Secure Candy Castle — A Prototype for Privacy-Aware mHealth Apps

Christoph Stach†
University of Stuttgart, Institute for Parallel and Distributed Systems
Universitätsstraße 38, 70569 Stuttgart, Germany
Email: Christoph.Stach@ipvs.uni-stuttgart.de

***Abstract*—Due to rising medical costs, the healthcare landscape is on the move. Novel treatment methods are badly required. Especially for the treatment of chronic diseases the usage of smart devices in combination with medical devices for telemedical screenings is a promising approach. If the patients are not in control of the collection and processing of their health data, privacy concerns limit their willingness to use such a method. In this paper, we present a prototype for an Android-based privacy-aware health game for children suffering from diabetes called *Secure Candy Castle*. In the game, the player keeps an electronic diabetes diary in a playful manner. In doing this, s/he is supported by various sensors. His or her data is analyzed and in case of a critical health condition, the game notifies authorized persons. With our approach, the user stays in control over his or her data, i. e., s/he defines which data should be shared with the game, how accurate this data should be, and even how the data is processed by the game. For this purpose, we apply the *Privacy Management Platform*, a fine-grained and extendable permission system.**

***Index Terms*—mHealth; privacy; diagnostic game; diabetes.**

I. Introduction

An efficient healthcare system is probably the greatest good for civil society. However, due to an aging population and rising numbers of chronic diseases such as diabetes, the healthcare system is already working at its limit. Thus, the application of new health information technologies gets more and more accepted by both, healthcare consumers and providers. The so-called *mHealth*—i. e., the usage of smart devices connected to medical devices—is highly beneficial for diagnosis, screening, and therapy of various diseases. Especially for the treatment of chronic diseases where hitherto an episodic care in a clinic is required, mHealth enables a telemedical caregiving to relieve physicians. Moreover, mHealth apps can be tailored to certain target groups [1]. Studies show, that health games are well-suited to teach young patients knowledge about their disease and integrate required therapeutic procedures into the patients' daily routine [2]. When dealing with health data, privacy should be mandatory—yet, this key problem is still unresolved [3].

Therefore, we look at *Candy Castle* an existing mHealth game for diabetic child [4]. Up to now this game does not consider privacy issues. Thus, we implement multiple privacy features in Candy Castle. The enhanced game is called ***S**ecure **C**andy **C**astle* (*SCC*). The realization of the privacy features is based on the ***P**rivacy **M**anagement **P**latform* (*PMP*) [5]. The developed concepts and components are generic and thus applicable to any mHealth app.

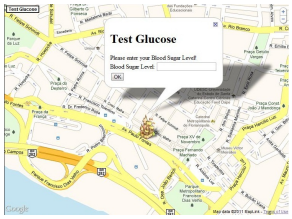
† This work was supported by a Google Research Award.

The remainder of this paper is as follows: In Section II we discuss existing health games and auxiliary systems for mHealth apps. Then we introduce our own preliminary studies concerning health games and privacy in Section III, namely Candy Castle and the PMP. Section IV describes based on these insights, how we enhance Candy Castle by regarding privacy issues. Finally, Section V presents the demonstration scenario in which the audience experiences SCC.

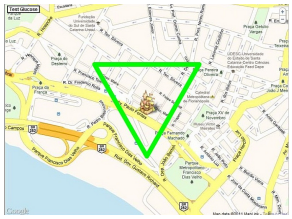
II. Related Work

There are basically two types of mHealth games: *educational games* and *diagnostic games*. The former game type's aim is to inform the patients about the characteristics of their disease, potential risk factors, or a healthy life style. An example for this kind of games is *Power Defense* [6]. In this game, adolescents learn the importance of a correct calculation and interpretation of diabetes-relevant health values. In general, educational games handle no sensitive data, as they are only a kind of interactive teaching book. Therefore, privacy in not an issue for these games. *mySugr* [7] is a diabetes diary for adolescents. This is a perfect example of a diagnostic game, since patients enter various health data (e. g., blood sugar values or bread exchange units) and the app performs analyses and processes the results. This generates a lot of private data that needs to be secured. mySugr faces this issue simply by introducing user accounts. Thereby the data is secured against unauthorized accesses by other users but the app itself has unrestricted access to the data and the user cannot track how his or her data is processed—i. e., privacy is at stake. In general, almost any so-called *mHealth* app lacks of a satisfying privacy management [3].

