PIggy-backed Key Exchange using Online Services (PIKE)

Wolfgang Apolinarski, Marcus Handte, Muhammad Umer Iqbal, Pedro José Marrón
Networked Embedded Systems
Universitaet Duisburg-Essen
Duisburg, Germany
firstname.lastname@uni-due.de

***Abstract*—This demonstration presents PIKE, a piggy-backed key exchange protocol that uses social networks (like Facebook) or business tools (like Google Calendar) to enable secure personal interaction. PIKE minimizes the configuration effort that is necessary to set up a secure communication channel among a set of devices. To do this, it piggybacks the exchange of cryptographic keys on existing online services which perform user authentication and enable the (secure) sharing of resources. To support encryption or authentication without Internet connection, PIKE relies on the automatic detection of triggers for upcoming personal interactions and exchanges keys before they take place. To demonstrate the broad applicability of PIKE, we present two example applications that show how its secure key exchange can be used in the real world. The first application uses PIKE to automatically share resources – in our case the readings of a GPS receiver – among a set of devices, when an event takes place. The second application relies on PIKE to enable the secure and automatic identification of individual visitors at the registration desk of a conference.**

***Keywords*-Key-exchange, online services, smart phones**

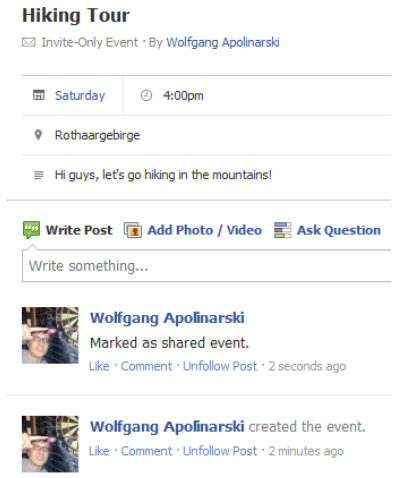

Figure 1. Shared Trigger Resource in Facebook

I. Introduction

Online services such as Facebook or Google Calendar have become an important and ubiquitous mediator of many human interactions. In the virtual world, they enable secure remote interaction by supporting restricted sharing of resources such as documents, photos or calendars between different users. Users are typically identified with a unique identifier and they authenticate themselves by means of passwords or similar mechanisms. The shared resources can then be tied to different sets of identifiers such as friend lists in Facebook or individual users in Google Calendar. To control the access to resources, the services use encrypted communication such as TLS and they require authentication upon resource access.

The success of online services indicates that this mediation model can effectively support secure remote interaction. However, when they are used to support personal interactions, issues arising from a remote connection such as higher response times or intermittent connectivity cannot be hidden by caching and synchronization. To mitigate this, it is possible to support personal interactions by means of local communication. Yet, in order to provide a similar level of security, this requires encryption and authentication mechanisms that must be configured. To avoid this problem, we have designed PIKE, a key-exchange protocol that aims at seamlessly extending the support provided by online services to enable non-mediated personal interaction.

The basic idea is to proactively piggyback the exchange of keys on top of the existing service infrastructure in such a way that neither manual configuration nor Internet connectivity is needed. Compared to existing approaches, PIKE exhibits three main advantages: First, in contrast to manual configuration [1], [2], it can easily scale to hundreds of interaction partners. Second, when compared to proximity-based key exchange [3], [4], it provides user-level as opposed to device-level authentication. Third, due to its proactivity, it does not require any Internet connectivity during the time the interaction takes place, which is needed by similar approaches [5], [6].

II. PIKE

In the following, we briefly outline PIKE. A more thorough description including an experimental evaluation can be found in [7]. PIKE uses shared resources stored in online services to trigger and perform a secure key exchange. An example for such a resource could be an event on Facebook (see Figure 1), an appointment in Google Calendar or any other resource that can be tied to a certain date and time.

