Jøsang *et al. Journal of Trust Management* (2015) 2:1
DOI 10.1186/s40493-014-0009-6

Journal of Trust Management
a SpringerOpen Journal

RESEARCH Open Access

Local user-centric identity management

Audun Jøsang[1*], Christophe Rosenberger[2], Laurent Miralabé[3], Henning Klevjer[4], Kent A Varmedal[5], Jérôme Daveau[6], Knut Eilif Husa[7] and Petter Taugbøl[8]

*Correspondence: josang@ifi.uio.no
[1]University of Oslo, Oslo, Norway
Full list of author information is available at the end of the article

Abstract

Identity management is a rather general concept that covers technologies, policies and procedures for recognising and authenticating entities in ICT environments. Current identity management solutions often have inadequate usability and scalability, or they provide inadequate authentication assurance. This article describes local user-centric identity management as an approach to providing scalable, secure and user friendly identity management. This approach is based on placing technology for identity management on the user side, instead of on the server side or in the cloud. This approach strengthens authentication assurance, improves usability, minimizes trust requirements, and has the advantage that trusted online interaction can be upheld even in the presence of malware infection in client platforms. More specifically, our approach is based on using an OffPAD (Offline Personal Authentication Device) as a trusted device to support the different forms of authentication that are necessary for trusted interactions. A prototype OffPAD has been implemented and tested in user experiments.

Keywords: Authentication; Trust; OffPAD

Introduction and background

Trusted interaction between users and service providers in online environments depends on robust mutual authentication. Given the exponential growth in the number of interconnected online entities, where each entity often gets multiple and changeable identities, the challenge of ensuring robust authentication between all these identities is daunting. Because authentication involves two parties - the relying verifier as well as the authenticated target - identity management requires process components at each location at least, and possibly at third-party locations as well. All parties that engage in online activities have identities that need to be managed. In the traditional client-service architecture, the identities of both the client and server must be considered. In addition, it must also be taken into consideration that there are people, organisations and systems with identities that need to be managed, both on the client and server sides. Finally, it is necessary to consider data authentication as a separate security service in the context of identity management. This is because the origin of data can not be derived from entity authentication alone. A realistic threat is for example that malware on a client platform could modify data provided by a user even if the user has been correctly authenticated. From these simple observations we quickly realise that the general identity management problem is quite complex.

Identity management can be defined as the set of technologies, processes, policies and standards that enables identification and authentication of entities as digital identities. Entities are typically persons or computer systems, where either can be active on the client side or on the server side. An identity is a set of attributes that characterise the entity, where one of the attributes typically is a name that serves as an identifier for the entity within a domain. Figure 1 below illustrates how the concepts of entity, identity, and attributes are related.

A digital entity is a set of digitally expressed attributes of an entity, where one of the attributes typically is a name or identifier for uniquely selecting the entity within a name-space domain. Identity attributes can be self-assigned such as usernames on social website, assigned by an authority such as social security numbers, dictated by circumstances such as a home address, or be intrinsically linked to the entity such as biometric characteristics. Identities can be transient (shorter period than the entity's lifetime), permanent (as long as the entity exists) or persistent (remains even after the entity no longer exists). Each entity can have multiple identities simultaneously or at different points in time. For example, a person's professional and private identities are typically partially overlapping (i.e. with some common attributes), and a person's various digital online identities can be partially overlapping or even totally separate without any common attributes.

Identity management for online services has traditionally focused on managing user identities through processes located on the server side, where solutions are optimized for simple management from the service providers' point of view. The *silo model* is the traditional identity management model used in this approach where each service provider (hereafter abbreviated as SP) governs a separate 'silo domain' with its own name space of user identities. A specific user normally has separate identities and names in different silo domains, and in addition must use a different authentication credential for each identity. The silo domain model results in severe usability problems due to the identity overload that it generates, i.e. that users accumulate a high number of identities.

Online activity typically leads to the accumulation of so many digital identities and passwords that it becomes a real challenge to manage them in an efficient and secure way. In recent years, security architects have therefore focused on new solutions that can improve the user experience, often dubbing such solutions as *'user centric'*. One of the main approaches to user-centric identity management is identity federation where the basic

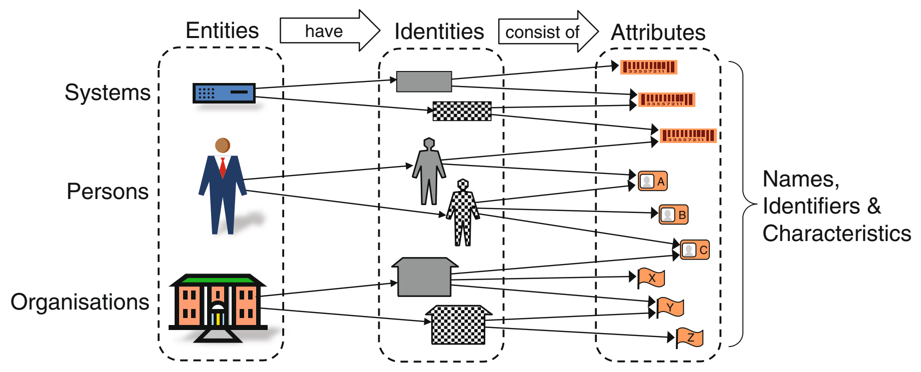

Figure 1 Relationships between entities, identities and attributes.

idea is to let users access multiple separate service domains with the same identity and credential. This is made possible by interconnecting the federated service domains with a common security protocol and by agreeing on a common security policy whereby user authentication by one identity provider is accepted by all other SPs (service providers) in the federation domain. Identity federation comes in many variations but is typically based on the SAML[a] standard which is implemented in applications such as Shibboleth, OpenId, YouProve, as well as FacebookConnect.

Identity federation can certainly simplify the user experience in many situations, but it does not fundamentally solve the problem of identity overload. There will always be different federation domains, because not all SPs will merge their respective user identity domains into a single federated domain. In addition, a side-effect of identity federation is that it requires additional trust in specific third parties, and can give monopolistic business power to specific parties in the federation. The identity provider in a federation is for example able to collect information about user activity. Selling this information for profit or using it for targeted advertisement constitutes a privacy breach in many jurisdictions. It is commonly assumed that users would abandon identity providers that fail to protect user privacy. Paradoxically, federated identity providers seem to be only minimally affected by trust erosion caused by such privacy breaches, which is a serious trust management problem. Finally, identity federation processes are mostly located on the server side or at some other location in the Internet cloud, which means that identity federation which is typically dubbed as user-centric in reality is network-centric.

We argue that there is a need for identity management solutions that can really solve the identity overload problem, that have simple trust requirements, that lead to balanced business power between SPs, and that are user-centric from both a practical and technical point of view. We use the term *local user-centric identity management* (abbreviated Lucidman) in order to distinguish between this identity management model and other models that are often called user-centric in the literature. More specifically, Lucidman not only provides robust security and adequate security usability, it is also based on placing technology and computing processes for identity management locally on the user side in the client-server architecture. An essential element of the user-side technology is the OffPAD (Offline Personal Authentication Device) which acts as a trusted device that can support strong multilateral authentication even in situations of compromised client platforms. This article describes Lucidman as a principle for identity management based on the OffPAD to support secure and user friendly mutual entity authentication as well as data authentication.

Our proposed solution is based on an external trusted device called an OffPAD (Offline Personal Authentication Device) as shown in Figure 2. The OffPAD enables different authentication classes described in more detail below.

A paradox in today's Internet computing environment is that we continue to build vulnerable client platforms while still expecting them to support trusted online interactions. According to PandaLabs' estimates, approximately one third (31.63%) of the world's PCs are infected with some sort of malware (Q2 2012) of which most (78.92%) are Trojans [1]. It must therefore be assumed that sooner or later a client platform will be infected by malware, which means that it can never be trusted. In order to ensure resilience in such adverse situations Lucidman also ensures trusted interactions even when it is assumed that client platforms are infected by malware.

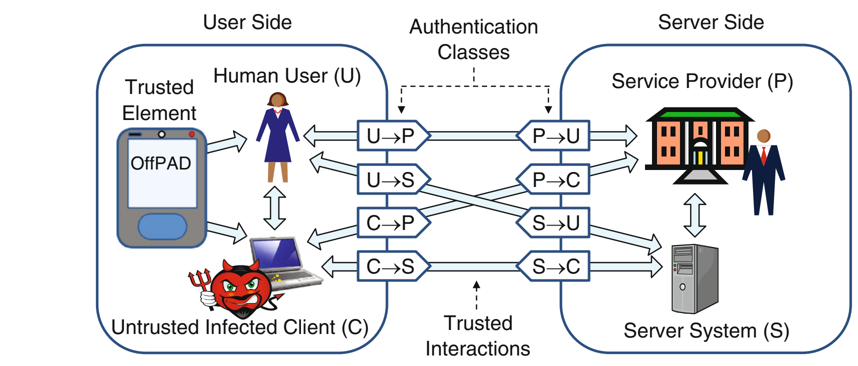

Figure 2 Trusted interactions in the presence of malware-infected clients.

