Understanding Smartphone Sensor and App Data for Enhancing the Security of Secret Questions

Peng Zhao, Kaigui Bian, Tong Zhao, Xintong Song, Jung-Min ''Jerry'' Park, Xiaoming Li, Fan Ye, and Wei Yan

Abstract—Many web applications provide secondary authentication 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 participates 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.

1 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 [2], [3], [4], [5], [6].

The "security" of a secret question depends on the validity of a hidden assumption: *A user's long-term personal history/information is only known by the user himself*. However, this assumption does not hold when a user's personal information can be acquired by an acquaintance, or by a stranger with access to public user profiles. An acquaintance of a user can easily infer the answers to the user's secret questions (e.g., "name of pet") [4]. Moreover, a stranger can figure out the answers leaked from public user profiles in online social networks or search engine results (e.g., "the hospital your youngest child was born in") [7].

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 [8].

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 smartphone 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 [9], [10]. 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/

- *P. Zhao, K. Bian, T. Zhao, X. Song, X. Li, and W. Yan are with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. E-mail: {zhaopeng, bkg, zt, sxt, yanwei}@net.pku.edu.cn, lxm@pku.edu.cn.*
- *J. Park is with the Department of Electrical and Computer Engineering, Virginia Tech., Blacksburg, VA 24060. E-mail: jungmin@vt.edu.*
- *F. Ye is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794. E-mail: fan.ye@stonybrook.edu.*

Manuscript received 6 Jan. 2015; revised 23 Dec. 2015; accepted 16 Mar. 2016. Date of publication 24 Mar. 2016; date of current version 5 Jan. 2017. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2016.2546245

or this activity heavily involves his time and effort in a short time period (e.g., running exercise) [11].

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' *short-term* smartphone usage.
- We evaluated the reliability and security of the three types of secret questions (blank-filling, true/false, and multiple-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 2. In Sections 3, we give an overview of the system design. We present our approach of creating secret questions in Section 4. In Sections 5 and 6, we evaluate the system performance over all created secret questions. We conclude the paper in Section 7.

2 Background and Related work

The blank-filling secret questions are dominant as the mainstream 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 [2], which indicated that the answers of 33 percent questions can be guessed by the "significant others" who were mainly participants' spouses (77 percent) and close friends (17 percent). Another similar study was conducted by Podd et al, which revealed a higher rate of successful guessing (39.5 percent) [3]. 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 [4].

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 [5] (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 [6]. However, today's mainstream secret question methods fail to meet this requirement. A recent study revealed that nearly 20 percent users of four famous webmail providers forgot their answers within six months [4]. 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) [12]. 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 [13]. 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.

3 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 Fig. 1 and will be elaborated next.

3.1 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 [14].

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 1. 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.

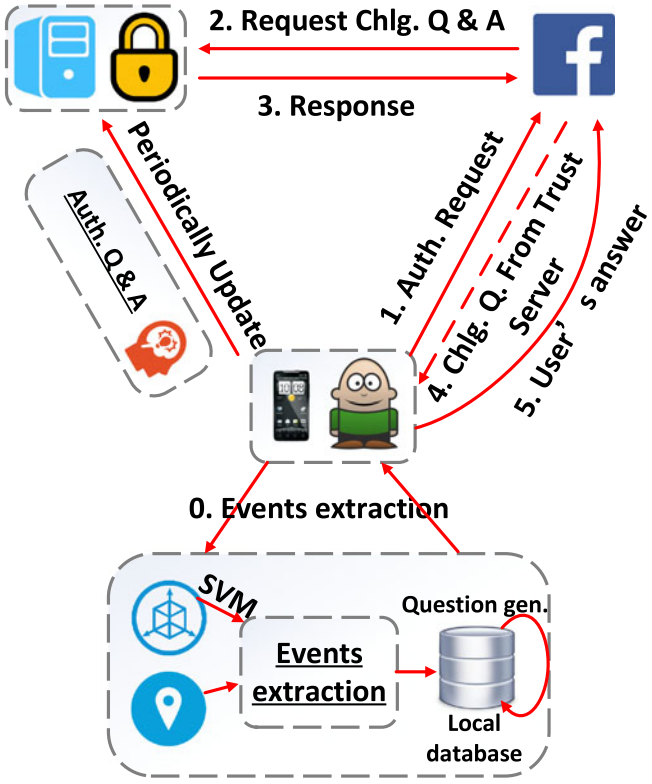


Fig. 1. System architecture of Secret-QA, for a typical user scenario of resetting the account password through answering the secret questions.

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 Fig. 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 4). 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 Fig. 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.

3.2 A Three-Phase Challenge Response Protocol

As shown in Fig. 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 Fig. 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 Fig. 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 Fig. 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 Fig. 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.

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)

3.3 Threat Models

Former studies including [2], [3], [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 versus stranger; with versus 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.

4 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 2 and 3.

TABLE 2
True-False Questions Created in the Experiment

No.	Question	Cat.
1	Did you leave campus yesterday?	GPS
2	**Did you do running exercise for at least 10min yesterday?**	**Acc.**
3	**Is there an item for next week in your calendar?**	**Calendar**
4	Did you charge your phone yesterday?	Charging
5	Did you take photos in the last three days?	Photo-taking
6	Is someone in your contact?	Contact
7	**Did you install some app?**	**App install.**
8	Did you call someone last week?	Call
9	Did you call someone last two weeks?	
10	Did you call someone last month?	
11	Was someone your most frequent contact last week?	
12	Was someone your most frequent contact last two weeks?	
13	Was someone your most frequent contact last month?	
14	Did you text someone last week?	SMS
15	Did you text someone last two weeks?	
16	Did you text someone last month?	
17	Was someone your most frequent SMS contact last week?	
18	Was someone your most frequent SMS contact last two weeks?	
19	Was someone your most frequent SMS contact last month?	
20	Did you use some app last week?	App usage
21	Did you use some app last two weeks?	
22	Did you use some app last month?	
23	Was some app your most frequently used one last week?	
24	Was some app your most frequently used one last two weeks?	
25	Was some app your most frequently used one last month?	

