisticated attackers could take advantage of online tools to increase their guess rate [5]. Thus, we consider threat models of the two above crossed factors (acquaintance vs. stranger; with vs. without online tools or external help): (1) acquaintance attacks using online tools, (2) acquaintance attacks without external help, (3) stranger attacks using online tools, (4) stranger attacks without external help.

IV. DESIGN OF CHALLENGE-RESPONSE PROTOCOL

We create three types of secret questions: A "True/false" question is also called a "Yes/No" question because it usually expects a binary answer of "Yes" or "No"; a "multiple-choice" question or a "blank-filling" question that typically starts by a letter of "W", e.g., Who/Which/When/What (and thus we call these two types of questions as "W" questions).

We have two ways of creating questions in either a "Yes/No" or a "W" format: (1) a frequency-based question like "Is someone (Who is) your most-frequent contact in last week?"; and (2) a non-frequency based one like "Did you (Who did you) call (Someone) last week?", shown in Tables II and III. Note that the secret questions created in our system are example questions that we have for studying the benefits of using smartphone sensor/app data to improve the security and reliability of secret questions. Researchers are free to create more secret questions with new question formats or by using new sensor/app data, which leads to more flexibility in the design of a secondary authentication mechanism.

A. True/false Questions

Location (GPS) related questions. The example question related to GPS is No. 1 "Did you leave campus yesterday?". The GPS sensor captures the location information of the

participants [18], [19] so that we could easily learn whether participants left campus far away enough with GPS coordinates recorded.

Since that the coarse grained GPS data has a typical mean error of 500 meters as described in Android API reference [20], and thus we determine a participant leaves the campus when the GPS location is 500 meters out of the campus area.

Motion activity (accelerometer) related questions. The example question related to accelerometer is No. 2 "Did you do running exercise for at least 10min with your phone carried yesterday?". There are many smartphone applications that help users to monitor their running activities. We can tell whether the participant is involved in running exercise using the accelerometer data, and in order to remove noise, we roughly set the minimum duration of detecting a user's involvement in running to be 10 minutes.

Smartphone usage (calendar, battery and camera) related questions. The questions derived from the calendar events is No. 3 "Is there an item planned for next week in your calendar?". As requested by participants, we only recorded whether there would be an item planned in next few days in the calendar; we did not access the content of any planned item in the calendar as it is a severe invasion of privacy.

We use the similar format to generate true/false questions related to battery charging and camera usage using Android API: "Did you do something with battery/camera in the past one or few days?" (Question No. 4 and 5 in Table II).

Questions on legacy app usage: contact, call, SMS. We generate true/false questions related to contact, call, SMS in a similar way. For example, No. 7 question is: "Is someone in your contacts on the phone?". True/false questions can be generated based on call and SMS history using the similar format: "Did you call/text someone?". Similar to other true/false questions, the correct answer to this question is randomly set as true or false with an equal probability.

- If the correct answer is set as "true", we randomly pick a name in the phone's contact, and replace "someone" in the question with this chosen name literally.
- Otherwise if the correct answer is set as "false", we create a fake name to replace "someone" in the question by the approach proposed by Luo et al [22]. This approach randomly picks a first name and a last name in phone's contact list, without colliding with an existing name in the list.

Questions on third-party app installment and usage. We obtain a list of third-party apps via Android API, and we also monitor the usage of these apps. We filter out "launcher" apps and EventLog itself in our monitoring experiment. "Launcher" apps are the default home screen applications on Android, e.g.,"Samsung Desktop". As the study [23] indicates, "launcher" apps are the most frequently called ones on Android systems, while users may not be aware of their unintentional usage of it. After that, we can generate a true/false question like the legacy app: "Did you install/use some app on your phone (in the past few days)?".

B. Multiple-choice and Blank-filling Questions

We create "W" questions in the form of multiple-choice and blank-filling by simply extending the true/false questions on legacy and third-party apps. For example, a true/false questions can be easily extended to be a "W" question: "who did you call/text?" (incoming and outgoing calls/SMS were treated equally), or a frequency-based "W" question: "Which app did you use most frequently?".

Answers to multiple-choice questions. For each multiple choice question, there are four options (only one correct option). The correct option is randomly picked with an equal probability of being any options. For example, as for Question No. 28 "Who did you call last week?" , we randomly pick a name in participant's last week call records, and the rest three are faked by names in the contact (meanwhile not appearing in the call records), then we randomly shuffle these names to be the options of the question.

We count the number of calls (or SMS) from/to every contact, or the number of times an app is used by a participant, for creating the frequency based question, e.g., No. 34 "Who was your most frequent contact last week?". If there are more than one most frequent contacts or most frequently used apps, any answer within these candidates is considered correct.

Answers to blank-filling questions. For each blank-filling question, we have a default correct answer that is set by our system, as well as an answer input by the participant in the memory test. We use the following method to determine whether an input answer matches the default correct one. First, we can easily filter out futile answers, and then we borrow the approach proposed by Stuart Schechter et al [4] to compare the input and default answers, i.e., to remove all non-alphanumeric characters, force letters into lower cases, and allow one error (an improved version of edit distance cost) for every five characters in the default answer.

