Determining Trustworthiness and Quality of Mobile Applications

Ilung Pranata, Rukshan Athauda, and Geoff Skinner

School of Design, Communication and IT
University of Newcastle, Australia
University Drive, Callaghan, NSW 2300, Australia
{Ilung.Pranata,Rukshan.Athauda,Geoff.Skinner}@newcastle.edu.au

Abstract. The growth of "smart" mobile devices, such as smartphones and tablets, has been exponential over the past few years. Such growth was mainly attributed to the development of mobile applications. To date, mobile applications have been increasingly used to improve our productivity and also to provide the entertainment contents. However, with a huge number of mobile applications that appear in the application stores; in particular those that provide similar functionalities, users are often confused with the selection of trustworthy and high quality mobile applications. At the current state, there is a limited research embarked to provide solutions for measuring the trustworthiness of mobile applications prior to download. Thus, the aims of this paper are to review the current research in this area and to discuss several issues in measuring the trustworthiness of mobile applications. In addition, this paper also proposes MobilTrust, a similarity trust measurement method to solve the identified issues.

Keywords: trust, reputation, mobile application.

1 Introduction

The proliferation of mobile computing technology has gained a significant momentum since its first introduction in the 70s. This can be seen from its growth rate that has rocketed over the years. According to the International Telecommunication Union (ITU), the subscribers of mobile devices have surpassed 5.3 billion in 2010 [1] while the total world population in the same year was just about 6.8 billion [2]. Such figures show that in 2010 alone, the percentage of mobile device subscribers is accounted for more than 75% of the world population. In the past few years, "smart" mobile devices such as smartphones and tablets have dominated the growth of mobile devices. An independent research firm IDC [3] published a study that shows the growth of "smart" mobile devices will reach 659.8 million in 2012, up to 33% from the previous year. Furthermore, IDC also forecasted that such growth will remain double digit in the years to come. This is mainly due to the strong user demand and also the production shift from the traditional mobile devices to the new era of "smart"

C. Borcea et al. (Eds.): MobilWare 2012, LNICST 65, pp. 192–206, 2013.

mobile devices. Therefore, it is evident that "smart" mobile devices such as smartphones and tablets have and will continue to become part of our everyday life.

The growth of "smart" mobile devices over the past few years has been predominantly caused by the exponential growth of mobile applications (termed as Mobile Apps). Such cause is mainly due to several benefits that Mobile Apps offer to improve their users' quality of life, such as functionalities, productivity improvements, entertainments, etc. For many years, the development of Mobile Apps was centered and managed by the device manufacturers, network operators and content providers. However, the introduction of application stores (i.e. Apple apps store [4] and Android Google Play [5]) has opened up application businesses to the hand of freelance developers and start-up companies. Since then, the number of mobile applications has increased exponentially. For example, Apple application store that started with only 500 apps in 2007 has reached 350k apps by March 2011. Similarly, Android Google Play reached 250k apps in the same time period [6].

Although mobile devices and its applications provide great benefits, it also produces significant threats for both individuals and organizations. Threats on the confidentiality of critical information and data privacy are just a few. Thus, in order to reduce such threats, there is a need for the users to trust Mobile Apps prior to downloading and consuming them. However, with a huge number of Mobile Apps appears in the application stores, many individuals and organizations are unsure on how to determine their trustworthiness. Nevertheless, determining the trustworthiness and quality of a mobile application is crucial. Thus, in this paper, we put our focus on discussing several issues pertaining to Mobile Apps trust measurement, and we also review several existing works in trust management. Additionally, we propose MobilTrust, a similarity method for determining the trustworthiness value of Mobile Apps.

The remainder of this paper is organized as follow: section 2 provides several reasons as to why initial trust of mobile applications is important to be determined, section 3 reviews several existing works in online trust, section 4 provides several issues in determining the initial trust of mobile applications, section 5 details our solution (termed as MobilTrust) to solve the identified trust issues, section 6 presents the implementation strategy for MobilTrust, section 7 provides the experimental simulations of MobilTrust, and section 8 provides the conclusion of this paper.

2 Why Initial Trust in Mobile Applications

Trust in electronic forefront, according to Grandison et al. [7], is defined as the competency belief that an agent would act reliably, dependably and securely within a given context. Further, authors in [7, 8] stress the importance of trust for the success implementation of any online environment. That is, trust significantly affects the decision of an entity to transact with other entity. The authors argue that both consumers and providers in an electronic market must trust each other before decisions to consume or to provide the services are made. If trust is not established between them, entities will not fully share their resources and fraudulent transactions may occur regularly. Such situation would disadvantage the honest consumers and

providers, and it further refrain them from taking the advantage of the online environment.