978-1-4244-9529-0/13/$31.00 ©2013 IEEE

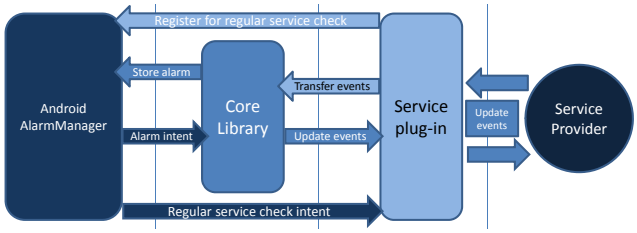

Figure 2. PIKE Architecture for Android

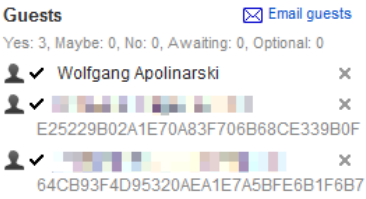

Figure 3. User-level keys, posted in the event in Google Calendar

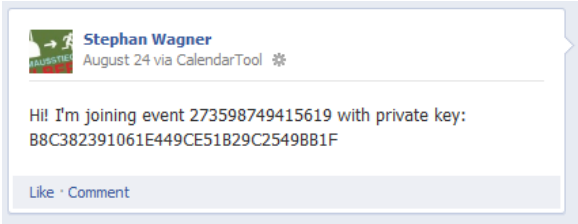

Figure 4. User-level key, posted at a participant's wall

PIKE analyzes the resources of a user regularly in order to detect changes. If a new resource is detected, a secure key exchange is performed. Thereby, PIKE uses the API of the online service to create a group key (shared with all other user's that have access to the resource) and creates additional user-level keys by appending or creating new resources to the existing resource. At the end, each user that has access to the resource possesses two keys, a group key for the shared resource and an individual user-level key that is shared with the creator of the resource. By integrating with an online service and using the service's resources as triggers, the key exchange itself is fully automated, meaning that there is no need for manual interaction between the users apart from the creation of a shared resource.

Following this concept, a typical usage scenario of PIKE works as follows: At first, a user (i.e. the initiator) creates the shared resource and specifies (e.g. *invites* in Facebook) the set of users (participants) that should be able to access the resource. Then, the devices of all participants will retrieve the resource as part of their regular synchronization process. The initiator's device will now create a group key and append it to the resource. A participant's device will retrieve and store this key for later use and create a user-level key which is then shared with the initiator only. To do this, PIKE may create new resources and apply the necessary resource restrictions that are offered by the underlying online service. Eventually, the initiator retrieves and stores all user-level keys and the group key, while the participants possess their individual user-level key as well as the group key. These keys will then be available and used at the time when they are needed. As keys are distributed proactively by PIKE, no Internet connection is necessary at this time.

Since PIKE is piggybacking the key exchange on an online service, the security of PIKE is based on the security that is enforced by the online service on shared resources. In many online services like Facebook or Google Calendar, it is possible to share resources securely with a pre-defined set of users, while other users are not able to see or retrieve the resources. PIKE builds on top of this functionality to enable a secure key exchange.

III. Demonstration

To demonstrate PIKE, we have implemented it as an extensible Android library which enables its use through smart phone apps. The library is used for common tasks like the creation of secure keys and the management of shared resources (i.e. storing keys and trigger time). Additionally, the library supports automatic WiFi hotspot creation and joining such that PIKE-enabled applications can easily establish a secure WiFi network. The architecture of our implementation (see Figure 2) is modular and can be extended with plug-ins for different online services. During the demonstration session, we will make use of two different plug-ins that integrate with some of the online services provided by Google and Facebook.

The Google Calendar plug-in uses appointments with multiple guests as a triggering resource and uses the Google Calendar API to access and manipulate them. For the distribution of the shared group key, the plug-in uses a (hidden) field called shared extended properties that is synchronized between all participants and the initiator. For the user-level keys, the plug-in uses the comment field. This field cannot be seen by other guests, if the event is marked private and guests are not allowed to see the guest list. The initiator's view of the user-level keys is depicted in Figure 3.

The Facebook plug-in uses Facebook's Graph API to access and modify resources. Facebook events are used as trigger resource and the event's Facebook wall is used to distribute a group key. User-level keys are posted on a user's wall which is a private place for discussions. The user-level keys can then be gathered automatically through the event initiator by visiting the Facebook profiles of the event's participants (Figure 4 depicts a user-level key).

