Offline Personal Authenticating Device applied in Hospitals and E-banking - OffPAD
Denis Migdal, Christian Johansen, Audun Jøsang

To cite this version:

HAL Id: hal-01589876
https://hal.archives-ouvertes.fr/hal-01589876
Submitted on 19 Sep 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License
Motivation

The concept of Lucidman (Local User-Centric Identity Management) is an approach to providing scalable, secure and user friendly identity and authentication functionalities.

In this context we demonstrate the OffPAD as a trusted device to support different forms of authentication.

The Lucidman/OffPAD approach consists of locating the identity management and authentication functionalities on the user side instead of on the server side or in the cloud.

OffPAD aims to strengthen authentication assurance, improves usability, minimizes trust requirements, and has the advantage that trusted online interaction can be achieved even in the presence of malware infection of client platforms.

A video for this demo is available online.

Authentication Classes and Types

<table>
<thead>
<tr>
<th>Authentication Classes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>System entity (client or server) and a legal/cognitive entity (human or organisation) brings into play multiple entities on each side in the client-server model.</td>
<td></td>
</tr>
</tbody>
</table>

Three authentication types

- **Syntactic** being the simplest, including X.800 certificates, which, e.g., does not prevent phishing attacks since the relying party is indifferent to the identity of the certificate owner;
- **Semantic** authentication includes syntactic and moreover the verification by the relying party that the remote entity has semantic characteristics that are compliant with a specific policy;
- **Cognitive** being the richest, requiring the relying party to have cognitive reasoning power, such as in humans or advanced AI systems. Cognitive authentication effectively prevents phishing attacks as users recognize the server identity, spotting a malicious owner of a legitimate certificate accepted by the browser.

Hardware and Firmware specifications

<table>
<thead>
<tr>
<th>Component</th>
<th>Description / Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>STM32F401 ARM Cortex-M4 32b MCU+FPU, 105 DMIPS, 256KB Flash / 64KB RAM</td>
</tr>
<tr>
<td>Secure display</td>
<td>e-Ink 2.5 inches</td>
</tr>
<tr>
<td>NFC transceiver</td>
<td>NFC1002A2EV, NFC Forum Certificated</td>
</tr>
<tr>
<td>Secure element</td>
<td>Java Card / Global Platform compliant, ST33FIMFEST Micro</td>
</tr>
<tr>
<td>microUSB interface</td>
<td>USB OTG 2.0 (High speed)</td>
</tr>
<tr>
<td>Fingerprint sensor</td>
<td>PPC1020 Touch Fingerprint Sensor Pixel matrix 192x192 @502 dpi</td>
</tr>
<tr>
<td>3 states switch</td>
<td>Mechanical switch: On / Off / Maintenance</td>
</tr>
<tr>
<td>Flash memory</td>
<td>16GB for private/secure storage</td>
</tr>
</tbody>
</table>

ASSUMPTIONS:

- We assume that the sensors integrated in the OffPAD are secure.
- OffPAD still makes use of the host phone for other sensors, like camera, thus a malware on the phone can communicate false information to the OffPAD.
- OffPAD also asks the host phone for the heavier computations, e.g., for OCR. However, all these inputs from the phone are considered in our scenarios as untrusted.
- The OffPAD is a trusted device, i.e. assumed to function as intended and to be adequately protected against relevant attacks. OffPAD is designed to withstand physical or software tempering.
- OffPAD is considered offline, meaning that communications follow controlled formats, during short and restricted time periods, not involving wireless broadband capabilities.
- Being offline eliminates exposure to Internet threats. Thus we assume that attackers are unable to exploit bugs in OffPAD’s firmware and applications.

OffPAD device

<table>
<thead>
<tr>
<th>microUSB</th>
<th>secure element</th>
<th>fingerprint</th>
</tr>
</thead>
</table>

Features

- **Portability:** OffPAD is designed as a phone cover connected to its host with a standard micro-USB interface. This makes the OffPAD a portable object, but not a second electronic object in the user’s pocket.
- **Biometrics:** Unlocking the OffPAD is currently done through fingerprint biometrics.
- **Usability:** OffPAD is intended to increase security assurance with minimal interference with the normal tasks of the user, yet automate some of the authentication and identity management related tasks.

Anywhere security: OffPAD allows to achieve trusted online interactions even on malware infected client platforms like the Android OS.

Demonstrators

**Data-US:** Authentication of user data by the service provider, based on OCR (Optical Character Recognition), alternatively displayed on the OffPAD e-Ink screen.

**SU:** Server authentication by the User, based on petname systems managed by the OffPAD.

**Auto-login:** Contextual automatic login/off based on indoor location of the OffPAD, using Sonitor’s system.

**Multi-login:** Automatic access to a resource conditioned on multiple users authenticated at once, also using TellU Smarttracker system.