Similar to the online environment, trust also plays a pivotal role in the mobile applications environment. With hundreds of thousands Mobile Apps that appear in the application stores, customers are always faced to make a decision whether to download and/or to consume the Mobile Apps. Such decision is even harder to make when there are several Mobile Apps that have similar functionalities appear in the application stores as customers need to decide the most trustworthy mobile application. From customers' point of view, they always prefer to download and consume a Mobile App that is functional, reliable and also with a good quality. However, selecting such functional, reliable and high quality Mobile App is challenging. This can be seen from several customers' comments that are found in the application stores in which customers downloaded the bad quality Mobile Apps, and they are frustrated with such buggy and low performance Mobile Apps. Therefore, there is a critical need to build the initial trust of Mobile Apps prior to downloading and consuming them.

From the security and privacy view point, the emergence of Mobile Apps further produces a number of threats to the confidentiality of information and data. A number of incidents occurred where Mobile Apps mined and harvested customer's confidential data, such as address books, photos, etc. [9, 10]. Such incidents clearly show the violation towards customer privacy and further disadvantage the customers. However, sadly to say, research in [11, 12] shows that more than half of popular Mobile Apps in Android and IOS under the study are transmitting customer data to the external servers. Besides the individual privacy concern, a growing number of organisations and businesses are also critical on the use of mobile devices by their employees [13]. They are extremely concerned about the capability of Mobile Apps to access and harvest the critical and confidential business documents (e.g. through business emails in the mobile devices). To address this concern, some businesses and organisations have prevented employees for using their devices for business related activites while most of them have implemented security measure and BYOD (Bring Your Own Devices) policies.

While implementing security measures and policies may reduce the risk of confidential business documents being released to the public, such measures and policies must also be supplemented and strengthened through the use of trust measurement. Most security practitioners would say that the best way to reduce the risk of documents leakage in Mobile Apps is by not installing the applications in the first place. However, such approach may not be favorable for the employees and businesses, particularly when Mobile Apps improve employee's productivity and bring benefits for businesses. Therefore, the efforts to safeguard the critical business information are left with two methods: (i.) educating employees for selecting the valid Mobile Apps, and (ii.) providing means to measure the trustworthiness of Mobile Apps prior to downloading and consuming them. Measuring trust of Mobile Apps is crucial as it provides the first and additional layer to security and privacy protection. This is also supported by authors in [14, 15] who argue that trust supplements security such that it improves the security protection of information and resources.

3 Related Work

At the current state, to the best of authors' knowledge, there is none research embarked in measuring the trustworthiness and quality of Mobile Apps. There are, however, several research that focus at protecting the security and privacy of user information from Mobile Apps, such as TaintDroid [11], PrimAndroid [17], etc. Although such research is important to reduce privacy violation and data leakage, the protection that they provide is functioning only after the user has downloaded or consumed the Mobile Apps, not prior to downloading or consuming them. This is where trust, as discussed in previous section, provides an extra layer and also serves as the first layer of protection.

Several prominent application stores, such as Apple application store and Android Google Play use a rating system to measure the trustworthiness and quality of the listed Mobile Apps. The recommender (or rater) is someone that has downloaded and consumed a Mobile App, and therefore he/she could provide the rating (in scale of 1 to 5 stars) and comment for others. The total rating of a Mobile App is the average of all raters' comments. Other users, particularly those who have not downloaded the Mobile App, tend to view the rating before making a decision as to whether to download the Mobile App. While the rating system is popular in use by several application stores, Authors in [18] show that such traditional rating system is prone to several misuses and unfair computation. Additionally, such rating system is also prone to several threat strategies as they do not measure the honesty of raters in providing their reviews. Moreover, our review on the jailbreak community (i.e. iPhone users that do not want to use the restrictive Apple Application Store but instead, they look for alternative markets, such as Cydia Market [19]) shows that there is no rating mechanism in presence to measure the trustworthiness of Mobile Apps.