An approach to face this problem is the use of cloud-based services for the storage, processing, and screening of health data. The *Microsoft HealthVault* [8] enables the secure storage of a personal electronic health record. The system acts as a man in the middle between a data producer (i. e., an mHealth app) and a data consumer (i. e., an analytic system). Both, the producer as well as the consumer still have to assure privacy on their own. *Google Fit* [9] is a similar system. In addition to the storage of health-related data, it offers a simplified integration of *Google Wear* devices in a secure way. The *ECHO* platform [10] is a holistic cloud-based health data management system, as it provides not only data integration and storage features but also



(a) Blood Sugar Metering



(b) Protective Walls

Figure 1. Screenshots of Candy Castle [4]

processing and screening functionalities. However, apps using such a system still have to gather health data, and therefore privacy protection measures are required for these apps. Sorber et al. suggest a hub and spoke architecture with a central unit called *Amulet* [11]. The Amulet collects health data from medical devices and provides this data in a privacy-aware manner for mHealth apps. Weerasinghe et al. enhance the hub and spoke approach by encrypting any data and only the central unit has the key [12]. Thus, linked apps only get access to data which is meant for them. However, Amulet is just a conceptual model and provides no implementation. For the realization of our diagnostic game, we apply the Privacy Management Platform which supports a similar hub and spoke architecture (see Section III-B). For the data analysis part, we rely on a system similar to ECHO.

III. Preliminary Studies

This work is based on two preliminary studies, namely Candy Castle and the Privacy Management Platform, which are both discussed in the following briefly.

A. Candy Castle

Martin Knöll initially introduced the idea of a diagnostic game for children suffering from diabetes called *Candy Castle* [2]. Together with young patients he created a game concept in which the player (i. e., the patient) is motivated to check his or her blood sugar level regularly. Moreover, the game visualizes the patient's condition in a manner appropriate for children. However, not only the patients should profit from Candy Castle but also their physicians. Usually, patients have to keep their health data in a handwritten diabetes diary—errors are inevitable. Therefore, Candy Castle creates an electronic diabetes diary in which the health data is transferred accurately. Physicians are able to analyze the entries in this electronic diabetes diary. Additionally, each health data entry is associated to a certain location, as physicians are interested

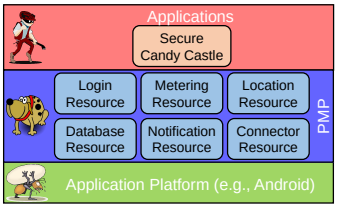


Figure 2. Embedding of the PMP in an Application Platform

in correlations between a patient's condition and his or her current whereabouts.

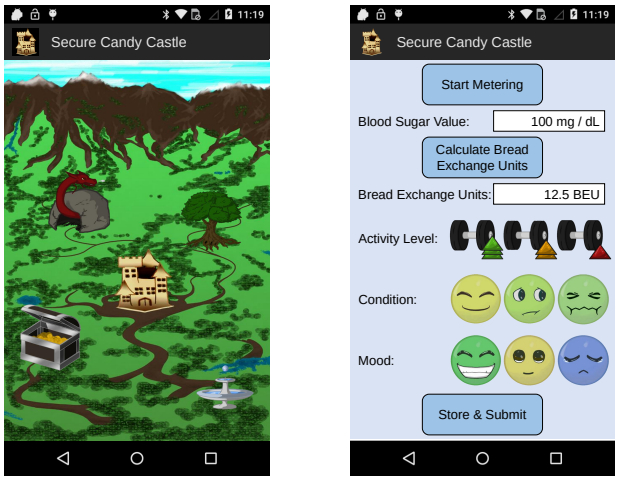
The game idea was implemented as a web app optimized for mobile devices [4]. The player "builds" his or her castle on a real world map at his or her current location by entering an initial blood sugar value (see Figure 1a). Since dark forces attack the castle at regular intervals, the player has to put up protective walls in his or her surrounding by entering additional blood sugar values (see Figure 1b). Each attack damages the walls whereby the player repeatedly has to remember to check his or her blood sugar level.