C. Definition and Thresholds of Determining A Good Question

A good secret question is defined as easy-to-remember and hard-to-guess, i.e., the majority of participants in the memory test could correctly recall the answer, and attackers could not significantly increase their chance more than a random guess. We set the threshold of easy-to-remember questions to be

80% for both true/false and multiple-choice questions i.e., 80% participants to correctly answer the question, according to the threshold used for traditional webmail secret questions [4]. A random guessing attack has a success rate of 50% and 25% for true/false and multiple-choice (one of four options) questions, respectively. Then, we set the threshold of hard-to- guess questions to be no more than 55% (or 30%)—i.e., less than 55% (or 30%) attackers can correctly guess the answer, which is approximately to be a random guess for true/false (or multiple-choice) questions.

V. EVALUATION AND EXPERIMENTS RESULTS

A. Experiment Setup

The reliability and security of our system mainly relied on the secret questions that Secret-QA created, so we carried out a user study to evaluate the performance of our system. Note that in the future work, we will consider establishing a probabilistic model based on a large scale of user data to characterize joined our Phase 3 experiment to provide us feedbacks. The detailed feedbacks are presented in Section V-G, VI.

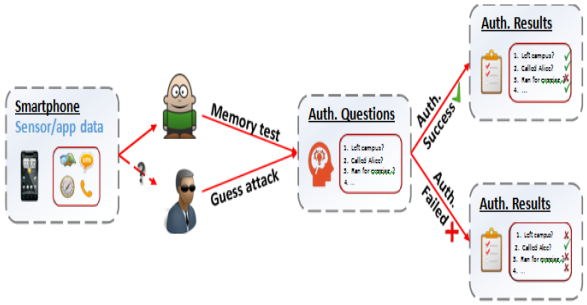

Figure 2. Three-phase experiment procedure .

B. Overall Experiment Results and Definition

In total, we create 525 true/false and 404 multiple-choice questions (select one out of four) and collect 10,558 answers; meanwhile we have 162 blank-filling questions and collect 1,783 answers from all participants. Our results show that the secret questions related to motion sensors, calendar, app installment, and one question related to call have the best performance, most of which have a high reliability over 90%, while the success rate of guessing attacks is as low as that of a random guess.

Experiment Modes. We have five different experiment modes: one in the memory test, and four under the threat models.

1) MT represents memory test, in which participants tried to recall answers of the questionnaires we generated;

2) A(ON) and A(OFF) represent the attacks from acquaintances with and without the help of online tools.

3) S(ON) and S(OFF) represent the attacks from strangers with and without help of online tools.

C. True/false Questions

1) Overall percentage of correct answers: Table IV shows the average percentage of correct answers for all true/false questions in different experiment modes. We can observe that there is a decrease in the average percentage of correct answers from MT to S(OFF). The memory test produces a percentage of 86.7%; acquaintance attackers can obtain a percentage around 59.8%; and the stranger can obtain only a success rate about 56.0%, slightly better than a random guess. Ten groups of bars in Figure 4 show the average percentage of correct answers for ten categories of questions under five experiment modes.

2) Performance of Ten Categories of Sensor/app Data: Location (GPS). We can conclude that participants can easily recall their location with a high accuracy rate of 91.7% in the memory test. However, its reliance to attack under A(ON) and A(OFF) is low; for example, the percentage of A(OFF) is 83.3%, which implies a very high success rate of the acquaintance guessing attack. In the meanwhile, we observe that online tools provide little help when answering a question like "Did you leave campus yesterday?".

As known, most of participants may turn off the GPS sensor to save battery life. Hence, it is not recommended for real world deployment due to energy consumption. Alternatively, GPS can be replaced by a location based service using a Wi-Fi or cellular positioning system in the real world deployment.

Motion activity (acc.). The true/false question related to accelerometer data can partially reflect one's "motion" activities. The high reliability and resilience to attacks as shown in Figure 4 indicate that this category of data can be used to create good secret questions.

Smartphone related events (calendar, charging and photo-taking). All these events are major operations of any smartphone user. Based on the results, we conclude that the calendar-related question data can also help create a good secret question owing to either the high reliability or the robustness against attacks.

However, charging the battery and taking photos lead to a vulnerability to statistical guessing attacks (which we will discuss in Section VI-C). Therefore, relevant questions are not good due to the high percentage of correct answers under four attack modes (Figure 4), as the attacker can blindly make a guess whether a user charges the phone or takes a photo everyday.

Legacy apps: contact, call and SMS. Unfortunately, The contact related question is a good one only when online tools are unavailable. Specifically, contact related questions are vulnerable to the guessing attackers using online tools, no matter an acquaintance or a stranger (the percentage of correct answer is 75.0% or 67.2%). This can be attributed to the method of creating a secret question with fake names: an attacker can easily filter out a fake name by searching through the contact list of the target user's OSN sites.