Due to the limited research focuses at measuring the trustworthiness of Mobile Apps, we extend the literature review to the current internet environments, such as peer-to-peer, e-commerce and mobile agent. Literature review classifies trust mechanisms into two main categories: centralized mechanism and decentralized mechanism. The centralized mechanism relies on single point of collection and computation of trust value. PathTrust [20], peer-to-peer multi-dimensional trust model [21], DEco Arch [22], and the e-commerce trust models such as Certificate Authority (CA) and Credential Provider (CP) belong to this category. On the other hand, the decentralized approach allows each entity to request feedback values from other entities in the environment. A consumer entity aggregates all feedback values and further uses these values to derive the total trust value of its provider entity. Some decentralized approaches have been proposed in internet environment such as TrustMe [23], PeerTrust [24], P2PRep [25], and EigenTrust [26]. One major issue with TrustMe, PeerTrust and P2PRep is they broadcast trust request to all peers in the environment for obtaining reputation feedbacks. Thus, it slows down the performance of the entire network.

EigenTrust incorporates both local trust (belief) and global trust (reputation) in its trustworthiness calculations. It uses a normalized principal eigenvector for computing

trust. However, EigenTrust suffers major drawback as it assumes that the honesty of the peers in providing the recommendations are based on the trustworthiness value of these peers in providing the services. Subjective Logic/TNA-SL [27] is another distributed trust mechanism that encompasses 3 degrees (belief, disbelief, and uncertainty) to derive the trustworthiness value of an entity. Its trust model focuses on the operators that represent logic for managing the feedbacks from referrals. REGRET [28] is a reputation system which analyzes the individual, social and ontological dimensions of entities. Several trust models have been proposed in multi-agents system environment, such as Travos [29] and BRS [30]. BRS measures trustworthiness of a provider using bayesian approach. Travos measures the trustworthiness of a provider by probabilistic and beta distribution approach that observe others' opinions and adjust these opinions with buyer's opinions.

4 Issues in Determining the Initial Trust of Mobile Applications

Trust in an electronic network can be divided into two types: direct (personal) trust and third party trust [31]. Direct (personal) trust is a situation where a trusting relationship is nurtured by two entities. This type of trust is formed after these entities have performed transactions with each other. For example, a user inherently trusts a Mobile App after he/she has consumed this Mobile App. On the contrary, third-party trust is a trust relationship of an entity that is formed from the third party recommendations. This means no previous transaction ever occurred between the two interacting entities, i.e. user trusts a Mobile App because this Mobile App is trusted and recommended by other users. We further termed direct trust as *belief* while third-party trust as *reputation* for the rest of this paper.

Belief can be straightforwardly determined due the availability of one's own past experience. However, trust value that is derived from the reputations, which is critical for measuring the trustworthiness of mobile applications, is often harder to compute. This is due to many factors as follows:

1. *Difficulty in finding other users that have consumed the Mobile Apps*: As trust through reputations is heavily relied on third-party (termed as raters) recommendations, there is a need for a user to identify other users (raters) that have downloaded and consumed the Mobile Apps for the purpose of requesting the recommendations. However, finding raters is a challenging task as raters are mostly unknown to the users.
2. *Relativeness perception of different users on the satisfaction levels of Mobile Apps*: The perception of each user on the satisfaction (i.e. quality, security, privacy level, etc.) of a Mobile App varies. For example, a user may rate a Mobile App as good although it has fair performance and it collects user's information. However, other users may rate the same Mobile App as bad.
3. *Dishonest raters in providing rating feedbacks*: It is highly possible that raters are malicious or dishonest in providing rating feedbacks. For example, the seller or developer of a Mobile App may get his friends and families to

give good rating to his application although it has low quality and violates privacy. In this case, the legitimate users may be tricked to believe that such application is good and therefore, they download and consume it.

4. *Several threat strategies subverting rating system*: Literature has presented a number of threat strategies that are used to subvert trust system [24, 26]. One of the most severe threat strategies is providers (i.e. sellers and developers of Mobile Apps) engage in a collaborative agreement to provide good ratings to each other Mobile Apps while give other Mobile Apps bad ratings.
5. *Incentives to rate*: Another challenge in building a successful reputation trust system is in providing the incentives for users to give their rating feedbacks.

Several prominent application stores such as Apple Apps store and Android Google Play suffer from the above issues, in particular issue no. 2-5. From the issues discussed above, it is evident that, in the absence of user own belief, the initial trustworthiness value of a Mobile App that is solely relied on the perceived reputations of others is harder to determine. Nevertheless, such initial trustworthiness is critical to be measured as consumers always tend to select the Mobile Apps that have good level of quality, privacy and security. Further, as discussed in previous section, trust provides the first and extra layer of protection. Thus, in the next section, we attempt to solve the identified issues by proposing our trust solution.