The OffPAD is not connected to the Internet nor to other public communication networks, but it can communicate with the human user and with local client platforms in a controlled way. The network architecture in Figure 2 consists of two entities (client and user) on the client side, and of two entities (server system and SP organisation) on the server side, which thereby extends the traditional client-server architecture that only has one entity on each side. The term *ceremony* [2] is typically used to denote network protocols that involve entities and interactions external to the traditional system entities, i.e. where nothing is out-of-band. The solutions described in this article consists of ceremonies that are described on a relatively high level of abstraction.

Below is first described related work. Subsequently are described the different authentication classes and authentication modalities, the OffPAD device itsefl, the ceremonies for local user-centric identity management, a security analysis of the OffPAD, the user experiment with the prototype OffPAD, and finally the discussion and conclusion.

The principles and technologies for local user-centric identity management and trusted interaction based on the OffPAD have been developed within the Lucidman project (Eureka project number 7161) which is a collaborative research project supported by the Franco-Norwegian Foundation in the period 2011-2013. Partners in the project have been the University of Oslo, Tellu and Vallvi in Norway, and ENSICAEN, TazTag and CEV in France.

Related work

Several authentication solutions (particularly unimplemented designs and recommendation) relying on an external device are present in the literature. Examples include the Pico by Stajano [3], MP-Auth by Mannan and van Oorschot [4] and Nebuchadnezzar by Singer and Laurie [5]. However, these devices only support authentication of client-side entities to server-side parties, i.e. typically user authentication, in contrast to the OffPAD which also supports the authentication of server-side entities to client-side parties, as well as data authentication. Below we briefly summarise related work on external authentication devices.

Pico

The Pico [3] is a device that authenticates a user through a challenge-response protocol. It stores private keys for communication with every application it supports authenticating to, in its on-board encrypted memory. Each supported application has one asymmetric key pair to communicate with Picos.

Challenges are presented as 2D visual codes (e.g. QR codes) to the Pico, and collected by the device's embedded camera. Transmission of the response is done over Bluetooth. The Pico solution requires changes to both the client and the server side. Most SPs are probably reluctant to consider changing their visual appearance to support another authentication scheme. Where the Pico is restricted to its own authentication scheme, the OffPAD authentication is done building on a pre-existing technology. Also, the Pico only supports user authentication, whereas the OffPAD supports multilateral authentication.

MP-Auth

The MP-Auth (Mobile Password Authentication) protocol was proposed as a means for moving password authentication (not the passwords themselves) to a personal device, protecting them against being collected by malware. In this protocol, an SSL tunnel is established between the user's mobile phone and the server to which he wants to authenticate. The user's password or credential is then entered on the phone and transmitted, protected by the SSL tunnel, to the server, authenticating the user [4].

MP-Auth's solution relays the communication and entering of a password to a mobile phone, but does not provide the benefit of identity management. This solution only supports user authentication.

Nebuchadnezzar

The authors of Nebuchadnezzar assume that trying to establish a trusted path of communication between a general purpose operating system and a server is a bad idea. They also assume that the other extreme: trying to run every application on a minimal, secured, locked down operating system, is also a bad idea. They therefore propose that the only sensible solution is a combination of the two. The position paper [5] describes the principles of the Nebuchadnezzar, which much like the OffPAD is an external device, and which is minimized with regard to features. The Nebuchadnezzar may be seen as a the OffPAD part that supports user authentication. However, our OffPAD concept also supports additional types of authentication, not just user authentication.

Trusted platform module

The TPM is mentioned here because it represent a security module integrated on computing platforms and that is resistant to malware attacks. The TPM also provides a method for cryptographic authentication of client platforms. The term 'TPM' is at the same time the name of a set of specifications [6] issued by the Trusted Computing Group, as well as the name of the hardware chip that implements these specifications. TPM chips are commonly installed in computer systems shipped since 2006, with the purpose of providing a robust hardware based mechanism for obtaining security assurance about various integrity aspects of systems. The TPM chip can e.g. be used to verify that the OS loader has not been modified by malware, and can also be used to authenticate (client) systems to external parties, which corresponds to client authentication (class [C $\rightarrow$ S]) in Figure 2. TPM based system authentication relies on the EK (Endorsement Key) pair which is a public/private key pair that is unique for each TPM, generated and installed by the manufacturer or the vendor of the TPM. However, the TPM can not support user authentication as [U $\rightarrow$ S], server authentication as [S $\rightarrow$ U], nor cognitive data authentication.

Authentication classes and modalities

This section explains the concepts of entity and data authentication, and introduces three distinct modalities of authentication called syntactic, semantic, and cognitive authentication.

Entity authentication

The distinction between a system entity (client or server) and a legal/cognitive entity (person or organisation) brings into play multiple entities on each side in the client-server model, as illustrated in Figure 2. A requirement for trusted interaction in this scenario is to have mutual authentication between pairs of interacting entities whenever relevant, leading to 4 possible types of mutual entity authentication as shown in Figure 2 and described in Tables 1 and 2.

Some of the entity authentication classes in Tables 1 and 2 are relatively impractical, such as [C → P] and [P → C], but they illustrate the generality of entity authentication with non-atomic user and server sides. The authentication class [U → P] is e.g. practiced when authenticating customers over the phone by asking about customer number, date of birth, etc. The X.800 standard focuses on entity authentication classes [C → S] and [S → C] which take place at the network protocol layers and are typically transparent to the human user.

For online service access the entity authentication classes [U → S] (user authentication) and [S → U] (cognitive server authentication, defined below) are the most relevant. The importance of these authentication classes emerges from the need for end-to-end security. End-to-end communication between the human user (U) and the server system (S) takes place during online service access. It is therefore pragmatic to require mutual authentication between those two entities. Traditional user authentication can provide [U → S] authentication. It is often incorrectly assumed that traditional server authentication with Browser PKIX[b] server certificates and TLS[c] provides [S → U] authentication, however in reality it does not. This might seem surprising but is in fact easy to understand [7].

For example, phishing attacks normally start with spam email messages that invite people to access a fake web site masquerading as a genuine web site that e.g. tricks the user into providing user Id and password. In a syntactic sense the fake phishing website can be correctly authenticated through TLS because the server certificate is validated by the browser. However, from a cognitive point of view this is not authentication because the website's identity is different from that intended by the user. The problem is due to the poor usability offered by current implementations of TLS [8,9] which does not facilitate cognition of identities nor any understanding of what they represent.

Table 1 Authentication classes for user-side entities as illustrated in Figure 2

Class	Authentication of user-side entities
[U → P]	User (U) authentication by the service provider (P)
[U → S]	User (U) authentication by the server system (S) (commonly called *user authentication*)
[C → P]	Client (C) authentication by the service provider (P)
[C → S]	Client (C) authentication by the server system (S)

Table 2 Authentication classes for SP-side entities as illustrated in Figure 2

Class	Authentication of SP-side entities
[P → U]	Service provider (P) authentication by the human user (U)
[P → C]	Service provider (P) authentication by the user client (C)
[S → U]	Server (S) authentication by the human user (U) (here called *cognitive server authentication*)
[S → C]	Server (S) authentication by the user client (C)

According to the X.800 standard, entity authentication is *"the corroboration that a peer entity in an association is the one claimed"* [10]. Here, 'association' means a connection, a session or a single instance of communication. We use the term 'interaction' as a general term for the same thing. So in case a victim user intends to connect to `https:\\www.paypal.com`, but is tricked into connecting to a phishing website called `https:\\www.peypal.com`, then the server certificate claims that the server identity is `www.peypal.com` which then is correctly authenticated according to X.800. However, something is clearly wrong here, and the failure to capture this obvious security breach indicates that the above definition of entity authentication is inadequate. What is needed is a richer modality of authentication. We define three authentication modalities where *syntactic authentication* is the poorest, where *semantic authentication* is intermediately rich, and where *cognitive authentication* is the richest modality, as described next.