*Questions in **boldface** are* good *ones.*

TABLE 3
Multiple-Choice and Blank-Filling Questions

No.	Question	Cat.
26	Who is in your contact?	Contact
27	**Which app did you install in your phone?**	**App install.**
28	Who did you call last week?	Call
29	Who did you call last two weeks?	
30	Who did you call last month?	
31	**Who was your most frequent contact last week? (b)**	
32	**Who was your most frequent contact last two weeks? (b)**	
33	Who was your most frequent contact last month? (b)	
34	Who was your most frequent contact last week?	
35	Who was your most frequent contact last two weeks?	
36	Who was your most frequent contact last month?	
37	Who did you text last week?	SMS
38	Who did you text last two weeks?	
39	Who did you text last month?	
40	**Who was your most frequent SMS contact last week? (b)**	
41	**Who was your most frequent SMS contact last two weeks? (b)**	
42	Who was your most frequent SMS contact last month? (b)	
43	Who was your most frequent SMS contact last week?	
44	Who was your most frequent SMS contact last two weeks?	
45	Who was your most frequent SMS contact last month?	
46	Which app did you use last week?	App usage
47	Which app did you use last two weeks?	
48	Which app did you use last month?	
49	What was your most frequently used app last week? (b)	
50	What was your most frequently used app last two weeks? (b)	
51	What was your most frequently used app last month? (b)	
52	What was your most frequently used app last week?	
53	What was your most frequently used app last two weeks?	
54	What was your most frequently used app last month?	

*Questions in **boldface** are* good *ones. A question with a marker "(b)" appended means this question is a blank-filling; otherwise it is a multiple-choice question.*

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.

4.1 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 10 min 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 [21].

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 2).

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)?".

4.2 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.

4.3 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 percent for both true/false and multiple-choice questions—i.e., 80 percent 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 percent 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 percent (or 30 percent)—i.e., less than 55 percent (or 30 percent) attackers can correctly guess the answer, which is approximately to be a random guess for true/false (or multiple-choice) questions.

5 Evaluation and Experiments Results

5.1 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 the reliability and the security of the secret questions. In our experiments, we recruited 88 volunteer participants, and carried out a three-phase experiment to study the security and reliability of secret questions that were created using smartphone sensor and app data.

5.1.1 Participant Recruitment

A total number of 88 students (48 males versus 40 females) in a university were recruited, excluding the members of our research lab. Each participant was first asked a questionnaire to indicate their experience of using OSNs, password recovery methods, and smartphones. Results show that many participants with a major in Chinese Literature may have less experience on smartphones' sensors; however, almost all students whose major is computer science are familiar with the concepts above. Hence, in our study, we use these groups of students as representatives of other populations for the following two reasons: (1) The scope of this work is to study whether using smartphone sensor/app data is helpful for secret-question based secondary authentication, and thus we need to exclude the impact of social and demographic factors as much as possible in the

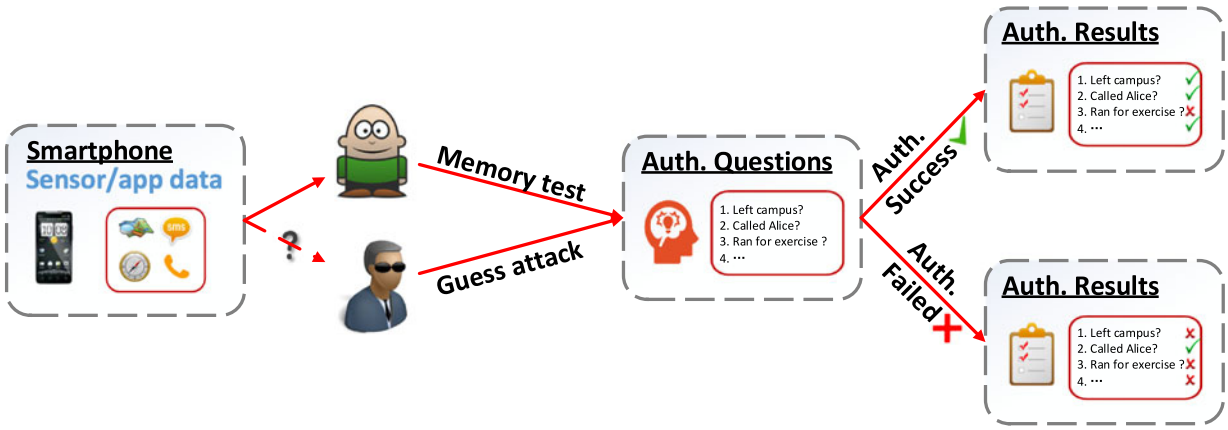


Fig. 2. Three-phase experiment procedure.

experiment, and students are the population that receives the least affection from these factors; (2) Young people like students have the necessary experience on setting and answering secret questions (or completing this experiment), and they use smartphones and online tools (e.g. OSNs, search engines) every day.

Accordingly, participants in our experiment meet the following requirements.

1) Participants should be undergraduates or graduates , and should not be full-time employed (i.e., his/her occupation should still be student);
2) Participants should have used Android[1] smartphone for at least one year;
3) Participants should access at least once per week to one of the well-known OSNs (within the list of social networking websites provided by [24]).

Requirement 1 is set to intentionally limit their cultural backgrounds, ages and careers, because these demographic factors may result in difference of memory performance [25]. Requirements 2 and 3 ensure that participants have the savvy on smartphones and OSNs to complete our experiments.