5 The Proposed Trust Model

Considering all issues that were discussed in the previous section, in this section, we present our proposed trust model for measuring the trustworthiness of Mobile Apps. We termed our proposed trust model as **MobilTrust**, a personalized binary trust model with a centralized approach. This personalized trust model takes into account the similarity measurement between the reported reputation values and the perception of the buyer. A thorough discussion on the similarity measurement and trust architecture will be provided later in this section.

For the rest of this paper, we termed the following:

- Mobile App is the mobile application that is available for download and/or consumption from the application stores.
- Buyer is someone that considers whether to download and/or to consume a Mobile App. Buyer will attempt to measure the trustworthiness of a Mobile App prior to download and consumption.
- Rater(s) is other user(s) that provides rating feedback(s) about a Mobile App. Raters are usually the previous buyers and consumers of a Mobile App.
- Rating feedback(s) is the reputation/trustworthiness value(s) of a particular Mobile App that is provided by the rater(s) and buyer. the rating feedback is in a scale of 0 (not trustworthy/not satisfied) – 1 (very trustworthy/very satisfied)

5.1 Classification of the Raters

In MobilTrust, we classify raters into two categories based on buyer's previous interactions with the raters. These categories are further defined as follow:

- *Known Raters*
  When computing the trustworthiness of a Mobile App, a buyer classifies a rater as a known rater under two conditions: (i.) if buyer has previously obtained and used rater's rating feedbacks on other Mobile Apps and (ii.) if buyer has provided his rating feedback on other Mobile Apps which he/she obtained the rater's rating feedbacks from. For example, a buyer previously consumed and provided his rating to a Mobile App *x* in which prior to consuming *x,* he/she obtained the rating feedbacks from rater *A, B*. When the same buyer considers the trustworthiness of another Mobile App *y* and he/she found that rater *A* and *B* have provided their rating feedbacks to *y*, rater *A* and *B* will be considered as the known raters due to their feedbacks on Mobile App *x*.

  As buyer has previously obtained the rating feedbacks from the known raters, buyer would be able to derive the similarity measure of the known raters. The similarity measurement will be discussed in the next section.
- *Unknown Raters*
  A rater is classified as unknown rater if the buyer has not obtained any previous rating feedback from this rater. Therefore, the buyer is not able to measure the similarity with this rater.

The classification of the raters plays a pivotal role in measuring the trustworthiness of a Mobile App in our proposed trust model. Such classification allows more precise trustworthiness measurement as it takes into account the differentiation between the raters with whom buyer has the experience and the new raters with whom buyer has no experience at all.

5.2 Introducing Similarity Measurement on the Rating Feedbacks

In order to measure the honesty and the perception similarity of each rater's rating feedback, we introduce the measurement of *similarity*. Fundamentally, similarity is the combination between honesty and perception of the rater's rating feedback, as depicted in (1). Honesty is about measuring the credibility of rater's rating feedback in telling truth opinion, while perception is about measuring the relativeness of opinions between rater's rating feedback and buyer's perception. Both honesty and perception of rater's rating feedback, or known as similarity value, are measured from previous rater's feedbacks on other Mobil Apps. Similarity value is important to be measured as it is possible that a rater acts malicious by providing dishonest feedbacks about the trustworthiness of a mobile application. It is important to note that various raters may have different similarity values that reflect their honesty and relativeness perception in providing the rating feedback.

$$\text{SIMILIARITY= HONESTY + PERCEPTION} \quad (1)$$

How does the similarity value of raters' rating feedbacks is assigned or measured? MobilTrust assigns the similarity value to each rater after buyer downloads and consumes a Mobile App. This is done by reviewing each rater's rating feedback with the validity of transaction and perception rating that is experienced by the buyer. Essentially, after buyer downloads and consumes a Mobile App, he/she will give his rating feedback about the trustworthiness of this Mobile App. MobilTrust then measures the compatibility between each rater's rating feedback and buyer's rating feedback. We further introduce the *SimilarityRange* to assign the similarity value for each rater's rating feedback. Any rater's rating feedback that is between the *SimilarityRange* is considered as compatible with buyer's rating feedback while the rater's rating feedback that is not between the *SimilarityRange* is considered as not compatible. MobilTrust assigns 1 (similar) as the similarity value for those rater's rating feedbacks that are within the *SimilarityRange*, and it assigns 0 (dissimilar) as the similarity value for those rater's rating feedbacks that are outside of *SimilarityRange*. Each similarity value will be added to the *TotalSimilarity* field in the central database and the number of past feedback (*TotalPastFeedback* field) will be increased. Algorithm 2 in sub-section 5.4 further details this process and section 6 provides detail on the implementation.