Authentication modalities:

- **Syntactic entity authentication:** *The verification by the relying entity that the unique name of the remote entity in an interaction is as claimed.*
  This basic form of entity authentication is equivalent to peer-entity authentication as in X.800. Syntactic authentication alone does not provide any meaningful security and can e.g. not prevent phishing attacks since the relying party is indifferent to the identity of the authenticated entity.
- **Semantic entity authentication:** *The verification by the relying entity that the unique name of the remote entity in an interaction is as claimed, and in addition the verification by the relying entity that semantic characteristics of the remote entity are compliant with a specific security policy.*
  Semantic entity authentication can be enforced by an automated system e.g. with a white list of identities that have been authorized for interaction.
- **Cognitive entity authentication:** *The verification by the cognitive relying party that the unique name of the remote entity in an interaction is as claimed, and in addition the verification by the relying party that semantic characteristics of the remote entity are compliant with a specific security policy, where the latter is supported by presenting in a user-friendly way identity attributes that enable the cognitive relying party to recognise relevant aspects of the remote entity and to judge policy compliance.*
  Cognitive entity authentication requires the relying party to have cognitive reasoning power, such as in humans, animals or advanced AI systems. This authentication modality effectively prevents phishing attacks because users recognise the identity of a server and decide whether it is the intended one.

Technologies for SP authentication are very different from those of user authentication, because user authentication (class [U $\rightarrow$ S]) typically takes place on the application layer e.g. with passwords, whereas SP authentication – interpreted as server system authentication (class [S $\rightarrow$ C]) – typically takes place on the transport layer e.g. with TLS.

User identity management is frequently discussed in the identity management literature, whereas SP identity management is mostly discussed in the network security literature. Lucidman applies to both types because users must manage their own identities as well as those of SPs, in order to be authenticated by server systems on a semantic level, and to authenticate the server systems on a cognitive level. Lucidman is aimed at providing adequate security assurance and usability for the management of both user identities and server identities, with the goal of enabling trusted interaction between online entities.

Data authentication

According to the X.800 standard data origin authentication is *"the corroboration that the source of data received is as claimed"* [10]. This is different from entity authentication because knowing the identity of a remote entity in a session is different from knowing whether the data received through a session with a specific remote entity genuinely originates from that entity. This difference might seem subtle at first glance but it is in fact fundamental for security, as explained below.

Malware infection on client platforms opens up for attacks against data authentication that entity authentication can not prevent. More specifically, entity authentication is insufficient for ensuring trusted interaction in case the client platform is compromised. A typical example is the situation of online banking transactions with mutual entity authentication. Even with strong 2-factor user authentication, and users' correct interpretation of server certificates, there is the possibility that malware on the client platform can modify data communicated between client and server platforms, and thereby compromise transactions. Current attacks against online banking are typically mounted in this way. Such attacks lead to breach of data integrity, but not a breach of entity authentication.

The preparation for this type of attacks typically includes tricking the user into installing a Trojan, i.e. a program that really or seemingly does something useful, but that in addition contains hidden malicious functionality that allows the attacker to take control of the client platform. During an online banking transaction the attacker uses the Trojan program to change transaction details without the user's knowledge. SpyEye, Zeus, IceIX, TDL, Hiloti, Carberp, and many others [11,12] are concrete examples of malware that enable such attacks.

The separation between the human/legal entity and the system entity on each side of a client-server session – as illustrated by Figure 2 – makes it necessary to specify which entity in particular is the assumed origin of data. In case e.g. the human user is assumed to be the origin, and the client system modifies data input by the user before it is sent to the server system, then this would be a breach of data origin authentication. However, in case the client system is assumed to be the origin, the same scenario would not be a breach of data authentication. The general rule is that the object of entity authentication must also be the origin of data authentication. For typical online transactions where the human user is directly involved and authenticated, the user must also be seen as the origin of data for data authentication purposes. Unfortunately current solutions for user data origin authentication are either non-existent or inadequate because they assume the client system to be the origin of data [13].

Users rely on visual cues to know whether a browser session is secured with TLS. After verifying that TLS is enabled, averagely security aware users will comfortably input their banking account and transaction details into the browser window. However, many users ignore that malware like those mentioned above has functionality commonly known as a *'web inject'* that can change the behaviour of the browser and modify input and output data arbitrarily.

The fact that entity authentication and data authentication are two separate security functions implies that it is necessary to have specific security mechanisms to ensure data integrity in online transactions. The OffPAD enables data origin authentication with high assurance and usability, as explained below.

Data origin authentication can also be implemented with different modalities, similarly to entity authentication. It is thus meaningful to speak about syntactic, semantic or cognitive data authentication.

OffPAD – the Offline Personal Authentication Device

The PAD (Personal Authentication Device) is described by Jøsang and Pope [14] as a secure device external to the client computer platform. It functions as a user-friendly personal identity manager for automated user authentication to any online service. The PAD is the conceptual predecessor to the OffPAD.

The OffPAD (Offline Personal Authentication Device) described by Klevjer *et al.* [15] and Varmedal *et al.* [16] is an enhanced version of the PAD, where an essential characteristic is to be offline, i.e. not connected to the Internet. Keeping the OffPAD offline strengthens its security by eliminating exposure to Internet threats. The requirement of being offline means that the OffPAD should not have Wi-Fi or wireless broadband capabilities, but can e.g. support limited use of NFC in short periods. The OffPAD represents local user-centric identity management because it enables secure and user friendly management of digital identities and credentials locally on the user side. The OffPAD supports authentication of both user and SP identities (i.e. mutual authentication) and can in addition support data authentication. For access to the OffPAD, the user must unlock the device by using e.g. a PIN, pass phrase, biometrics or other adequate authentication credentials. A possible OffPAD design is illustrated in Figure 3.

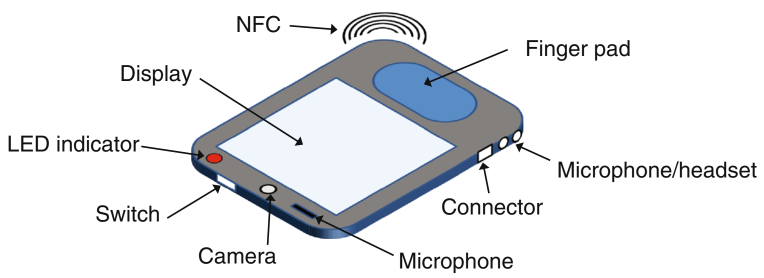

Figure 3 OffPAD design ideas.

The OffPAD is a trusted device, meaning that it is assumed to function as intended and to be adequately protected against relevant attacks. Attacks originating from the Internet are effectively eliminated by keeping the OffPAD offline. The OffPAD has limited connectivity to client platforms e.g. through wireless NFC or through USB cable which theoretically represents an attack channel. These communication channels must therefore be carefully controlled, e.g. by sanitizing the received data. Protection against attacks resulting from physical theft is to have traditional access control based on PIN and biometrics [17,18], combined with some level of physical tamper resistance.

It is not required that the OffPAD operating system and applications are free from vulnerabilities, because it is assumed that attackers are unable to exploit such vulnerabilities since the OffPAD is offline. In that sense, a specific software bug which would have been a vulnerability in an online system is strictly speaking not a vulnerability in the OffPAD because it can not be exploited. In other words, security vulnerabilities are eliminated simply by keeping the OffPAD offline.

The OffPAD may have several interfaces for communication. Microphone, touch screen and camera may be used for voice and face recognition, and a fingerprint reader may be used for both authenticating to the device and elsewhere.

The requirement of being offline does not totally exclude electronic communication with the OffPAD, but means that the communication follows controlled formats and takes place in short, restricted time periods. This decoupling from networks strengthens security as it removes threats from outside attacks.

Although electronic communication channels is normally be disconnected, limited communication is possible. NFC (Near Field Communication) as well as a backup USB connection are seen as the most suitable communication technologies for the OffPAD. Both technologies are fast, USB guarantees (physically) that the correct device is connected, and NFC gives high visual assurance that the correct device is connected. This limits the threat of a man-in-the-middle attack when connecting an OffPAD to a computer.

The first connection to the OffPAD requires Trust-On-First-Use (TOFU), also known as leap-of-faith [19]. On first use, there is no cryptographic way to verify the connection between the device and the client platform, the trust must simply be based on the physically observed set-up. On the first connection, pairing between the device and computer occurs, so that the subsequent connections can be verified to be between the same device and computer.

For our prototype implementations the TazCard device produced by TazTag[d] was used as a prototype OffPAD. The TazCard does not have all the functions described above which must be understood as an ideal OffPAD that currently does not exist. The TazCard is an android device with a touch screen that communicates via NFC and USB, which means that it has the basic characteristics of an OffPAD.