5.1.2 Three-Phase Experiment Design

We carried out a three-phase experiment as illustrated in Fig. 2: we first collected participants' sensors/apps data using EventLog (however only 42 participants agreed to install EventLog due to privacy concerns), then we asked participants to answer questions related to their data; meanwhile the guess attacks were launched by participants' acquaintances and strangers; the authentication succeeded if the participant could provide most correct answers in most questions. Finally, we invited participants to give feedbacks on experiments.

Degree of Acquaintance Between Participants and EventLog Installation. A questionnaire was assigned to participants to show the acquaintance relationship between any two of them. We simply used integers of 1, 2 and 3 to define three increased levels of acquaintance between two different participants–"never heard about", "know about him/her", "acquaintance".

1. We choose Android smartphones for experiments because of its open API to capture the sensor and app data, without any preference on the smartphone operating systems.

We asked participants to decide whether to install EventLog on their phones, which logged personal information (e.g., call records) and thus it would invade users' privacy in some extent. Only part of participants agreed to install it and proceed the experiments, and the 88 recruited participants are divided into two groups: Group A (42 participants) denoted the set of Android smartphone users that were willing to grant full permissions to install the client software; and Group B denoted the set of other participants that did not install the client software because of privacy concerns.

Privacy Protection. We notified participants in Group A a disclaimer that all the data would be encrypted and only available for experiment analysis, and the data would be destroyed one year after the end of experiments.

Phase 1—Secret Question Generation. Given participants' knowledge and preference, we only created secret questions that utilize the data of top-ten ranked categories in Table 1. The created secret questions would not be disclosed to participants until Phase 2, and each question appeared in form of "true-false", "multiple choice" (select one out of four) or "blank-filling". All the questions created are shown in Tables 2 and 3, and we introduce our approach to generate questions in Section 4. Note that we did not generate all three types of questions for each category of sensor/app or related event due to poor reliability. For example, we did not create such a blank-filling question for the calendar app—"how many meetings will you have tomorrow?"—because such a question incurred too much memory difficulty for the user.

Prevention of Cheating in Phase 1. To avoid cheating behaviors in the experiments, we thus did not publicize the results of Phase 1 study, neither did we inform participants that there would be more experiments in Phases 2 and 3. We told them this was a survey regarding the user behaviors, smartphone usage and its privacy.

Phase 2—Memory Test and Guessing Attack. After one month of Phase 1, we carried out a memory test for 42 participants who installed the EventLog app to answer the questions. Meanwhile, we allowed all 88 participants to launch guessing attacks against other participants' answers.

Design of the Memory Test. We ran a memory test for each participant in Group A to answer the secret questions generated by his own data, according to his memory in the past one month. A question could be skipped if the participant

TABLE 4
The Average Percentage of Correct Answers for All Sensor/App Events, Under Each Experiment Mode

Experiment modes	Percentage
MT	86.7
A(ON)	60.9
A(OFF)	58.8
S(ON)	57.0
S(OFF)	55.1

was strongly unwilling to answer (due to privacy issues). It was also feasible that a participant did not keep our EventLog online all the time, and thus his data was insufficient to generate all secret questions. In addition, every participant should indicate the extent of invasion of privacy for each question, details will be discussed in Section 5.7.

Assignment of Acquaintance/Stranger Attackers. In the experiment, all participants joined the guessing attacks against each participant in Group A, such that each participant in Group A would be attacked by at least three acquaintances (that had indicated a degree of acquaintance value of 3) and at least three strangers (that had indicated a degree of acquaintance value of 1). Note that a participant in Group A could be a participant who joined the memory test, and also an attacker who guessed the answers to others' questions.

Our lab assistants distributed every attacker a slip of paper, indicating their attack targets with student IDs and full names. Targets were a combination of acquaintances and strangers, up to four people.

In each attack launched by the attacker i against participant $j \in A$, the attacker would acquire the same questions their target had answered. They were first required to guess the answers to questions of each of their targets, without any external help. Then, we encouraged them to use search engines, OSNs and campus information systems (public online tools) to research and guess the answer *again*.

Prevention of Cheating in Phase 2. To eliminate hints and prevent collusion as much as possible during the tests, we enforced the following rules in Phase 2.

- Each question could be answered once and only once via our custom built web-questionnaire interface.
- Participants would not know the next question before finishing the current one.
- If a question would be asked in more than one type, then this question would appear in the order of "blank-filling", "multiple choice" and then "true/false".

We restricted participants from communicating with each other by asking them to turn off their mobile devices (announced as a courtesy to other participants), isolating them in separate rooms, and monitoring their behaviors. All participants were proctored by lab assistants who were responsible for preventing cheating, monitoring their online tools usage, and providing participants with guidance about how to input on the web-questionnaire as well.

Phase 3—Post-test Feedback. We found that some of experiment results obtained in Phases 2 were hard to explain, and thus we invited participants to discuss with us about their answers as well as to make comments. In total, 32 participants joined our Phase 3 experiment to provide us feedbacks. The detailed feedbacks are presented in Section 5.7, 6.

5.2 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 percent, 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.

5.3 True/False Questions

5.3.1 Overall Percentage of Correct Answers

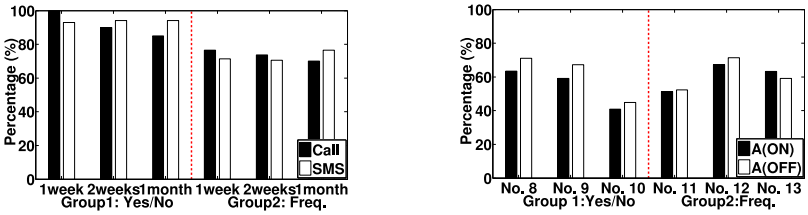
Table 4 shows the average percentage of correct answers for all true/false questions in different experiment modes. We can observe that there is a decrease in the average percentage of correct answers from MT to S(OFF). The memory test produces a percentage of 86.7 percent; acquaintance attackers can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

5.3.2 Performance of 10 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 percent 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 percent, 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



(a) (b)

Fig. 3. (a) Results of call and SMS related questions in the memory test. (b) Online tools are misleading for attackers.

shown in Fig. 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 6.3). Therefore, relevant questions are not good due to the high percentage of correct answers under four attack modes (Fig. 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 percent). 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) to attacks. As shown in Fig. 4, the average percentage is of slight difference between categories of call record and SMS in every experiment mode.