Still, we cannot find many good true/false questions can be created using the data of legacy apps (call and SMS), due to either a low reliability or a high vulnerability (low resilience).

From our prospective, the undergraduates/graduates students in a university are our potential popular users, as the experiment results prove. With respect to the young and highly educated people, we believe our smartphone databased second authentication schemes can be also applied to them as well, because this group of people usually have good memory and relatively high techsavvy obtained in their school days. However, as for the elder people, our approach

may be a bit challenging for them, but the conventional password recovery system does not work well for them.

In this paper, our research provides a guideline that shows which sensors/apps data and which types of questions are suitable for devising secret questions. Researchers are free to investigate more questions for different age groups, which leads to more flexibility in the design of a secondary authentication mechanism.

VI. CONCLUSION

In this paper, we present a Secret-Question based Authentication system, called "Secret-QA", and conduct a user study to understand how much the personal data collected by smartphone sensors and apps can help improve the security of secret questions without violating the users' privacy. We create a set of questions based on the data related to sensors and apps, which reflect the users' short-term activities and smartphone usage. We measure the reliability of these questions by asking participants to answer these question, as well as launching the acquaintance/stranger guessing attacks with and without help of online tools, and we are considering establishing a probabilistic model based on a large scale of user data to characterize the security of the secret questions. In our experiment, the secret questions related to motion sensors, calendar, app installment, and part of legacy apps (call) have the best performance in terms of memorability and the attack resilience, which outperform the conventional secret question based approaches that are created based on a user's long term history/information.

References

[1] R. Reeder and S. Schechter, "When the password doesn't work: Sec- ondary authentication for websites," S & P., IEEE, vol. 9, no. 2, pp.43–49, March 2011.

[2] M. Zviran and W. J. Haga, "User authentication by cognitive passwords: an empirical assessment," in Information Technology, 1990.'Next Decade in Information Technology', Proceedings of the 5th Jerusalem Confer- ence on (Cat. No. 90TH0326-9). IEEE, 1990, pp. 137–144.

[3] J. Podd, J. Bunnell, and R. Henderson, "Cost-effective computer security: Cognitive and associative passwords," in Computer Human Interaction, 1996. Proceedings., Sixth Australian Conference on. IEEE, 1996, pp. 304–305.

[4] S. Schechter, A. B. Brush, and S. Egelman, "It's no secret. measuring the security and reliability of authentication via secret questions," in S & P., IEEE. IEEE, 2009, pp. 375–390.

[5] S. Schechter, C. Herley, and M. Mitzenmacher, "Popularity is every thing: A new approach to protecting passwords from statistical-guessing attacks," in USENIX Hot topics in security, 2010, pp. 1–8.

[6] D. A. Mike Just, "Personal choice and challenge questions: A security and usability assessment," in SOUPS., 2009.

[7] A. Rabkin, "Personal knowledge questions for fallback authentication: Security questions in the era of facebook," in SOUPS. ACM, 2008, pp. 13–23.

[8] J. C. Read and B. Cassidy, "Designing textual password systemsfor children," in IDC., ser. IDC '12. New York, NY, USA: ACM, [9] H. Ebbinghaus, Memory: A contribution to experimental psychology Teachers college, Columbia university, 1913, no. 3.

[9] F. I. Craik and R. S. Lockhart, "Levels of processing: A framework for memory research," Journal of verbal learning and verbal behavior, vol. 11, no. 6, pp. 671–684, 1972.

[10] T. M. Wolf and J. C. Jahnke, "Effects of intraserial repetition on short-term recognition and recall." Journal of Experimental Psychology, vol. 77, no. 4, p. 572, 1968.

[11] A. Babic, H. Xiong, D. Yao, and L. Iftode, "Building robust authen-tication systems with activity-based personal questions," in SafeConfig. New York, NY, USA: ACM, 2009, pp. 19–24.

[12] H. Kim, J. Tang, and R. Anderson, "Social authentication: harder than it looks," in Financial Cryptography and Data Security. Springer, 2012, pp. 1–15.

[13] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, ser. SenSys '13. New York, NY, USA: ACM, 2013, pp. 13:1–13:14. [Online]. Available: http://doi.acm.org/10.1145/ 2517351.2 5173 67

[14] "libsvm on android," GitHub, 2015. [Online]. Available: https: //github.com/cnbuff410/Libsvm-androidjni .

[15] "Sensor event listener on android," Android Developer, 2015. [Online]. Available: http://developer.android.com/reference/ android/ hardware/SensorEventListener.html

[16] J. Clark and P. van Oorschot, "Sok: Ssl and https: Revisiting past challenges and evaluating certificate trust model enhancements," in Security and Privacy (SP), 2013 IEEE Symposium on, May 2013, pp. 511–525.

[17] J. Whipple, W. Arensman, and M. S. Boler, "A public safety application of gps-enabled smartphones and the android operating system," in SMC. IEEE, 2009, pp. 2059–2061.