Local user-centric identity management

The OffPAD represents technology for improving the user experience and for strengthening security, which already makes it user centric. Since the OffPAD in addition is located on the user side, it is physically local to the user, and thereby represents technology for local user-centric identity management. The OffPAD can be used for a number of different security services [16], but this article only focuses on how it enables trusted interaction through mutual entity (user and server) authentication as well as data origin authentication. The OffPAD can also support incremental authentication modalities, i.e. syntactic, semantic or cognitive authentication, as shown in Figure 4.

In the sections below is described how the OffPAD enables mutual user-server entity authentication as well as data authentication. Each authentication type is illustrated with a ceremony [2] which is simply a protocol where relevant actions by users and the context environment are included. The 3 ceremonies can be chained and seen as one general ceremony that provides all 3 types of authentication, starting with server authentication, followed by user authentication and finally data authentication. The novelty of our solutions is that it supports trusted interaction even in the presence of malware infected client platforms.

Server authentication supported by the OffPAD

Secure access to online services is typically implemented with mutual authentication, i.e. with user authentication on the application layer and server authentication on the transport layer. Since mutual authentication goes between the user and the server, we require server authentication by the user as [S $\rightarrow$ U] which most often is not satisfied in current implementations, and user authentication by the server as [U $\rightarrow$ S] which currently is satisfied in most implementations.

Despite strong cryptographic server authentication, the typical implementation of TLS in browsers only enforces syntactic server authentication, so that any server identity is accepted as long as the certificate is correctly validated. This is a serious vulnerability which is also the reason why phishing attacks often succeed even when TLS is being used for server authentication [20].

Because phishing works fine without server certificates, most phishing sites do not have server certificates. Phishing victims often do not know the difference between a http connection (without certificate) and a https connection (with certificate and TLS), so server certificates are probably seen by attackers as an unnecessary expense. However, there are phishing sites with certificates, and paradoxically, phishing site certificates are automatically validated by browsers.

Current browser implementations of TLS with certificates do not support semantic or cognitive server authentication, so using TLS with certificate does not offer any meaningful authentication. In order to provide cognitive server authentication, a *petname system* [21-23] must be used, as described below.

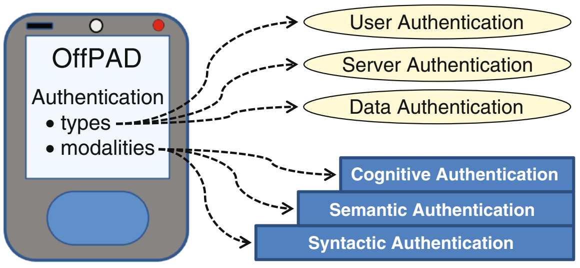

Figure 4 Authentication types with incremental modalities enabled by the OffPAD.

Reliable authentication of online entities require globally unique names that are understood by people. Domain names were designed to represent online identities of organisations, but domain names alone can be difficult to interpret.

Company names, trademarks and logos are typically used for recognising organisations in the real world, but would not be suitable for global online identification and authentication. This mismatch between names used in the online world and in the real world creates confusion about which unique domain name to expect when accessing online services. Without knowing which name to expect, authentication becomes meaningless.

Three fundamental desirable properties of names described by Bryce 'Zooko' Wilcox-O'Hearn [24] are to be global[e], unique[f] and memorable[g]. To be memorable a name has to pass the so-called *'moving bus test'* [25] which consists of testing whether an averagely alert person is able to correctly remember the name written on a moving bus for a definite amount of time, e.g. 10 minutes after the bus has passed. A name is unique if it is collision-free within the domain [23].

The triangle of Figure 5 where each of the three desirable properties of names are placed in one of the corners is commonly known as Zooko's triangle, and represents the basic foundation for the Petname Model [21,24].

Wilcox-O'Hearn claimed with supporting evidence that no name space can be designed where names generally have all three desirable properties simultaneously. A visual analogy of this idea is created by placing the three properties at the three corners of a triangle. In a triangle, the three corners are never connected by a single line, only pairs of corners are connected. The edges joining the corners then illustrate the possible properties that a name space can have. Wilcox-O'Hearn suggested to design name spaces of global and unique names (pointers), and name spaces of memorable and unique names (petnames).

The Petname Model consists of mapping a common name space of pointers to individual name spaces of petnames, which thereby combines all three desirable properties. A petname system is a system that implements the Petname Model. The OffPAD is ideal for hosting a petname system, so a simple petname system was implemented on the OffPAD represented by the prototype TazCard device.

In our implementation, the petname system on the OffPAD receives server certificates during the TLS handshake. In order to support cognitive server authentication, the server domain name (pointer) – received in a certificate – is mapped to a user-defined petname representing the SP, as illustrated in Figure 6. The server certificate is also validated in the

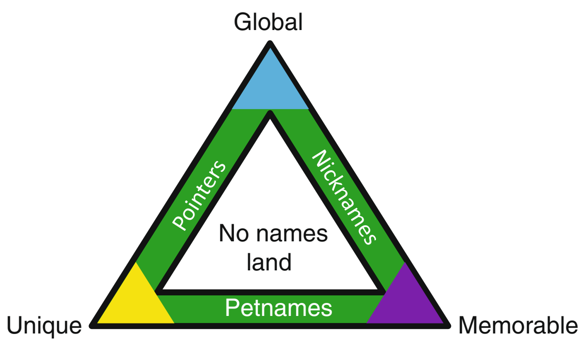

Figure 5 Zooko's triangle.

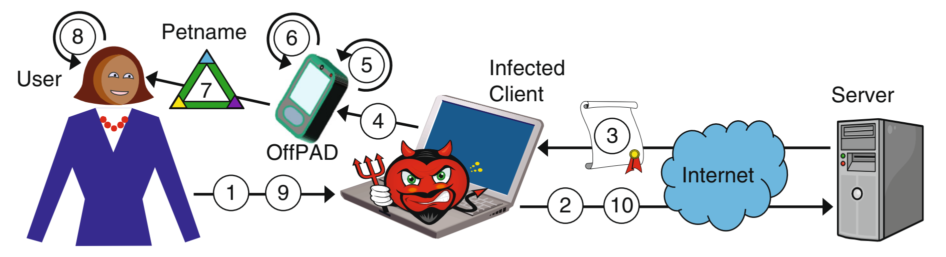

Figure 6 Ceremony for server authentication with the OffPAD and with malware-infected client.

traditional way, which provides syntactic server authentication. Strengthened authentication assurance can be obtained by having server certificates signed under DNSSEC, which would give a very high Server Authentication Assurance Level according to the server authentication framework proposed by Jøsang *et al.* [20].

The actions/messages of the ceremony of Figure 6 are described in Table 3.

The petname system combined with e.g. TLS enables manual trust evaluation by the user [21,26]. More specifically, a petname system in combination with TLS provides cognitive server authentication, which is precisely the security service needed to prevent phishing attacks.

If the user navigates to a web site where the petname system finds no match between its pointer (domain name) and a petname, the petname system should alert the user, or ask the user to add a new petname for the new service. Petname systems support cognitive authentication and protect users from falling victim to phishing attacks [21].

In a petname system, users create personal petnames to represent globally unique pointers for services that they use. When a specific service is accessed, the user recognises it by its petname, not by its pointer. A petname system makes it easy to manage the list of mappings between petnames and pointers, and automatically translates between pointers and petnames.

User authentication supported by the OffPAD

Online service users typically accumulate so many identities and related passwords that it becomes a usability challenge to manage them securely. In addition, since it is reasonable to assume that typical client platforms are infected by malware, storing and using

Table 3 Sequence of messages and actions for server authentication ceremony

Nr.	Message/action description
1.	User initiates secure TLS connection through client platform
2.	Client platform contacts server
3:	Server returns server certificate containing public key
4.	Server certificate is forwarded to OffPAD
5.	Server certificate is validated (syntactic server authentication)
6.	Server certificate is mapped to petname
7.	Petname is presented to user
8.	User performs cognitive server authentication
9.	User approves server authentication
10.	TLS connection established between client and server

credentials such as passwords on these platforms is a security risk. Passwords can be intercepted by Trojans either through keystroke logging, RAM-scraping, or by screenshots when shown on the screen in clear text. Even automatic log-in and identity management applications such as LastPass are not safe, as they release the clear text password to the web browser (or other application) during authentication, leaving it visible in memory for the Trojan to steal.