From all similarity values that a buyer assigned to each rater in the past, MobilTrust derives the average similarity value of each rater that will be used for the initial trust computation of new Mobile App. We derive the average similarity value by dividing the *TotalSimilarity* field and *TotalPastFeedback* field of each rater that are obtained from the database as shown in (2).

$$Sim(i) = \frac{TotalSimilarity(i)}{TotalPastFeedback(i)} \quad (2)$$

Let *i* denote the rater who provides the rating feedback on a new Mobile App, *Sim(i)* denote the average similarity value of rater *i*. *TotalSimilarity(i)* denote the total similarity value of past rating feedbacks given by rater *i* on other Mobile Apps. *TotalPastFeedback(i)* denote total number of past rating feedbacks given by rater *i*.

Note that, the average similarity value can only be computed for the known raters as buyer has previously obtained their rating feedbacks from other Mobile Apps. For the unknown raters, due to non-availability of previous rating feedbacks, MobilTrust assigns 0.5 (neither similar nor dissimilar) as their average similarity values.

5.3 Computing Mobile Apps Trustworthiness

Once the average similarity value of each rater is computed, the trustworthiness of a Mobile App from buyer's perspective will be derived. MobilTrust computes the trustworthiness of a Mobile App based on (3).

$$Trust(x) = \sum_{1}^{i} \left(\frac{Sim(i)}{\sum_{1}^{i} Sim(i)} * RF(i) \right) \quad (3)$$

Let *Trust(x)* is the trustworthiness value of a Mobile App *x* that is computed from buyer's perspective. *Sim*(*i*) is the average similarity value of rater *i*. *RF(i)* is the rating feedback that is given by rater *i* on Mobile App *x*.

In order to increase the accuracy of trustworthiness computation, we utilize the exogenous approach [32] in MobilTrust. That is, the rating feedbacks which average similarity value does not meet a particular threshold (*SimilarityThreshold)* will not be counted in the trustworthiness computation. We further set the *SimilarityThreshold* value to 0.5 (neither similar nor dissimilar) such that the rating feedback which average similarity value is below such threshold is discarded. Algorithm 1 further shows the procedures in computing the trustworthiness of a Mobile App.

Algorithm 1. Computing Trustworthiness of a Mobile App

```
Input:
R = a set of all raters that provided rating feedbacks.
Sim = a set of average similarity values of the raters.
Temp = temporary variable array to hold all raters that will be
included in trust computation.
TotalAvgSim = total average similarity values that are included
in computation.
Output: Trust(x).
Algorithm:
TotalAvgSim = Null;
for i = 1 to Length(R) do
  Retrieve Sim(i) from database;
  if Sim(i) ≥ SimilarityThreshold  then
       Temp ← i;
       TotalAvgSim += Sim(i);
  end if
end for
if Length(Temp) > 0 then
  for j=1 to Length(Temp) do
       Retrieve Sim(j) and RF(j) from database;
       Compute (3) using Sim(j), RF(j) and TotalAvgSim;
  end for
end if
```

The computed trustworthiness value of a Mobile App (*Trust(x)*) ranges from 0 (not trustworthy) to 1 (very trustworthy). The range of MobilTrust trustworthiness value can be easily adapted to the 5-star rating that is commonly used in the application

stores. For example, using the step of 0.2, the results could be: 0 trust value means no star, 0.2 trust value means 1 star, 0.4 trust value means 2 stars, and so on.

5.4 The Learning Algorithm for Assigning Similarity Value

Once a buyer provides his rating feedback on a Mobile App, MobilTrust will automatically assign the similarity value to the raters. As discussed in previous sub-section 5.2, MobilTrust assigns the similarity value to the raters by evaluating whether their rating feedbacks are within the *SimilarityRange* of buyer's rating feedback. It assigns either 0 (similar) or 1 (dissimilar) based on the inclusivity.