**Strong auth.:** Strong authentication required for accessing sensitive information or tasks, using biometric fingerprint authentication of the user by the OffPAD.

Acknowledgements

We thank all OffPAD project members who have either put effort into parts of this demo or have contributed with great ideas or discussions, particularly to:

- Leonard Dallot, Laurent Miralabe, and Guillaume Cornet (TaTag, manufacturer of secure mobile hardware),
- Knut E Haug and Stian Marka (TellU, providing IoT platform and services),
- Marius P. Haugen (U.Oslo),
- Christopher Rosenberger and Estelle Cherrier (ENS Caen GREYC lab),
- Amir Taberkoni (Sonitor, manufacturer of indoor locating solutions),
- Petter Taugbøl (Valbi, managing coordinator).

Presenters

- Denis Migdal, denis.migdal@ensicaen.fr, GREYC Lab
- Christian Johansen, cristi@UiO.no, University of Oslo
- Audun Jøsang, audun.josang@mm.uis.no, University of Oslo

Links

- Detailed Technical Report
- Video
- offpad.org

Demo given at the 23rd Annual ACM Conference on Computing and Communications Security (CCS 2016), during “Mayor’s dinner“ in Vienna.
**Data authentication demonstration**

Sequence of messages/actions for data authentication:

1. **User** types the transaction data in a browser window on the client computer.
2. **User** activates the OffPAD to take a snapshot of the browser window.
3. **OffPAD** sends the snapshot to the server computer.
4. **MAC** generates a signature based on the transaction data and the user’s credentials.
5. **Server** verifies the signature and ensures that the transaction data is valid.
6. **Server** responds with a challenge to the client.
7. **Client** sends a new query with the Challenge and the response.
8. **Server** validates the response and sends the transaction data to the client.

**Notes:**
- **OffPAD** can be replaced by other systems, e.g., based on wireless points.
- **Software** includes interaction with hospital system.
- **Software** can communicate with OffPAD device.
- **Software** interacts with the hospital system.
- **Software** allows the user to log in or access personal TV set-up.
- **Access Control engines** can ensure that OffPAD is in the user's possession.

**User and Server authentication and HTTP XDAA**

Sequence of messages/actions for user and server authentication:

1. **User** initiates connection with the client/browser (which is not trusted).
2. **Server** sends HTTP XDAA challenge to client, along with the Bank’s ID in a certificate (maybe using DNSSEC).
3. **Challenge** and **BankID** are forwarded by the browser to **OffPAD**.
4. **OffPAD** sends the challenge to the client computer.
5. **Client** forwards response to **server**.
6. **OffPAD** computes the response to the challenge from the server.
7. **Server** validates the response and sends the transaction data to the client.

**Notes:**
- **PETNAME** systems for Cognitive authentication.
- **Cognitive** authentication of **Server** by **User**.
- **Context based Access Control**
  - **Hospital system** may require the **User** to authenticate through the **OffPAD** using biometric fingerprint before allowing access.

---

**Petnames systems for Cognitive authentication**

Zooko's triangle for Petname Models.

Bryce “Zooko” Wilcox-O’Hearn described in 2005 three fundamental desirable properties of names:
- global (like DOIs or URLs); unique (collision-free within a domain); memorable (or human-meaningful).

**The Petname Model** consists of two name spaces:
- one of global and unique names (pointers);
- one of memorable and unique names (pet-names);
- mapping a name space of pointers to individual name spaces of petnames, which thereby combines all three desirable properties.

**HTTP Extended DAA (XDAA)**

HTTP Digest Access Authentication (HTTP DAA, RFC 7616) is an authentication protocol:
1. client sends a query (HTTP GET);
2. server responds with a Challenge (HTTP 401);
3. client prompts for username and password;
4. client computes the Response;
5. client sends a new query, with the Response;
6. server answers the query (HTTP 200).

**Klevjer, Varmetal, & Jøsang** extend HTTP DAA:
- challenge forwarded to the OffPAD;
- password generated from user biometric;
- HA1 can be precomputed and stored on the OffPAD.

**Paired and Take in use**

- **Nurse** takes **OffPAD** in use at beginning of shift by authenticating through biometric fingerprint.
- **Device** loads user profile and registers to **TellU**.
- **Nurse** pairs location **TAG** to **OffPAD** device.
- **Pairing** information sent to **TellU** server.

**Context based Access Control**

- **Continuous biometric authentication methods** can ensure that **OffPAD** is in the user’s possession.

**Multiple users Access Control**

- **Both Nurse and Patient** are in front of a **Terminal**.
- **TellU** system triggers an access control event parameterized by the objects.
- **Sensitive information** can now be displayed, e.g., **Patient Record** can be shown only when the Patient is in the room.