The OffPAD may be used to manage and authenticate a user to a system in a secure way. This would improve usability by providing a tool for identity management, and would also improve security in several respects. In a traditional scenario where the user types his password, or the password is decrypted by a password manager (e.g. LastPass), the password is exposed in the computer's memory and is vulnerable to attacks such as key logging or memory inspection. A solution for password authentication using an OffPAD is proposed by Klevjer *et al.* in [15], consisting of an extension to the original *HTTP Digest Access Authentication* scheme specified as a part of the HTTP standard in [27]. User credentials are stored in a hashed format on both the server and the OffPAD. When the user via the client requests a protected resource, the server responds with an authentication challenge, which on the client side is hashed with the user credentials and returned to the server. The server does the same challenge-response calculations locally, and compares the result and the response. If the two values match and the user corresponds to an authorized entity, the user is granted access to the resource. This can be done securely through an insecure channel, such as over HTTP, not requiring an extra connection to the server, just a browser plug-in or extension.

In this section, we describe a method for local user-side identity management based on the OffPAD, combined with an extension of the well-known HTTP Digest Access Authentication protocol. A brief overview of the existing HTTP Digest Access Authentication standard is provided next. We then describe our method of combining the OffPAD with extended HTTP Digest Access Authentication. The advantage of our method is that it totally prevents exposure of passwords on potentially vulnerable client platforms, and thereby represents secure local user-centric identity management solution. A proof-of-concept implementation of the method has been developed for the OffPAD prototype [28].

HTTP Digest Access Authentication (short: DAA) originates from the challenge-response authentication framework described in the original HTTP 1.0 specification [29]. It is a web standard for access control to a service or domain called *realm* by user authentication over HTTP. DAA was first defined in 1997 in RFC 2069 [30] and refurbished in RFC 2617 [27] in 1999. Its intended use is on the World Wide Web, but it is perfectly implementable for protection of local resources, or in any situation where application level access control is required[h].

The actions/messages of the ceremony of Figure 7 are described in Table 4.

Extended Digest Access Authentication (short: XDAA) represents an extension of traditional HTTP DAA in two respects. The actual IETF standard RFC 2617 is extended to allow more than just username and password as valid credential sets. The authentication process itself is also extended, physically, in that it is moved to another location. All client-side calculations done in the authentication phase are outsourced to the OffPAD.

Our XDAA is beneficial both for security and usability. By managing the user credentials on an external device, we get a local user-centric identity management system,

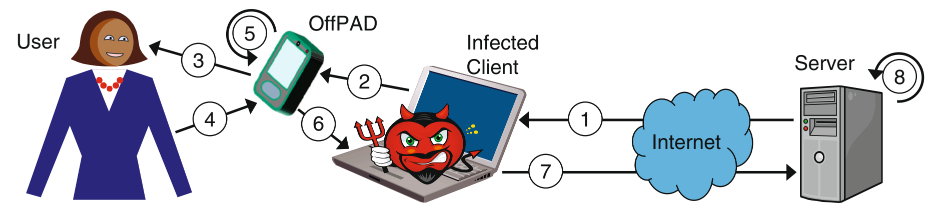

Figure 7 Ceremony for user authentication with the OffPAD and with malware-infected client.

so users are no longer required to remember their passwords. Moving the challenge-response calculations and handling of the values critical to authentication over to a mostly offline device, we reduce the risk of exposing these values. Moving the identity management over to such a device alleviates the cognitive and physical strain on the user during authentication, as well as removing the time penalty brought by user interaction in most situations[i].

In its simplest form (using the OffPAD and no further protection mechanisms), XDAA can be used with any HTTP server supporting original HTTP DAA without change to the server system. The immediate benefit is that the user's credentials themselves are never present. They are never shown on the screen, never exposed in any vulnerable state in the computer's memory and never transferred in clear text.

Data authentication supported by the OffPAD

Users generally rely on what they see on a computer display to read the output of transactions, to verify that they type correctly, and to ensure that the data being sent through online transactions is according to their intentions. In general, all this depends on the integrity of the computing platform to which the VDU (Visual Display Unit) is connected. Assuming that the computing platform is infected with malware it is *a priori* impossible to trust what is being displayed to be 100% correct [31-35].

The prospect that the computer display can lie to us is both frightening and real. This problem is amplified by the fact that we often read data from platforms that are not under our control, and that hackers have incentives for trying to manipulate the systems and the way data is displayed. For example, typical attacks against online banking consist of using a malicious Trojan on a client computer to send falsified transaction data to the bank server while displaying what the user expects to see.

In order to provide data authentication, some online banks offer SMS authorization of transactions, which consists of mirroring the received transaction data (destination

Table 4 Sequence of messages and actions for user authentication ceremony

Nr.	Message/action description
1.	Server sends http digest access authentication challenge to client
2.	Challenge from server is forwarded to OffPAD
3:	OffPAD presents user authentication request to user
4.	User approves authentication request
5.	OffPAD computes response to challenge from server
6.	Response is sent from OffPAD to client
7.	Client forwards response to server
8.	Server verifies response which completes user authentication

account and amount to be transferred) together with an authorization code by SMS to the user. After verifying the correctness of the transaction data, the user returns the authorization code via the browser to confirm that the integrity of the transmitted transaction data. In case of an attack, it is assumed that the user will notice when transaction data have been changed, so the attack can be stopped by canceling the transaction. This method can in theory provide strong data authentication, but it puts a relatively high cognitive load on the user for verifying the transaction data. In a study, it has been shown that about 30% of users fail to notice when transaction data in an SMS message have been changed, which means that 30% of attacks would succeed even in the presence of SMS authorisation [13]. The problem with SMS-authorisation is poor security usability, which fortunately can be solved with Lucidman as is explained below.

Figure 8 illustrates a simple ceremony for data origin authentication, i.e. to ensure that what is displayed on the VDU corresponds to what is being transmitted to other parties in online transactions. The method assumes that the user has an OffPAD with integrated camera, OCR (Optical Character Recognition) and communication functions. The user first captures a screenshot from the VDU with the OffPAD camera, then uses the OffPAD to recover the displayed data from the image through OCR, and finally to compute a MAC (Message Authentication Code) which is sent along with the original transaction data. The MAC enables the recipient server to authenticate the received original data.

The actions/messages of the ceremony are described in Table 5.

Even though it is assumed that the client platform is infected, it is easy to see that attackers will not be able to falsify the transaction data undetected. Falsified transaction data would produce a MAC mismatch, which would be discovered by the server in (8).

In order to successfully falsify data, the attacker would have to compromise both the client platform and the OffPAD simultaneously. Since the OffPAD is offline, it is assumed that the OffPAD will not be exposed to threats from the Internet. A more detailed threat analysis is provided in the next section.

Security analysis of the OffPAD

The OffPAD is considered as a trusted device, which means that it is assumed to provide the authentication services described in the sections above, more specifically 1) cognitive server authentication, 2) user authentication with never present cleartext passwords, and 3) cognitive data authentication. Should the OffPAD fail, then the mentioned authentication services would be invalidated. In this section we discuss possible threats against the OffPAD device and its integration into the security ceremonies described above. The discussed threats are summarized in Table 6.

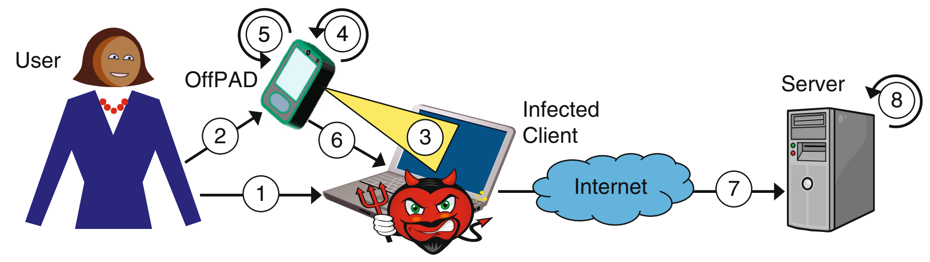

Figure 8 Ceremony for data authentication with the OffPAD and with malware-infected client.

Table 5 Sequence of messages and actions for data authentication ceremony

Nr.	Message/action description
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:	Snapshot is taken of the text displayed in the browser window on the VDU
4.	The OCR function recovers the transaction data from the snapshot
5.	MAC generation with the transaction data and the user-password as input
6.	OffPAD send the MAC to the client computer
7.	Client computer sends transaction data together with MAC to server
8.	Server verifies that the MAC corresponds to the received transaction data.

Unauthorized use of device

If an unauthorized person were able to use the functionality of the OffPAD device, the attacker might be able to activate the user authentication function as well as the data authentication function. This could have serious consequences, e.g. the masquerading as the user and executing fraudulent transactions. Furthermore, the attacker could breach the device integrity by making unauthorized modifications and configurations changes on the device, such as defining assigning existing petnames to fraudulent websites. In case the legitimate owner is unaware of the integrity breach and continues to use the OffPAD, the consequence could e.g. be that the owner is tricked to access fraudulent websites in the belief that they have been authenticated as legitimate websites by the petname system on the OffPAD.