Algorithm 2. Learning Algorithm for Assigning Similarity Value

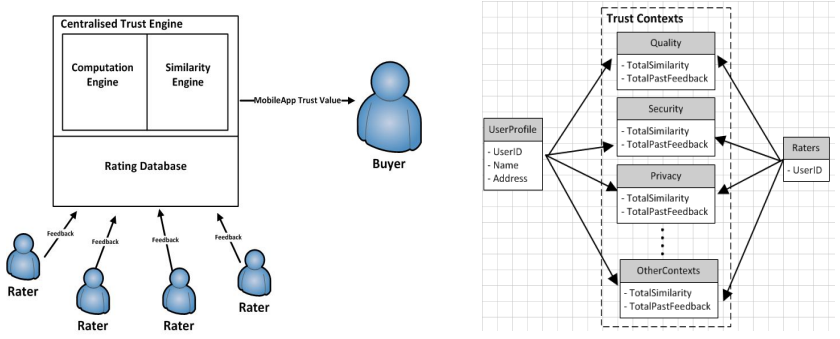
```
Input:
R = a set of all raters that provided rating feedbacks.
RF = the rating feedback obtained from R.
RF(buyer) = the rating feedback obtained from the buyer.
TotalSim = a set of total similarity value of R.
TotalPastFeedback = a set of total number of R past feedbacks.
Algorithm:
Retrieve RF(buyer) from database;
for i = 1 to Length(R) do
  Retrieve RF_i from the database;
  if RF_i ≤ (RF(buyer) + SimilarityRange) and RF_i ≥ (RF(buyer) –
  SimilarityRange)  then
    TotalSim_i += 1;
    TotalPastFeedback_i += 1;
  else
    TotalSim_i += 0;
    TotalPastFeedback_i += 1;
  end if
end for
```

Note that both *TotalSim* and *TotalPastFeedback* from Algorithm 2 are subjective for each buyer and they are stored in the central database (will be detailed in next section).

This learning algorithm is crucial for determining the similarity value of each rater based on buyer's perspective. Further, this learning algorithm also serves as incentives for buyers to keep providing their rating feedbacks on the Mobile Apps that they have downloaded and consumed. Buyers that do not provide the rating feedbacks will be disadvantaged as they are not able to derive the similarity value of the raters for the subsequent Mobile Apps download.

6 Implementation Strategies

As discussed briefly in the section 5, we propose the use of centralized mechanism in MobilTrust for both trust computation and rating databases. The centralized approach is selected due to its simplicity and also its appropriateness to the current mobile applications architecture, in which mobile applications are hosted and distributed by the central application stores. The centralized trust architecture is composed of two main components: the rating database and the centralized trust engine, as depicted in figure 1a. The rating database stores the rating feedbacks for all listed Mobile Apps in the application stores as well as the similarity values of all raters and buyers. Raters (or buyers after they downloaded the Mobile Apps) provide their rating feedbacks to the rating database. The centralized trust engine consists of (i.) computation engine for computing the trustworthiness value of a Mobile App (sub-section 5.3) and (ii.) similarity engine for assigning the similarity value (sub-section 5.4).



(a) (b)

Fig. 1. MobilTrust Implementation: (a) centralized trust infrastructure (b) buyer-rater trust context relationships

In order to identify each user in MobilTrust, we leverage the use of user ID which has been used in several application stores, such as Apple ID [33] and Google Account [34]. In MobilTrust, each user is given a unique user ID (in UserProfile database table) for downloading and/or rating Mobile Apps. Such user ID becomes an identifier for each user in MobilTrust. Note that, in future implementation, this user ID can be in form of user accounts in the respective application stores.

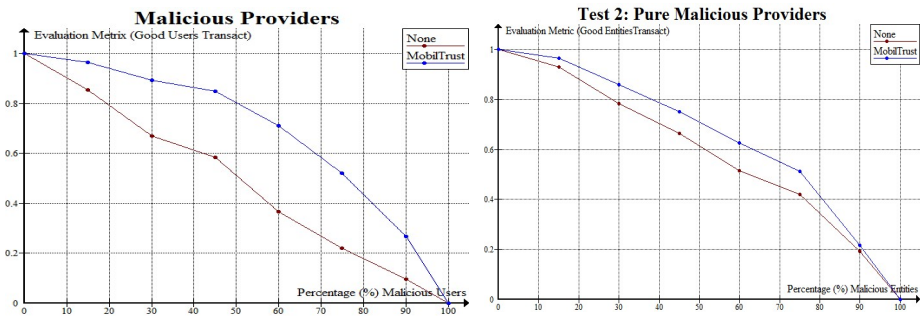
The relationships of buyers and raters in the rating database are depicted in figure 1b. For facilitating the personalized trust computation, each buyer has a list of the raters with whom he/she has obtained the rating feedbacks from. For each rater that is associated with the buyer, buyer has the *TotalSimilarity* and *TotalPastFeedback* for computing the average similarity of rater based on a number of pre-defined trust contexts. Trust contexts (e.g. quality, security, privacy, etc.) allow more expressiveness in measuring the trustworthiness of a Mobile App.