We create two groups of questions regarding the legacy app data: (1) the Yes/No group include question No. 14–16 and 20–22 and (2) the frequency-based group has 17–19, 23–25. We are able to make the following observations shown in Figs. 3a and 3b.

1) The reliability of call-related questions keeps decreasing, but that of SMS increases, as the experiment period increase from one week to one month.
2) Regular yes/no questions have a higher reliability than those based on frequency in either the call record or SMS category.
3) Online tools are misleading for acquaintance in some way, because the percentage of A(ON) is often lower than A(OFF) in the category of call.

Feedbacks of participants collected in Phase 3 and can better interpret these observations:

1) It is easier to memorize the SMS record in a long period of time for emerging adults, but it is better to memorize the call record in a short term;
2) It is easier to answer Yes/No questions rather than frequency based questions.
3) College students' offline communication behaviors via phones are different from their online behaviors over the OSNs. Thus, online tools can be misleading when answering the secret questions related to offline call and SMS activities.

App Installment/Usage. The true/false question related to the app installment is a good question, with a percentage of about 50.0 percent against all attack modes. Moreover, when we exclude the questions related to pre-installed apps (e.g., Samsung app stores) and only use data based on the third-party apps. Then, we observe a 8.4 percent increase in the memory test, without an significant increase under each threat model.

Given questions on app usage, the participant has a high rate of recalling the correct answers (over 80 percent), while the attacker has a success rate around the threshold 60 percent as shown in Fig. 4. Then, we use two groups of questions to investigate the security and reliability of these questions in details. The *Yes/No group* includes questions No. 20, 21 and 22 that are non-frequency based, and *Freq. group* 2 includes frequency-based questions No. 23, 24 and 25.

As shown in Fig. 5a, the reliability of questions in *Yes/No group* decreases as the length of the experiment period increases from one week to one month. Thus, the secret questions should be created based on the data collected within one or two weeks, otherwise more than 20 percent participants may forget the answers.

Fig. 5b compares the two groups of questions in terms of the resilience to four attack modes. Results show that frequency-based group 2 questions are less secure than group 1 questions, and the attackers can further increase

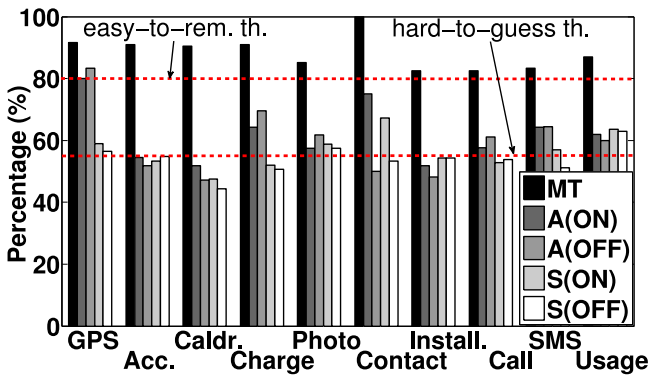
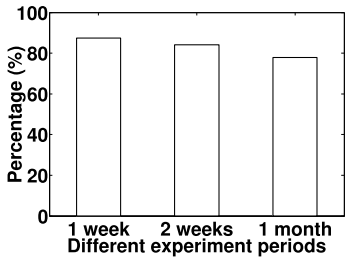
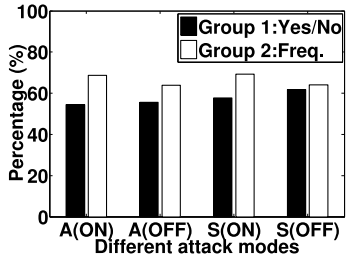


Fig. 4. Average accuracy in different experiment modes of true/false questions. Two dotted lines represent the two thresholds of determining the easy-to-remember and hard-to-guess questions.



(a) Reliability decreases in MT when the experiment period increases;



(b) Difference in resilience to attacks between groups.

Fig. 5. Results in app usage questions.

their success rates using online tools. This observation is attributed to two facts: (1) the frequently used apps may be similar for the attacker and the participant, and it is easy for an attacker to make a guess; and (2) the attacker can figure out which app the participant frequently uses by searching with help of OSNs.

Our conclusion is further justified by the participants themselves based on the participants' feedbacks collected in Phase 3 of the experiment: as time elapses, the participant is not sure whether a certain app has been used. However, as for attackers, it is not surprising that the attack success rate increases when cracking the frequency based questions, given that there are quite a few apps that dominate the users' time [23].

5.4 Multiple-Choice Questions

Multiple-choice questions focus on the following five categories: contact, app installment, app usage, call and SMS. Unlike true/false questions, only one type of multiple-choice questions (app installment related) is considered *good* according to the results in Fig. 6.

Legacy Apps: Contact, Call and SMS. Multiple-choice questions on contacts are vulnerable to attackers with online tools. Specifically, acquaintance attackers of A(ON) gains a success rate of 67.6 percent, which is 18.5 percent higher than the rate gained by attackers of A(OFF); meanwhile, stranger attackers of S(ON) gains a success rate that is 33.3 percent higher than that of attackers of S(OFF).

Questions related to call and SMS are terribly vulnerable to acquaintance attacks. In contrast, strangers are unable to crack the answers to these multiple-choice questions with or without online tools. So generally speaking, legacy apps related question in multiple-choice format cannot be used due to the vulnerability to acquaintance attacks.