To counter these threats, the OffPAD must authenticated its owner to unlock its functionality. User authentication must be based on owner credentials that can consist of PIN (something you know), on various biometrics modalities (something you are or do) such as fingerprint scan, face and iris recognition, touch and handling dynamics, or on something you have (physical token). Owner authentication and the design of owner credentials are outside the scope of this article. Suffice to say that it must be designed to provide adequate security assurance and usability.

Unauthorized use of the device can also be based on tampering. The OffPAD device must therefore be resistant to physical and electronic tampering, but these aspects are also considered to be out-of-scope.

Unauthorized use of unlocked device

If the owner unlocks the device and then leaves it unattended, it can be abused to access all protection realms to which the owner is authorized. The impact level of this abuse can be reduced by introducing a session timeout on the device. This way, an attacker has

Table 6 Threats against the OffPAD

Nr.	Threat name
1	Unauthorized use of device
2	Unauthorized use of unlocked device
3	Loss of owner credentials
4	Loss or destruction of device
5	Interception or modification of OffPAD communication
6	Data injection attacks
7	Ceremony protocol attacks

limited time to commit unauthorized actions. While he is also able to modify and delete identities on the device, there is no direct way for the attacker to recover the passwords on the device. Another prevention mechanism could be the use of continuous owner authentication through touch and handling dynamics.

Loss of owner credentials

By loss of owner credentials we mean that the owner is unable to use his or her credentials to activate the device, not that an attacker has gained control of the owner credentials. The owner can lose the ability to use credentials for various reasons, e.g. when forgetting the PIN or the fingers are damaged or too dirty to be recognized by the fingerprint scanner. In such situations there must exist a back-up procedure to unlock the device. This could e.g. be based on having separate primary and secondary owner credentials, where e.g. unlocking keys similarly to PUK (PIN Unlocking Key) for SIM cards, and/or on a physical key/token can represent secondary credentials to be used in case the primary credentials fail for some reason. The fundamental principle for secondary credentials is that they must provide equal or stronger security assurance than the primary owner credentials, but the usability might be inferior to that of the primary credentials. Other than these general comments, the topic of owner credentials recovery is outside the scope of this article.

Loss or destruction of device

Incidents of loss, theft or destruction of mobile devices occur relatively frequently, rendering the functions of the device unavailable. In such situations it is important that replacement devices can be obtained and configured relatively easily. For the OffPAD, this would require that a new OffPAD device can be configured with e.g. user passwords and petnames in an user friendly and secure manner. This requirement must be based on back-up procedures which e.g. to a storage device via a USB connection. The back-up data can be stored in cleartext format or in encrypted form, where both methods need to be studied and analysed further with regard to security assurance and usability. The design and analysis of back-up procedures are outside the scope of this article.

Interception or modification of OffPAD communication

As mentioned above the OffPAD must be able to communicate with client terminals e.g. trough a NFC radio link, or through a USB cable. This communication channel can be subject to passive interception or active modification, both of which represent a security and privacy threat. However this threat is relatively minor, as it is already assumed that the client platform is infected by malware. Thus, attacks against the communication channel would not cause significantly more damage than can already be caused by malware on the client platform. The only relevant aspect is that the possibility of attacking the communication channel represents a strengthening of threats that already exist. For example, in case the client platform is not infected by malware, an external third party could eavesdrop on the communication and thereby pose a threat to user privacy, but would be unable to attack authentication functions or the transactions themselves. The possibility of attacking the communication channel between the OffPAD and the client terminal requires physical proximity, which makes the attack difficult to execute in general and relatively easy to discover.

Data injection attacks

Data injection consists of sending malformed data to the OffPAD with the aim of causing harm to the OffPAD. Data injection could be generated by malware on the client computer with which the OffPAD communicates, or by malicious computers in physical proximity to the OffPAD. In case of software vulnerabilities on the OffPAD malformed data could damage the integrity of the OffPAD or cause data leakage, similarly to SQL injection attacks. It is therefore crucial that the format of received and transmitted data is well defined and validates, and that any malformed data can be properly handled and rejected. The data format will have XML at its core, but the detailed specification and the requirements for interpretation are outside the scope of this article.

Ceremony protocol attacks

Attacks can be directed at other elements of the security architectures than the OffPAD itself. The three proposed authentication ceremonies involve 4 entities and consists of up to 8 communication instances between those entities. A formal analysis if these ceremonies requires significant effort and is the subject of studies during the research activities of the OffPAD project started in 2014.

User experiment

A user experiment was conducted to assess whether cognitive server authentication with a petname system can improve users' resistance to phishing attacks, and to find out whether a petname system on an external device like the OffPAD is something that people would like to use. The experiment was conducted as part of a Master's project [36] at the University of Oslo.

Type of experiment

Data from user experiments can be collected qualitatively or quantitatively. Both methods have their advantages and disadvantages. Qualitative data gives insight, but can be challenging to analyse. Quantitative data can be subject to statistical analysis, but if taken out of context can often be misleading.

Several elements indicated that qualitative data would be best for our study. From a practical point of view we had to use one special computer with one specific connected device, which would make it difficult to let multiple participants undergo the experiment simultaneously to collect a sufficient amount of statistical data. The most important reason for collecting qualitative data was that it would give us more insight into the participants' thinking than statistical data could do.

For collecting qualitative data it is possible to use different types of methodologies [37], e.g. Conversation analysis, Analytic Induction, or Discourse Analysis. We did a simple case study combined with a usability experiment, where the user was observed while using the system, followed by an interview. Before doing the experiment we received the permission from the Data Protection Office for Research at the University of Oslo.

Selection of participants

While there is no agreed number of subjects in a qualitative study, Nielsen [38] states that a usability experiment only needs 5 subjects to spot the most relevant usability issues. The results can then be used to improve the prototype which in this way can be iteratively tested and improved in an economical way. Our purpose was not to iteratively improve our prototype, but simply to see whether cognitive server authentication as implemented in the prototype can give the intended positive security effect, as a proof of concept. Only a few participants would be needed to observe that effect. Using a high number of participants could be meaningful for validating the usability of the prototype when it is more finished and close to being introduced in a market.

The general consensus among experts is that the required number of participants depends on multiple factors [39]. Some of the factors to consider are the depth of the interview, what we want to get out of the study and the type of participants in the experiment. There is also some more tangible factors to take into account, like resources and time. When setting a number of participants the first factor we considered was the type of participants to invite. The target user group for an external device like the OffPAD consists of persons who regularly use services on the Internet where the goal of the experiment was to determine whether cognitive server authentication could provide increased security. With a prototype OffPAD available, the second goal was to find out if participants would be happy to use it. We concluded that six participants would be sufficient to get an indicative answer to these question.

The participants in this test were mainly students of informatics, but we chose to also include two students who did not have informatics as their field of study, to see if they would have other opinions than the rest. The results indicated that there was no clear difference.

Experiment set-up

The purpose of cognitive server authentication is to make it easy to detect fake websites. So for this experiment we created 4 fake sites to mimic 4 real websites. The subjects where asked to log in with a non existing account, because the petname system does not depend on having a user account on a specific website. Using real accounts would be problematic for three reasons. The first is that we would have to create accounts for non-existing users which would probably breach usage policies. Secondly, it was easier to implement a fake website with a page that says "Wrong user name or password" than to make a page that mimic the content served on the real site. Finally we did not want the users to enter their own user names and passwords for the real websites because it would make the subjects unwilling to participate. The fake sites consisted of a login-page and an account-refused page. The login-page of each of the fake sites were made to look almost exactly the same as that of the corresponding real websites. The domain names used are given in Table 7 below.

The real websites had domain names with .com as top level domain, whereas the corresponding fake sites had domain names with .ccm (the letter 'o' is replaced with the letter 'c') as top level domain. As .ccm does not exist as top level domain, the fake domains used

Table 7 Domain names for real and fake sites

Real domains	Fake domains
nb-no.facebook.com	nb-no.facebook.ccm
accounts.google.com	accounts.google.ccm
www.linkedin.com	www.linkedin.ccm
twitter.com	twitter.ccm

in this test were simply added to the host-file in Windows on the computer used for the experiment. This file is checked for a domain name to IP-address link before Windows asks the network's DNS-servers. These domain names pointed to a web server controlled by us.

All the real websites used TLS, but the fake websites did not. We considered whether to add server certificates for the fake sites, by generating a self-signed CA certificate and add it to the web browser repository. We concluded that it was not necessary, and that it would in fact be interesting to observe the participants' reaction to the lack of TLS for the fake sites.