B. The Privacy Management Platform

The ***P**rivacy **M**anagement **P**latform* (*PMP*) is a fine-grained and extendable permission system for application platforms [5]. The PMP is an intermediate layer separating potentially dangerous apps form the OS (see Figure 2). In this way, apps cannot access system functions and data directly, but they have to ask the PMP for permission. This is realized by introducing interfaces called *Resources*. Each Resource provides access to a certain functionality or type of data (e. g., the Location Resource provides access to data related to the current location). When an app requests location data, the PMP informs the user and s/he can grant or deny the request. With this in mind, each Resource defines *Privacy Settings*. Via the Privacy Setting a user can grant a request with constraints—e. g., for the Location Resource there is an accuracy setting to reduce the accuracy of the location data. Additionally, s/he can instruct the Resource to pass solely faked data. Further Resources can be added at runtime need-based. E. g., the Metering Resource manages the access to external medical devices such as glucometers. This Resource is required only, when such a medical device is available. The Resources which are relevant for this paper are depicted in Figure 2 and discussed in Section IV. In order to use the PMP, apps have to encapsulate their functionalities within so-called *Service Features*. The PMP enables a user to disable certain Service Features whereby not only the app's functionality but also its data access is reduced. To reduce the configuration effort for the ultimate users, the PMP allows to apply presets provided by trusted third parties. For more information about the PMP, please refer to the literature [5], [13].

IV. Secure Candy Castle

As the concept of Candy Castle is accepted by both, patients and physicians (see [2]), we attend to the privacy issues of which every diagnostic game suffers from [3]. On that account,



(a) Main Screen (b) Questionnaire

Figure 3. Overview of the Secure Candy Castle's Key Functionalities

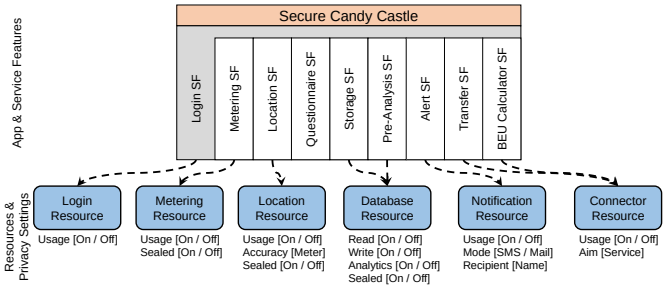


Figure 4. Mapping of SCC's Service Features to PMP Resources

we split the app in two parts, namely a mobile front-end for the actual game as well as the data acquisition and a cloud-based back-end for the data pooling as well as thorough medical analyses. As the latter is already well studied (e. g., [10]), we concentrate on a privacy-aware approach for the former. For that purpose, the ***S**ecure **C**andy **C**astle* (*SCC*) for Android is based on the PMP.

As an initial step, we replace the real world scenario in which the player's castle is positioned in his or her surrounding by a fantasy scenario, as shown in Figure 3a (each icon stands for a measurement). This serves a dual purpose: On the one hand, such a fantasy world is better suited for the mainly young target audience. On the other hand, third parties cannot see where the player has taken the measurements—and therewith detect his or her regular whereabouts—when they are able to take a look at the game. Internally, each spot of the fantasy map is mapped to a real world location whereby correlations between measured data and location can still be studied by physicians.