App Installment and Usage. Results show that the multiple-choice question related to the app installment data is both secure and reliable to serve as a good secret question. In contrast, the questions related to the app usage fail to maintain a high reliability or a low resilience to the guessing attack.

5.5 Blank-Filling Secret Questions

Blank-filling questions are only available for categories of call record, SMS and app usage data, and they are all frequency-based questions. We offer participants to have up to two attempts to input the answer to a question, and the input is determined as correct if either attempt hits the answer.

5.5.1 Reliability and Resilience to Attacks

According to [4], conventional secret question based authentication methods demonstrate a reliability of at least 80 percent (easy-to-memorize threshold), and the success attack rate of less than 22 percent (hard-to-guess threshold). As it shows in Fig. 7, the average reliability of blank-filling questions for every category is approximately 80 percent in most cases, and the resulting success attack rates are much lower than 22 percent (except the category of app usage). Therefore, the security of blank-filling secret questions to attacks can be enhanced using smartphone data on call and SMS records.

5.5.2 Willingness to Answer

In the experiment, we exclude all answers in which participants expressed being unable to provide an answer after manually identifying numerous indicators from participants, such as "unknown" and "don't have one".

Despite little unwillingness and uncomfortableness reported by participants, there are some individuals who provide the exact same answer (2.6 percent) in the second attempt when dealing with blank-fillings; some others (0.9 percent) provide futile answers such as unintelligible "nicknames", references to someone unknown (e.g., "his girlfriend"), and a long string of random characters. We do not remove these answers because they may be a real-world problem for unskilled or troublemaking attackers, even if the answers are useless in this case.

In overall, the high willingness (they are willing to answer 1,442 among 1,479 questions) shows the effectiveness of answer collection in the experiment of blank filling questions.

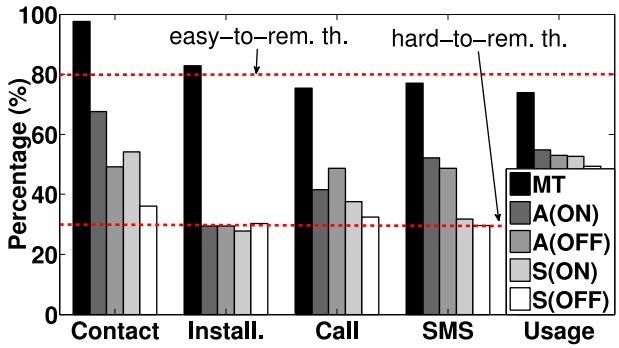
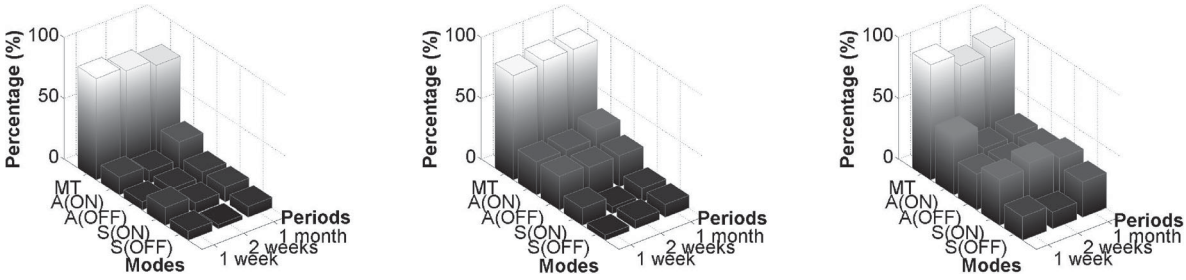


Fig. 6. Average accuracy in different operations modes of multiple-choice questions. Two red dotted lines represent the two thresholds of determining the easy-to-remember and hard-to-guess questions.



(a) Call; (b) SMS; (c) App usage.

Fig. 7. Reliability and resilience to attacks of blank-filling questions.

5.6 Authentication by Combining Multiple Lightweight Questions

Existing authentication methods depend mainly on blank-filling questions, because the lightweight questions are subject to the random guessing attacks, e.g., a 50 percent success rate for an attacker given a true-false question.

However, it is feasible to combine multiple lightweight (e.g., true-false and/or multiple choice) questions sequentially to lower the success rate for an attacker. The reduction of attacker's success rate depends on how many lightweight questions we want to combine. For example, three lightweight true-false questions will not incur too much user-input efforts, but lead to a low success rate of 12.5 percent for a random guessing attacker, which is lower than the hard-to-guess threshold.

We conduct an experiment by randomly choosing one of four existing threat models, and authenticate a participant's identity if he/she correctly answers at least three out of four questions in Table 5. This leads to a very low success rate of no more than 7.5 percent, given any threat model.

5.7 Feedbacks on Invasion of Privacy

We require each participant to choose an integer from $[1, 5]$ in order to indicate the invasion of privacy in the EventLog experiment, where 1 represents "no privacy invasion", 3 represents "moderate privacy invasion", 5 represents "severe privacy invasion".

SMS is Privacy-Sensitive. As shown in Table 6, even if we promise not to read the contents of SMS, participants still feel that their privacy is invaded when we collect SMS data. As a result, SMS data is not recommended to be used for creating secret questions as a part of authentication.

Local Deployment to Minimize Privacy Disclosure. As mentioned before, secret questions can be generated locally based on the data collected by the phone itself, without being uploaded to the cloud, thus minimizing the risk of privacy disclosure at the server side. Besides, protecting user privacy in the cloud have been widely studied [26], [27], which is out of the scope of this study.

Potential Privacy Leakage Exists in Questions. Although we tried to eliminate hints as much as possible, we discover that the existence of hints in the options to a question may result in potential privacy leakage. For example, the attacker may be able to obtain four potential contacts by guessing the question *Who did you call last week?* (four options provided). Similar privacy leakage also exists in true/false questions, however the degree of privacy intrusion decreases significantly because true/false questions provide only options of true or false. If the attacker have unlimited times of trying the same question repeatedly, the privacy leakage would be severe.