To demonstrate the broad applicability of PIKE, we present two applications that we built on top of our PIKE library. The first application shares resources – in our case the readings of a GPS receiver – securely among several devices. To do this, it protects the readings using a shared group

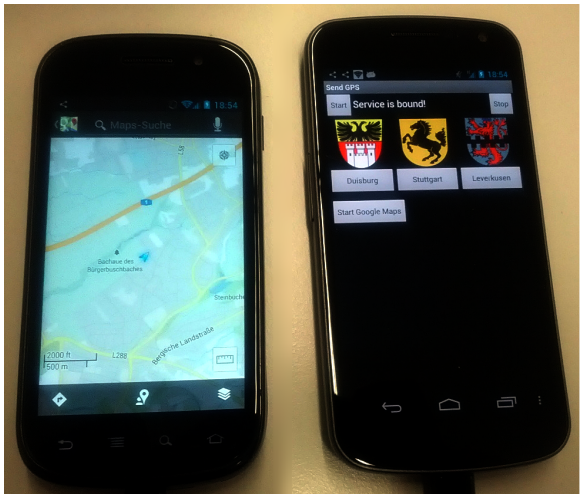

Figure 5. Energy-efficient Resource Sharing Application (Left: Participant Receiving GPS Readings in Google Maps, Right: Initiator Sending GPS Readings)

key that is established automatically by PIKE via Google Calendar. The second application provides configuration-free registration services for large scale events such as a scientific conference. For this, the application uses user-level keys that are created and exchanged automatically by PIKE via Facebook.

A. Energy-efficient Resource Sharing

The energy-efficient resource sharing application (c.f. Figure 5) shares the readings of a single GPS receiver securely among multiple devices. Using PIKE, the devices automatically perform a piggybacked key exchange in response to a new shared meeting in their Google calendars. During the meeting, the initiator's device automatically starts up a WPA2-encrypted WiFi hotspot and shares its GPS readings. The participants automatically connect to the hotspot and join the network using the shared key[1]. After joining the network, the participants receive the GPS readings from the initiator's device and can use them in different applications via Android's *mock locations* feature.

Our laboratory measurements on a Galaxy Nexus running Android 4.1.1 show that a device only running the GPS receiver (747 mW) does need more power than a device connected to a WiFi network, actively receiving the GPS readings (629 mW). A device running the hotspot and the GPS receiver (789 mW) does need more power than a device only receiving GPS readings, but this overhead is outweighed by the total group consumption even when using only two devices (i.e. 2xGPS = 1494 mW compared to 1xGPS and hotspot + 1xWiFi = 1418 mW).

[1]Note that this process does not need an active Internet connection, since PIKE performs the key exchange before the meeting takes place.

B. Configuration-free Conference Registration

The configuration-free conference registration application eases the registration process at conferences using Facebook events. When PIKE detects the event, the participants will automatically exchange a shared group and a user-level key which are then used for registration purposes. At the start of the conference, the participants use their mobile device to identify themselves at the registration desk. In order to do this in a secure manner, they send a verifiable message containing their name to the device of the organizer. This is done by signing the message using the user-level key. Since PIKE is used for the key exchange, the organizers can validate the key and mark the participant as registered. This process is fully automatic, no manual configuration at the conference site is necessary. Furthermore, the devices are not required to be connected to the Internet as PIKE will detect the shared event days before the start of the conference and then exchange and store the keys that will be used at the registration desk.

Acknowledgments

This work has been supported by UBICITEC e.V. (European Center for Ubiquitous Technologies and Smart Cities) and GAMBAS (Generic Adaptive Middleware for Behavior-driven Autonomous Services). GAMBAS is funded by the European Commission under FP7 with contract number FP7-2011-7-287661.

References

[1] Y.-H. Lin, A. Studer, H.-C. Hsiao, J. M. McCune *et al.*, "Spate: small-group pki-less authenticated trust establishment," in *Proc. of the 7th int. conf. on Mobile sys., app., and services*, ser. MobiSys '09, 2009.

[2] J. M. Mccune, A. Perrig, and M. K. Reiter, "Seeing-is-believing: Using camera phones for human-verifiable authentication," in *In IEEE Symp. on Security and Privacy*, 2005.

[3] L. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Ubicomp 2001: Ubiquitous Computing*, 2001.

[4] R. Mayrhofer and H. Gellersen, "Shake well before use: Authentication based on accelerometer data," in *Pervasive Computing*, 2007.

[5] O. Foundation, "Openid authentication 2.0 - final," December 2007. [Online]. Available: http://specs.openid.net/auth/2.0

[6] D. Hardt, "The oauth 2.0 authorization framework, draft-ietf-oauth-v2-31," July 2012.

[7] W. Apolinarski, M. Handte, M. U. Iqbal, and P. J. Marrón, "A piggy-backed key exchange using social networks (pike)," in *Perv. Comp. and Comm. (PerCom), IEEE Int. Conf. on*, march 2013.