We encapsulate the functionalities in nine separated Service Features which can be turned on and off individually (see Figure 4). The *Login SF* ascertains the identity of the user. Therefore, it uses the *Login Resource* which provides a login form and performs the authentication. In this way, the sensitive login data is only known to the PMP and not to SCC. As the Login Resource can be used by any app, the user only needs a single password for all of his or her apps (cf. *OAuth*). The *Metering SF* comprises any task related to the capturing of health data. In order to do this, SCC does not have to deal with the connection to external medical devices and their communication protocols. It simply has to request the data from the *Metering Resource* which is able to connect to various medical devices. Due to security reasons, heath data cannot be altered intentionally as faked data leads to false diagnostic results. However, the PMP offers a *Seal* for sensitive data. I. e., the PMP encrypts this data before sending it to the inquiring app. Thereby only trusted Resources can process the data and the app has no access to it (cf. [12]). The *Location SF* adds location information to the health data. It gets the location data via the *Location Resource* which requests this data from the most accurate location provider currently available. Nevertheless, the user can reduce the accuracy or even use faked locations. The *Storage SF* caches the health date on the device in case of transmission problems. The *Pre-Analysis SF* performs analyses on these cached data, e. g., to detect critical conditions preterm. Both Service Features use therefore the *Database Resource*. This Resource provides an encrypted data container which can be accessed in a SQL-like manner. When the Pre-Analysis SF detects a critical condition, the *Alert SF* notifies authorized persons (e. g., the patient's parents). This is realized via the *Notification Resource*. This Resource supports various communication methods (e. g., SMS or mail) and the user specifies which method has to be used for which app. Moreover, s/he defines which persons can be contacted by an app. The *Transfer SF* sends the health data to a back-end for further analyses. As SCC has no permission to access the Internet, it has to send the data via the *Connector Resource*. Thereby, it can send the data only to a user-defined aim and cannot stealthily pass the data to untrusted third parties. User studies indicate that in addition to the blood sugar values, data such as the patient's current activity level, condition, or mood should be added to the diabetes diary similar to the mySugr app (realized in the *Questionnaire SF*). Furthermore, we add a bread exchange units calculator to SCC similar to the one of the Power Defense app (realized in the *BEU Calculator SF*). A capture of these auxiliary services is given in Figure 3b. Among others, the BEU Calculator queries external services for a certain food product's BEU value. Since especially unpacked food does not provide this information, the BEU Calculator is useful tool for patients.

By using the PMP, SCC has no Android permission at all and therefore can only process the sensitive data via the regulated and secured interfaces of the Resources. Thus, the game's mobile front-end is completely secured concerning any privacy issues. Additionally, as the Resources are generic and can be used by any app, this approach is also a huge contribution towards the interoperability of any mHealth app.

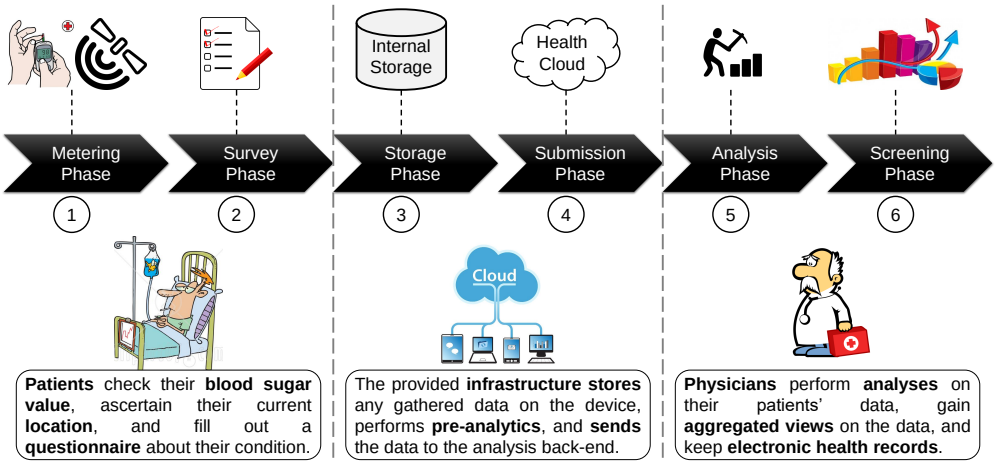


Figure 5. Phases of the Demonstration Scenario for Secure Candy Castle

V. Demonstration Scenario

In our demonstration scenario, we cover every aspect of SCC. In our hands-on demonstration the audience slips into the role of both, patients and physicians. At any stage, the audience is able to modify the Privacy Settings and witness the impact on SCC. That way the audience gets a feeling of how our approach assures the patient's privacy throughout the whole game. Figure 5 displays the presented six phases:

In Phase (1) after logging in, the patient performs a metering of his or her blood sugar value and current location. Due to sanitary reasons, we do not use a real glucometer in the demonstration session as this requires the patient's blood for the metering. Instead we simulate the glucometer with a second Android smart device. Subsequently in Phase (2), s/he completes the questionnaire by describing his or her current activity level, condition, and mood. Also the bread exchange units calculation can be tested. For this purpose, we use a self-made web service which provides BEU information for an exemplary selection of foods.

Phase (3) and Phase (4) are executed automatically and only the effects can be witnessed. Initially, the patient's data is cached in the smart device's internal memory. In the process, the data is pre-analyzed whereby outliers caused by measuring error or critical conditions can be detected at an early stage and corrective actions can be executed (e. g., ask the patient to repeat the metering or send an alert to authorized persons). Whenever a connection to the back-end is available, any new health data entry is sent to it.

Finally, the audience slips into the role of a physician. The back-end processes the patients' data according to rules defined by the physician in Phase (5). Thereby, s/he only gets an aggregated view on relevant data. However, s/he can look into details whenever it is necessary. For this data, the physician has access to various graphic renditions in Phase (6). Additionally, s/he maintains electronic health records with our system, as the back-end can be used by several mHealth apps for different diseases. Keep in mind, that Phase (5) and Phase (6) is out of this paper's scope. Therefore, the realization is only a proof of concept. For a comprehensive analytics mHealth back-end, see [10].

Acknowledgments

The PMP results from a close collaboration with Google Munich office. Hence, we would like to thank Google for their support and their suggestions for improvements.

Furthermore, we thank our student Corinna Giebler for assistance with the realization of the SCC demonstrator and her comments that greatly improved this work.

References

[1] D. Siewiorek, "Generation Smartphone," *Spectrum, IEEE*, vol. 49, no. 9, pp. 54–58, 2012.

[2] M. Knöll, ""On the Top of High Towers..." Discussing Locations in a Mobile Health Game for Diabetics," in *IADIS GET '10*, 2010.

[3] Y. Bai *et al.*, "Issues and Challenges in Securing eHealth Systems," *Int. J. E-Health Med. Commun.*, vol. 5, no. 1, pp. 1–19, 2014.

[4] C. Stach and L. F. M. Schlindwein, "Candy Castle — A Prototype for Pervasive Health Games," in *PerCom '12*, 2012.

[5] C. Stach and B. Mitschang, "Privacy Management for Mobile Platforms — A Review of Concepts and Approaches," in *MDM '13*, 2013.

[6] E. Bassilious *et al.*, "Power Defense: A Serious Game for Improving Diabetes Numeracy," in *CHI '12*, 2012.

[7] D. Payne, "MySugr," *Nursing Standard*, vol. 29, no. 33, pp. 31–31, 2015.

[8] V. Bhandari, *Enabling Programmable Self with HealthVault*. O'Reilly Media, Inc., 2012.

[9] S. M. Mishra, *Wearable Android: Android Wear and Google FIT App Development*. Wiley Online Library, 2015.

[10] M. Bitsaki *et al.*, "An Integrated mHealth Solution for Enhancing Patients' Health Online," in *MBEC '14*, 2014.

[11] J. Sorber *et al.*, "An Amulet for Trustworthy Wearable mHealth," in *HotMobile '12*, 2012.

[12] D. Weerasinghe *et al.*, "Device Data Protection in Mobile Healthcare Applications," in *Electronic Healthcare*, ser. LNICST, vol. 1, Springer, 2009, pp. 82–89.

[13] C. Stach and B. Mitschang, "Design and Implementation of the Privacy Management Platform," in *MDM '14*, 2014.