7 Simulation Results

We performed two preliminary simulations in RM simulator [16] to measure the effectiveness of MobilTrust. In such simulations, we considered a typical Mobile Apps environment in which user can consume and produce Mobile Apps. Our simulation environment consisted of 100 users and run over 1000 downloads for each test cycle. There are 50 Mobile Apps simulated in the environment, and each Mobile App can be offered by more than one user. This is to simulate the real Mobile Apps environment in which several providers may offer similar mobile apps. For the purpose of collecting the statistics, we modified the *SimilarityRange* in algorithm 2 such that rater whose similarity rating (*RF*) is higher than 0.5 while buyer's similarity rating (RF_{buyer}) is positive was considered as similar (Thus, similarity rating of 1 will be given), and vice versa. In each simulation, we collected statistics from 5 test cycles and averaged the results. We were particularly concern on the valid downloads performed by the "good" users. The collected statistics are assessed in the following evaluation metric:

$$Metric: \frac{\#\ of\ valid\ Mobile\ Apps\ downloaded\ by\ "good"\ entities}{\#\ of\ transactons\ performed\ by\ "good"\ \ entities}$$

In the first simulation, we filled our simulation environment with a number of malicious providers (i.e. provide invalid Mobile Apps but always provide credible feedbacks). For each step of 15%, we ran the simulation and obtained the statistics as shown in figure 2a. The results show that MobilTrust has effectively reduced the number of invalid download performed by "good" users when compared with no trust model in the environment. This further demonstrates the success of MobilTrust algorithms to reduce the invalid downloads.



(a) (b)

Fig. 2. MobilTrust Evaluation: (a) malicious users (b) purely malicious users

In the second simulation, we filled our simulation environment with purely malicious users (i.e. consistently provide invalid Mobile Apps and non-credible feedbacks). The purely malicious users give a significant threat to the Mobile Apps

environment. The non-credible feedbacks that they provide may reduce the credibility of "good" Mobile Apps while improving the credibility of "bad" Mobile Apps. For each step of 15%, we ran the simulation and obtained the statistics as shown in figure 2b. The results show that MobilTrust has successfully increased the number of valid Mobil Apps download for the "good" users.

8 Conclusion

This paper has reviewed the current state of art in mobile applications trustworthiness measurement. Further, it shows the importance of trust as the first and extra layer of protection in mobile applications environment. Determining the initial trustworthiness of mobile applications is challenging due to several issues such as finding the raters, different perceptions, dishonest rating feedbacks, several threat strategies and also the unavailability of incentives. This paper further provides a unique trust model, termed as MobilTrust, for solving the identified trust measurement issues. An important feature of MobilTrust is the similarity value that measures the honesty of raters in providing feedbacks and the similarity perceptions between raters and buyer. The trustworthiness of a mobile application is computed based on the average similarity value of the raters and also the raters' rating feedbacks. Several trust formulas and algorithms have been introduced to measure the trustworthiness of mobile applications and also to learn and update the average similarity value of the raters. These trust formulas and algorithms also measure the credibility of each rater and are used to mitigate the treat strategies. In addition, the learning algorithm provides incentives for buyers to provide their rating feedbacks. This paper also introduces the centralized implementation strategy for MobilTrust. Future work will be focusing on the evaluation of trust formulas and algorithms in reducing the invalid transactions and also its effectiveness against other threat strategies.

References

1. International Telecommunication Union (ITU), ITU estimates two billion people online by end 2010, Access to mobile networks available to over 90% of world population 143 countries offer 3G services, Press Release Report (2010), viewed at http://www.itu.int/net/pressoffice/press_releases/2010/39.aspx
2. PRB, 2010 World Popilation Data Sheet (2010), http://www.prb.org/publications/datasheets/2010/2010wpds.aspx
3. IDC Research, Worldwide Smartphone 2012-2016 Forecast and Analysis, Research report (2012), http://marketresearch.com.
4. Apple, Apple Application Store (2012), http://itunes.apple.com/us/genre/ios/id36?mt=8
5. Google Play, Android Google Play (2012), https://play.google.com/store?hl=en