To this end, blank-filling questions are preferred to devise the authentication scheme. Moreover, in the real-world deployment, the number of trying to answer the challenge questions should also be limited to avoid above potential privacy leakage.

6 Discussion

Our results prove that questions related to motion sensors, calendar, app installment, and part of legacy app usage history (e.g., phone calls) have the best performance. Hence, we discuss the overarching issues related to real-world deployment and experimental details here.

6.1 Feedback on Battery Life

In experiment Phase 3, we also survey the battery usage of participants. However, 87.5 percent of participants complains that EventLog consumes much battery. We analyze the reasons blaming for high battery consumption as the following:

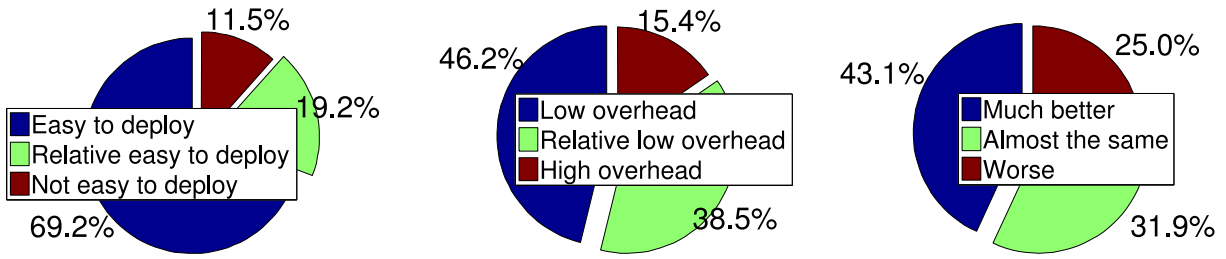
- The EventLog app requires a coarse-grained GPS service, and such a service consumes excessive battery.
- The EventLog app polls every 30 seconds to track the app on-screen using Android API, and such

TABLE 5
A Combination of Four Lightweight Questions

No.	Question
2	Did you do running exercise for at least 10min yesterday?
3	Is an item planned for next week in your calendar?
7	Did you install some app?
27	Which app did you install in your phone? (multiple choice)

TABLE 6
Participants' Indications of Privacy Invasion

Events	Avg. level	Events	Avg. level
GPS	2.0	Contact	2.7
Accelerometer	1.3	App installment	2.0
Calendar	2.2	Call	2.6
Charging	1.2	**SMS**	**3.5**
Photo-taking	2.1	App usage	2.4



(a) Deployment cost; (b) Overhead; (c) In contrast with conventional schemes.

Fig. 8. Feedbacks on system usability in three aspects.

behaviors demand a continuous workload of the CPU, and therefore the battery runs out quickly.

The first challenge is also studied by [28], in which WiFi (cellular) positioning system is adopted to replace GPS service to reduce the high battery overhead. As our approach only studies whether the user has left the campus, location information from WiFi or cellular positioning system (approximately 500 meters [29]) is of sufficient accuracy. Researchers are free to add more location aware questions by adopting other positioning systems.

We can address the second challenge by killing the "polling" backend service when users lock the screen; and restart it when users unlock the screen using Android API [30] (obviously the user cannot start any app when the phone is locked), in this way we can significantly reduce the battery consumption while the smartphone enters lockup (background) mode without missing any application launch events.

Finally, prior work on general battery life optimization [31] can also be applied to this case; however, we design our EventLog as an experimental prototype rather than a real-world product to-be-deployed, which is out of the scope of this study.

6.2 System Usability and Overhead

We further asked the participants to evaluate the system usability in the following three aspects.

1) *The Deployment Cost*. As shown in the following Fig. 8a, most users consider Secret-QA client app easy to install and use on their smartphones, because our client app is mostly running backend. Note that the EventLog client app requires users' operation only for the client setting and the secret-question based authentication.
2) *Overhead*. Fig. 8b shows that the overheads are acceptable for some users in our EventLog app with battery optimization. In future work, we will try to adopt WiFi or cellular location based service instead of GPS to further improve the battery life. Besides, the HTTPS traffic cost is almost negligible because our system will train and classify the motion related events locally, rather than sending the raw data of accelerometer/gyroscope to the server, and the root cause for HTTPS cost is the periodic update of secret questions/answers in an encrypted format.
3) *Comparison With Conventional Secret-question Based Authentication Schemes*. When compared with conventional secret-question based authentication methods, most users consider that it is easier to memorize the answers under Secret-QA and it has a better security against the guessing attacks due to the dynamic generation of the questions regarding to short-term user events.

6.3 Vulnerability to Statistical Guessing

The statistical guessing attack aims at identifying the most popular answers to each question and trying each one until no more guesses are allowed. Our research findings indicate that three categories of questions related to battery charging, photo-taking, and app usage, are statistically guessable, as shown in Fig. 9.

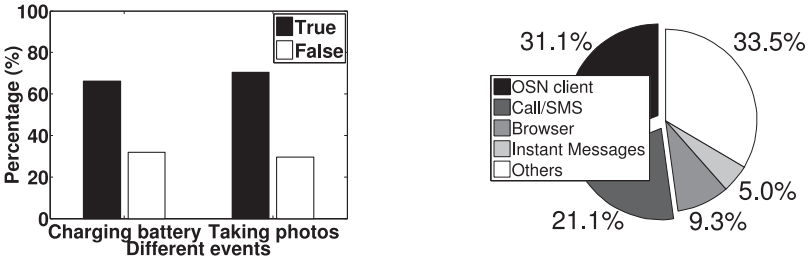
- Questions derived from battery charging data should be reconsidered, as the similar result of statistics and feedbacks implying that emerging adults are likely to charge their phones every day.
- It is possible for an attacker to crack questions like "Did you take any photos using camera in the last three days?" by just answering "yes" though, the answer may change due to demographic factors and users' behaviors.
- In terms of app usage, the top 10 percent popular apps can cover more than 50 percent of answers: a mobile client of OSN ranks first, with a percentage of 31.1 percent. Legacy apps come to the second place. The third, and the forth ones are the browsers and instant messaging apps.