Under the development of the fake sites and during the experiment Google Chrome was used as web browser. Most of the sites that we made copies of used JavaScripts to check if the domain name was the correct and redirect if not. So all of the images were saved locally and the JavaScripts removed. We made a change to the web server and the Chrome extension so that only this browser on this machine could open the phishing sites, thus making it impossible for anybody else to discover the phishing sites through the Internet.

Interview guide and questions

An interview guide was developed to introduce the participants to the experiment. The guide was based on Pathfinder International's guide to designing and conducting interviews [40]. The whole interview was conducted in Norwegian. A short summary in English is provided below.

The first part was an introduction to the experiment and interview. Where the subject was informed that the interview was audio recorded, where the recording would be kept confidential and the published results would not identify the subject. The subject was also informed that he or she was free to answer each question and could decide to terminate the interview at any time. It was important to respect the persons who offer their time for the experiment. The participants received a brief introduction to the petname system. Then the participant was presented with four obfuscated website links, and was asked to access each of the corresponding websites, log on with a given user name and password and add a petname for each website. The user name and password did not correspond to real accounts so the participant was simply given the message "Wrong user name and password" as expected. The participants were informed that a valid account was not a requirement for the petname system to work.

Then the participants were divided into two groups. The first group (hereafter refereed to as group A) did Exercise I first and then Exercise II, and the other group (refereed to as group B) did it in the opposite order.

After finishing these exercises, the participant was interviewed about their experience. The question are given below. The session concluded with an opportunity for the participant to add any last comments or remarks.

- **Exercise I, with OffPAD:** The participant sits down in front of the computer with a web browser running and the OffPAD connected. The participant is then invited to click on 4 different links in random order and to try to log on to the corresponding websites. One of the sites is fake, but it is different from the fake site of Exercise II. During this exercise the participants also uses the OffPAD connected to the computer.

- **Exercise II, without OffPAD:** The participant sits down in front of a computer with a web browser running. The participant is then invited to click on 4 different links in random order and to try to log on to the corresponding websites. One of the sites is fake, but it is different from the fake site of Exercise I. During this exercise the participant does not use the OffPAD.

Some of the questions were aimed at both checking if the participant has understood the petname system, and to see whether the petname system was easy to use. Then there were questions aimed at getting the participants' feelings and thoughts around their use of the petname system on the OffPAD. Finally there were questions aimed at determining the degree of awareness about possible phishing attacks with or without the petname system. We used the term 'OffPAD' in the questions to refer to the device they used, and the term 'petname system' when referring to the service it provided. The questions used in the experiment were the following:

1. While not using the OffPAD
   - Did all web sites work normally?
   - If you noticed anything unusual, what was it?
2. While using the OffPAD
   - Did all web sites work normally?
   - If you noticed anything unusual, what was it?
   - Would you have noticed the phishing site without the OffPAD?
   - If you think everything was normal, why did you not notice the warning on the OffPAD?
3. General questions to the experiment
   - How did your vigilance change after identifying the first phishing page?
   - How did your sense of security change when using the OffPAD?
   - Were you more or less aware during the experiment than usual?
   - How do you think the use of an OffPAD will impact your daily Internet use?
4. What would you think about using the Petname system...
   - if it was on your smart-phone?
   - if it was an own device?
   - if it was on a multi-purpose authentication device?
5. Normally when accessing and logging on to websites, how aware of phishing attacks are you?
6. How do you consider the possibility for phishing attacks against your person?
7. What do you think about the usability of the OffPAD prototype?
8. Is there something else you want to add?

Results

The experiment was conducted in the period 9 – 13 March 2013, and had six participants. Given the relatively low number of participants the statistics from the test are not representable for any group of people. However, their opinions and feedback about the system is valid. For some questions it is worth observing the number of participants who took one or the other stand, because it indicates whether it was a single point of view or if it reflected a more common opinion.

As a general observation, none of the participants reported any significant usability problems. Subject to the limited confidence provided by the small number of participants, it seems that the OffPAD prototype's user-friendliness is satisfactory.

Of the six persons we interviewed, only two did actually check the URL and discovered both the phishing sites before they used the user name and password. There was also one who saw some error in a graphical element, but when he or she checked it, he or she did not notice the change in the domain name.

None of the participants reacted to the missing TLS indicator in the web browser. All of the real pages used TLS, but none of fake ones did. A vigilant expert would maybe detect the phishing sites from the missing TLS. However, simply observing TLS on a website is no guarantee that the website is not a phishing site. We could very well have implemented the fake websites with certificates and TLS, and then even an expert would fail to detect the fake websites if he or she only used the presence of TLS as indicator.

In group B, which first did the exercise with the OffPAD, participants were generally more aware of possible attacks. This is because after detecting one phishing site they became more alert. It could be argued that the Hawthorne Effect [41] was at play, where a subject is more alert or efficient because they know they are being observed. However, this experiment was too small and simple to say anything about the possible influence of the Hawthorne Effect.

Without OffPAD

When not using the OffPAD and the petname system, four out of the six participants noticed the phishing site. One participant in group A said that he or she always checked the URL. In group B two of the participants said they detected the phishing site because they paid special attention to it during the experiment. Both claimed in their answers that they noticed something they thought was abnormal, but that that later turned out to be perfectly normal. The first person thought the URL was too long even if it was exactly the same length as the original, the other thought he read 'fakebook' in the URL even though the URL said 'facebook' as expected. The last participant in group B who noticed the phishing site without the OffPAD did so right away, he or she did also pinpoint the phishing site before trying to log into it in the exercise with the OffPAD.

With OffPAD

When using the OffPAD and the petname system, all participants detected the phishing site. The subjective feeling of security experienced with the OffPAD varied from indifference to a heighten sense of security with the OffPAD. Those who had an equal sense of security said the OffPAD did not add much to their own vigilance. The subjects who felt safer described the OffPAD as providing extra security protection when surfing the Internet. One of the important points mentioned by one of the subjects was that a petname system might make him or her more reckless on the Internet. This comment indicates that if the petname system is poorly implemented (e.g. not responding to GET-requests) it can give a false sens of security, making users an easier target for phishing attacks.

Feedback about the device

All the participants were negative to using an external device that only provided a petname system. Two participants mentioned that they would have considered it if it was in the size of a key chain. A third participant made the comment "Who should pay for this anyway?". It is a fair comment as such a device might be just as expensive as a simple smart phone.

Most of the participants were positive to a multi-purpose authentication device, as this would not require then to carry a separate device just for the petname system. One had an different approach to this, he or she said they would be more skeptical to a device with all of their credentials. "If it gets lost then you have more problems", referring to the case of theft where someone takes your device and manages to unlock it to access every service you are using.

The most preferable solution for all of the participants was to user their mobile phone in one way or another. One added "If it goes seamlessly, then I would use some time in the beginning to set it up. (...) As long as I only need to do define a petname for each website once".

Resistance to Phishing

Answers from the participants indicated that phishing sites are not considered to be a significant problem for them. This was because they either did not care or they did not believe they could not easily be tricked to access a phishing website since they mostly relied on their own bookmarks or typed the domain name themselves when logging in to a site.

There was one exception, where a participant considered phishing as a big threat. "You can lose much if you get phished". This person also mentioned indirectly to have been tricked by a phishing attack. This was also the person who discovered the phishing site before the petname system would do it. All participants considered themselves to be very unlikely targets of a directed spear-phishing attack, because they were only students without access to very sensitive information.

Usability of the prototype OffPAD

All of the participants was positive to the prototype, ranging from "It's quite alright" to "Surprisingly good". The size of the device was also mentioned here, where the participants typically commented that it has to be made smaller. The participants made positive remarks about its response time. One pointed out "It was intuitive and easy to use (...) for me it was no problems, but if you are not a technical person it might be a bit hard". It did not seem to be any problem to use the device for the two non-technical participants either.

Discussion and conclusion

It is challenging to provide trusted interactions between users and SPs in an environment where client platforms are infected by malware. Local user-centric identity management is a novel concept that provides a solution to this challenge, and is based on using an OffPAD which is an offline personal authentication device.

Complexity is the enemy of security, so the OffPAD should be simple and be limited in functionality. In contrast to smart phones or tablets that are designed to be open and have maximum connectivity and flexibility, the OffPAD should be a closed platform and have strictly controlled connectivity. These design principles are aimed at reducing the attack surface. The challenge is to offer adequate usability despite these constraints.