6. BusinessInsider, Number of Apps Available at Smartphones' Apps Stores (2011), viewed at http://articles.businessinsider.com/2011-03-09/tech/30011803_1_app-store-google-s-android-market-twitter
7. Grandison, T., Sloman, M.: A Survey of Trust in Internet Applications [IEEE Communications Surveys and Tutorials, Fourth Quarter] (2000), http://www.comsoc.org/pubs/surveys/
8. Josang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provisioning. Decision Support System 43, 618–644 (2007)
9. Osborne, C.: IOS Apps: Massive invasion of user privacy, ZDNet news (2012), http://www.zdnet.com/blog/igeneration/ios-apps-massive-invasion-of-user-privacy/15138
10. Lowenshon, J.: Congress probing iOS developers on user privacy, address books, CNet news (2012), http://news.cnet.com/8301-27076_3-57402957-248/congress-probing-ios-developers-on-user-privacy-address-books/
11. Enck, W., Gilbert, P., Chun, B.-G.: Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones (2010)
12. Smith, E.: iPhone applications & privacy issues: An analysis of application transmission of iPhone unique device identifiers (UDIDs), http://www.kompatscher.biz/phocadownload/iPhone-Applications-Privacy-Issues.pdf
13. Ferro, G.: BYOD Policies vs. the Realities of Corporate IT, NetworkComputing.com, http://www.networkcomputing.com/wireless/240000916
14. CoreGrid, D.IA.03 Survey Material on Trust and Security, European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies, Technical Paper (2004)
15. Rasmusson, L., Janssen, S.: Simulated Social Control for Secure Internet Commerce. In: Proceedings of the 1996 New Security Paradigms Workshop, Lake Arrowhead, CA, USA (1996)
16. University of Pennsylvania, TM/RM Simulator (March 2012), http://rtg.cis.upenn.edu/qtm/p2psim.php3
17. Benats, G., Bandara, A., Yu, Y., Colin, J., Nuseibeh, B.: PrimAndroid: Privacy Policy Modelling and Analysis for Android Applications. Presented at 2011 IEEE International Symposium on Policies for Distributed Systems and Networks, Pisa, Italy (2011)
18. Dellarocas, C.: Immunizing Online Reputation Reporting Systems against Unfair Ratings and Discriminatory Behavior. In: The Proceedings of Second ACM Conf. Electronic Commerce (2000)
19. Cydia Market, http://cydia.saurik.com/
20. Kerschbaum, F., Haller, J., Karabulut, Y., Robinson, P.: PathTrust: A Trust-Based Reputation Service for Virtual Organization Formation. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 193–205. Springer, Heidelberg (2006)
21. Ion, M., Danzi, A., Koshutanski, H., Telesca, L.: A Peer-to-Peer Multidimensional Trust Model for Digital Ecosystems. Presented at the Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008), Phitsanulok, Thailand (2008)
22. Schmidt, S., Steele, R., Dillon, T.: DEco Arch: Trust and Reputation Aware Service Brokering Architecture in Digital Ecosystems. Presented at the Inaugural IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST), Cairns, Australia (2007)

23. Singh, A., Liu, L.: TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems. Presented at the Third International Conference on Peer-to-Peer Computing, Sweden (2003)
24. Xiong, L., Liu, L.: Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Transactions on Knowledge and Data Engineering 16, 843–857 (2004)
25. Damiani, E., Vimercati, S.: Managing and Sharing Servents' Reputations in P2P Systems. IEEE Transactions on Knowledge and Data Engineering 15, 840–854 (2003)
26. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The EigenTrust Algorithm for Reputation Management in P2P Networks. Presented at the 12th ACM International Conference on World Wide Web, USA (2003)
27. Jøsang, A., Hayward, R., Pope, S.: Trust Network Analysis with Subjective Logic. In: Proceedings of the 29th Australasian Computer Science Conference (2006)
28. Sabater, J., Sierra, C.: REGRET: A reputation model for gregarious societies. In: Proceedings of the Fifth International Conference on Autonomous Agents, Montreal, Canada (2001)
29. Teacy, W.T.L., Patel, J., Jennings, N.R., Luck, M.: Travos: Trust and reputation in the context of inaccurate information sources. Journal of Autonomous Agents and Multi-Agent Systems 12 (2006)
30. Josang, A., Ismail, R.: The Beta Reputation System. In: Proceedings of the 15th Bled Electronic Commerce Conference (2002)
31. Entrust: The concept of trust in network security, Version 1.2 [White Paper] (2000, April 2011), http://www.entrust.com/resources/pdf/trust.pdf
32. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems (2005) (to appear)
33. Apple, What's an Apple ID?, https://appleid.apple.com/cgi-bin/WebObjects/MyAppleId.woa/
34. Google, Google Accounts, http://www.google.com/intl/en/landing/accounts/index.html#utm_campaign=en&utm_medium=et&utm_source=gaia