6.4 Frequency Affects Reliability among Blank-Filling Questions

We studied the relationship between the frequency and reliability in our follow-up study, in which 26 participants were invited to use EeventLog for one more week, but this time we only concentrated on events/apps related to blank-filling questions.

The results are summarized as below: participants have 5.6 calls/4.3 SMS a day on average (this number may vary in different groups of people), however the number of apps they launch in a day varies from 31 to 147 (note that the same app launched at different times are all treated as different launch events). We then ask them to answer the following 9 questions:

- *Who was your the most and 2nd/3rd most frequent contact last week? (call record)*



(a) Charging, photo-taking; (b) Popular apps.

Fig. 9. Vulnerability to statistical guessing.

- *Who was your the most and 2nd/3rd most frequent SMS contact last week? (SMS)*
- *What was your the most and 2nd/3rd most frequently used app last week? (app usage)*

As shown in the Fig. 10, questions on the most frequent events have the best reliability (80 percent and up), which conforms to our previous experimental results. However, regarding the 2nd or 3rd most frequent events, the reliability decreases are different. For example, the reliability of 2nd most frequently used app drops drastically, because the participants complains that they use varies of apps a day and they cannot figure out the correct answer with so many candidates.

6.5 Participants' Demographic Factors

During the experiment, participants are limited to students who have the savvy of smartphones and OSNs, because we intend to exclude the impact of social and demographic factors as much as possible. But in the meanwhile, we also narrow down the scope of potential users.

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 data-based second authentication schemes can be also applied to them as well, because this group of people usually have good memory and relatively high tech-savvy 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.

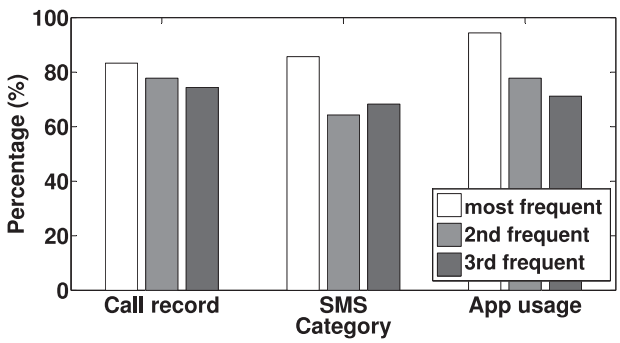


Fig. 10. The reliability varies between the most and 2nd/3rd most frequent events.

7 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: Secondary authentication for websites," *IEEE Security Privacy.*, vol. 9, no. 2, pp. 43–49, Mar. 2011.

[2] M. Zviran and W. J. Haga, "User authentication by cognitive passwords: An empirical assessment," in *Proc. 5th Jerusalem Conf. Inf. Tech., Next Decade Inf. Tech., (Cat. No. 90TH0326-9)*, 1990, pp. 137–144.

[3] J. Podd, J. Bunnell, and R. Henderson, "Cost-effective computer security: Cognitive and associative passwords," in *Proc., 6th Australian Conf. Comput.-Human Interaction*, 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 *Proc. 30th IEEE Symp. Security Privacy.*, 2009, pp. 375–390.

[5] S. Schechter, C. Herley, and M. Mitzenmacher, "Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks," in *Proc. 5th USENIX Conf. Hot Topics Security*, 2010, pp. 1–8.

[6] D. A. Mike Just, "Personal choice and challenge questions: A security and usability assessment," in *Proc. 5th Symp. Usable Privacy Security*, p. 8. ACM, 2009.

[7] A. Rabkin, "Personal knowledge questions for fallback authentication: Security questions in the era of facebook," in *Proc. 4th Symp. Usable Privacy Security*, 2008, pp. 13–23.

[8] J. C. Read and B. Cassidy, "Designing textual password systems for children," in *Proc. 11th Int. Conf. Interaction Des. Children*, 2012, pp. 200–203.

[9] H. Ebbinghaus, *Memory: A Contribution to Experimental Psychology*. New York, NY, USA: Teachers college, Columbia University, 1913, no. 3.

[10] F. I. Craik and R. S. Lockhart, "Levels of processing: A framework for memory research," *J. Verbal Learning Verbal Behavior*, vol. 11, no. 6, pp. 671–684, 1972.

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

[12] A. Babic, H. Xiong, D. Yao, and L. Iftode, "Building robust authentication systems with activity-based personal questions," in *Proc. SafeConfig*. 2009, pp. 19–24.

[13] H. Kim, J. Tang, and R. Anderson, "Social authentication: Harder than it looks," in *Proc. 16th Int. Conf. Financial Cryptography Data Security*, 2012, pp. 1–15.

[14] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proc. 11th ACM Conf. Embedded Networked Sens. Syst.*, 2013, pp. 13:1–13:14. [Online]. Available: http://doi.acm.org/10.1145/2517351.2517367.

[15] (2015). libsvm on android, *GitHub* [Online]. Available: https://github.com/cnbuff410/Libsvm-androidjni.

[16] (2015). Sensor event listener on android, *Android Developer* [Online]. Available: http://developer.android.com/reference/android/hardware/SensorEventList ener.html.

[17] J. Clark and P. van Oorschot, "Sok: Ssl and https: Revisiting past challenges and evaluating certificate trust model enhancements," in *Proc. IEEE Symp. Security Privacy*, May 2013, pp. 511–525.

[18] J. Whipple, W. Arensman, and M. S. Boler, "A public safety application of GPS-enabled smartphones and the android operating system," in *Proc. IEEE Int. Conf. Syst., Man Cybernet.*, 2009, pp. 2059–2061.

[19] S. Kumar, M. A. Qadeer, and A. Gupta, "Location based services using android (lbsoid)," in *Proc. IEEE Int. Conf. Internet Multimedia Services Archit. Appl.*, 2009, pp. 1–5.

[20] (2013). Android api reference about location criteria [Online]. Available: http://developer.android.com/reference/android.

[21] M. Oner, J. A. Pulcifer-Stump, P. Seeling, and T. Kaya, "Towards the run and walk activity classification through step detection-an android application," in *Proc. Conf. IEEE Eng. Med. Biol. Soc.*, 2012, pp. 1980–1983.

[22] W. Luo, Q. Xie, and U. Hengartner, "FaceCloak: An architecture for user privacy on social networking sites," in *Proc. Int Conf. Comput. Sci. Eng.*, 2009, vol. 3, pp. 26–33.

[23] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," in *MobiSys.*, 2010, pp. 179–194.

[24] (2013). Top 15 most popular social networking sites until march 2014, *eBizMBA* [Online]. Available: http://www.ebizmba.com/articles/social-networking-websites.

[25] L. Nyberg, L. Bäckman, K. Erngrund, U. Olofsson, and L.-G. Nilsson, "Age differences in episodic memory, semantic memory, and priming: Relationships to demographic, intellectual, and biological factors," *J. Gerontology Series B: Psychological Sci. Social Sci.*, vol. 51, no. 4, pp. P234–P240, 1996.

[26] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[27] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[28] A. Drozd, S. Benford, N. Tandavanitj, M. Wright, and A. Chamberlain, "Hitchers: Designing for cellular positioning," in *Proc. 8th Int. Conf. Ubiquitous Comput.*, 2006, pp. 279–296. [Online]. Available: http://dx.doi.org/10.1007/11853565_17.

[29] R. Faragher and P. Duffett-Smith, "Measurements of the effects of multipath interference on timing accuracy in a cellular radio positioning system," *Radar, Sonar Navigat., IET*, vol. 4, no. 6, pp. 818–824, Dec. 2010.

[30] (2014). Android service API introduction, *Google Android API* [Online]. Available: http://developer.android.com/reference/android/app/Service.html.

[31] M. Dong, T. Lan, and L. Zhong, "Rethink energy accounting with cooperative game theory," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 531–542. [Online]. Available: http://doi.acm.org/10.1145/2639108.2639128.

Peng Zhao is currently working toward the master's degree in the School of EECS at Peking University. His research interests focus on mobile security.

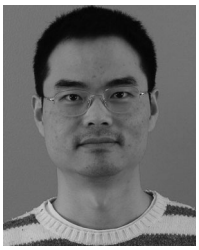
Kaigui Bian received the PhD degree in computer engineering from Virginia Tech, Blacksburg, USA in 2011. He is currently an associate professor in the Institute of Network Computing and Information Systems, School of EECS, Peking University. His research interests include mobile computing, cognitive radio networks, network security, and privacy.

Tong Zhao received his PhD degree in computer science from Peking University, Beijing, China in 2014. He is currently an engineer in the Institute of Network Computing and Information Systems, School of EECS at Peking University. His research interests include ad-hoc networks, mobile computing, and network performance analysis.

Xintong Song received the BSc degree in computer science and technology from Peking University, Beijing, China, in 2012. He is currently working toward the PhD degree from the Institute of Network Computing and Information Systems, School of EECS, Peking University. His research interests include mobile crowd sensing, mobile computing, and wireless networks.

Jung-Min "Jerry" Park received the PhD degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2003. He is currently a professor in the Department of Electrical and Computer Engineering at Virginia Tech and the Site Director of an NSF Industry-University Cooperative Research Center (I-UCRC) called Broadband Wireless Access and Applications Center (BWAC). He is an executive committee member of the National Spectrum Consortium. His research interests include cognitive radio networks, dynamic spectrum sharing, wireless security and privacy, applied cryptography, and wireless networking. Current or recent research sponsors include the US National Science Foundation (NSF), National Institutes of Health (NIH), Defense Advanced Research Projects Agency (DARPA), Army Research Office (ARO), Office of Naval Research (ONR), and a number of industry sponsors. He is a recipient of a 2014 Virginia Tech College of Engineering Faculty Fellow Award, a 2008 NSF Faculty Early Career Development (CAREER) Award, a 2008 Hoeber Excellence in Research Award, and a 1998 AT&T Leadership Award. He is currently serving on the editorial boards of the *IEEE Transactions on Wireless Communications* and the *IEEE/KICS Journal of Communications and Networks*.

Xiaoming Li received the PhD degree in computer science from Stevens Institute of Technology (USA) in 1986 and now he is a professor at Peking University, Beijing, China. His research interests include Web search and mining online social network analysis. He is an editor of *Concurrency and Computation* and *Networking Science.*



Fan Ye received the BE and MS degrees from Tsinghua University, Beijing, China, and the PhD degree from UCLA, Los Angeles, CA, USA. He is an assistant professor in the ECE Department of Stony Brook University, Stony Brook, NY, USA. He has published more than 60 peer reviewed papers with over 7,000 citations according to Google Scholar. He has 21 granted/pending US and international patents/applications. He was the co-chair for the Mobile Computing Professional Interests Community at IBM Watson for two years. He received the IBM Research Division Award, 5 Invention Achievement Plateau awards, and the Best Paper Award for International Conference on Parallel Computing 2008. His current research interests include mobile sensing platforms, systems and applications, Internet-of-Things, indoor location sensing, and wireless and sensor networks.

Wei Yan received the MS degree in computer science from the National University of Defense Technology, Changsha, China. She is currently an associate professor in the Institute of Network Computing and Information System, School of EECS at Peking University. Her major research interests include mobile networks, security and privacy, Internet of Things, and Internet of Vehicles.