Communication between the client computer and the OffPAD requires drivers and software installed on the client computer. In case the client computer has been infected with malware the communication with the OffPAD could give attackers some information about the OffPAD, and potentially an opportunity to compromise the OffPAD itself. It is therefore important that the OffPAD enforces strictly controlled connectivity, e.g. by not having a direct connection to the Internet, and by enforcing time limits for connections to client computers or to other systems. The user should explicitly activate a connection, the OffPAD should clearly indicate when a connection is active, and should indicate when the connection ends. Typically, every connection should not last longer than one second at a time, just enough to exchange authentication information. If the communication takes much longer, the user is likely to leave the OffPAD connected to the client computer which would introduce new attack vectors.

The proposed ceremonies are designed to cross-check the processes of server authentication, user authentication and data authentication so that malware on the client platform can not break the integrity of these processes without detection. Malware could of course disrupt the processes and thereby cause a denial of service attack, but to protect against this form of attack is not within the scope of our proposed solutions.

Carrying an OffPAD as a separate device could represent a barrier for many people, so it is worth investigating whether the OffPAD could be integrated in a smart phone in a secure way. Modern mobile phones, or smart phones, are packed with advanced features and must be considered a 'general purpose computing platform'. This certainly provides great flexibility and support for many new business models, but it also opens up many new attack vectors. From 2010 to 2011 Juniper MTC reported a 155% increase in malware on mobile phones [42]. It should be noted that all the different mobile phone operating system manufacturers are trying to make their system more secure. At the same time, the market pressure enforces them to provide more connectivity and more flexibility into their devices, which necessarily also introduces new vulnerabilities. This makes a normal mobile phone unreliable for high security applications. A candidate smart phone is the 'TPH-One' produced by TazTag.

The integration of the OffPAD with a smart phone must basically be to implement within the same physical device two separate sub-systems, one being the OffPAD and the other the smart phone. Given the limited size of a smart phone, it would be practical to let both sub-systems share the I/O (input & Output) channels such as scree, microphone and loudspeaker.

However, a compromised smart phone could enable attackers to trick the user into believing that she is interacting with the OffPAD sub-system. It is therefore crucial that the user can be assured that she is interacting with one or the other sub-system, and not confuse the two sub-systems. To achieve this goal the user must be able to select each sub-system specifically, and the platform must clearly indicate which sub-system the user is interacting with at any time.

In this article we described local user-centric identity management as an architecture for providing robust authentication services of various types and modalities. In particular we have described the OffPAD as a personal device that supports mutual authentication between user and server, as well as authentication of data sent from the user to the server. These solutions can be integrated with many of the existing online security services and can provide trusted interaction even in environments where client systems are infected by malware.

The level of authentication assurance provided by our solutions can be compared the AAL (Authentication Assurance Levels) according to various frameworks for user authentication such as US EAG (Electronic Authentication Guideline) [43], the Norwegian FANR (Framework for Authentication and Non-Repudiation in Electronic Communication with and within the Public Sector) [44], the Australian NeAF (National e-Authentication Framework) [45], EU STORK (Secure Identity Across Borders Linked) [46], the Indian ePramaan Framework for e-Authentication [47], or ISO 29115 Entity authentication assurance framework [48]. According to these frameworks, our user authentication solution would be placed on AAL 3 (Authentication Assurance Level) on a scale from 1 to 4, where AAL 4 is the highest. However, our solutions could also support AAL 4 by implementing support for user certificates in the OffPAD. It can be mentioned that there are currently no frameworks for server authentication assurance levels, although it has been proposed [20,49].

Initial evaluations of the different authentication services have been carried out on the prototype OffPAD device. User experiments have already demonstrated that the OffPAD has the potential for good usability. Future work will focus on solutions for back-up and for updating software and the operating system on the OffPAD. Also to be investigated is whether the OffPAD system can be integrated as part of another device such as a smart phone, without compromising the security of the OffPAD. The goal is to find solutions that offer both strengthened security as well as increased usability, which thereby provides a robust basis for trusted interaction in online environments. Finally, it can be mentioned that research on local user-centric identity management continues in the OffPAD project during 2014-2016, supported by the *eurostars* research programme.

Endnotes

[a]Security Assertions Markup Language
[b]PKIX: Public-Key Infrastructure based in X.509 certificates [50]
[c]TLS: Transport Layer Security
[d]TazTag is a Lucidman project partner.
[e]Called *decentralized* in [24]
[f]Called *secure* in [24]
[g]Called *human-meaningful* in [24].
[h]The DAA challenge-response method is not restricted to HTTP.
[i]User interaction is necessary in situations where no identity or multiple identities are available for the user to authenticate with, the password is wrong, or another error appears.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
AJ edited the article and provided sections Introduction and Background and Related Work. KAV and AJ provided section Authentication Classes and Modalities, KAV, AJ and LM provided section OffPAD – The Offline Personal Authentication Device. AJ, CR, JD, KEH and PT contributed to section Local User-Centric IdentityManagement. HK and AJ provided section Security Analysis of the OffPAD. KAV provided section User Experiment, and all authors contributed to the Discussion and Conclusion. All authors read and approved the final manuscript.

Acknowledgements
The work reported in this article has been partially supported by eurostars project number E!8324 OffPAD, and by the Eureka project number 7161 LUCIDMAN.

Author details
[1]University of Oslo, Oslo, Norway. [2]ENSICAEN, Caen, France. [3]TazTag, Rennes, France. [4]Nets Norway AS, Oslo, Norway. [5]Domeneshop AS, Oslo, Norway. [6]CEV, Saint Lo, France. [7]Tellu, Asker, Norway. [8]Vallvi, Oslo, Norway.

Received: 22 July 2013 Accepted: 8 August 2014
Published online: 08 January 2015

References

1. PandaLabs (2012) PandaLabs Quarterly Report, Q2. http://press.pandasecurity.com/wp-content/uploads/2012/08/Quarterly-Report-PandaLabs-April-June-2012.pdf Accessed 2012-11-01
2. Ellison C (2007) Ceremony Design and Analysis. Cryptology ePrint Archive, Report 2007/399. http://eprint.iacr.org/2007/399
3. Stajano F (2011) Pico: No more passwords! In: Proceedings of the 19th International workshop security protocols. pp 49–81
4. Mannan M, van Oorschot PC (2011) Leveraging personal devices for stronger password authentication from untrusted computers. J Comput Secur 19(4):703–750
5. Laurie B, Singer A (2009) Choose the red pill and the blue pill: a position paper. In: Proceedings of the 2008 new wecurity paradigms workshop. ACM. pp 127–133
6. TCG (2014) Trusted platform module library specification, family "2.0", level 00, revision 01.16. Trusted Computing Group, Beaverton, Oregon, USA
7. Jøsang A (2012) Trust Extortion on the Internet. In: Proceedings of the 7th International workshop on security and trust management (STM 2011). Springer, Copenhagen
8. Jøsang A, AlFayyadh B, Grandison T, AlZomai M, McNamara J (2007) Security usability principles for vulnerability analysis and risk assessment. In: The proceedings of the annual computer security applications conference (ACSAC'07)
9. Jøsang A, Møllerud PM, Cheung E (2001) Web security: The emperors new Armour. In: The proceedings of the European conference on information systems (ECIS2001)
10. ITU (1991) Recommendation X.800, Security Architecture for Open Systems Interconnection for CCITT Applications. International Telecommunications Union (formerly known as the International Telegraph and Telephone Consultantive Committee), Geneva. (X.800 is a re-edition of IS7498-2)
11. Bodmer S (2012) SpyEye being kicked to the curb by its customers?. Research Note, Damballa Inc. http://www.damballa.com
12. Nayyar H (2010) Clash of the Titans: ZeuS v SpyEye. SANS Institute InfoSec Reading Room
13. AlZomai M, AlFayyadh B, Jøsang A, McCullag A (2008) An experimental investigation of the usability of transaction authorization in online bank security systems. In: The Proceedings of the Australasian information security conference (AISC2008)
14. Jøsang A, Pope S (2005) User-centric identity management. In: Clark A (ed). Proceedings of AusCERT 2005
15. Klevjer H, Jøsang A, Varmedal KA (2013) Extended HTTP digest access authentication. In: Proceedings of the 3rd IFIP WG 11.6 Working conference on policies & research in identity management (IFIP IDMAN 2013). Springer, London
16. Varmedal KA, Klevjer H, Hovlandsvåg J, Jøsang A, Vincent J, Miralabé L (2013) The OffPAD: requirements and usage. In: The 7th International conference on network and system security (NSS 2013)
17. Baloul M, Cherrier E, Rosenberger C (2012) Challenge-based speaker recognition for mobile authentication. In: IEEE Conference BIOSIG
18. Beton M, Marie V, Rosenberger C (2013) Biometric secret path for mobile user authentication: a preliminary study. In: International conference on mobile applications and security management (ICMASM)
19. Arkko J, Nikander P (2002) Weak authentication: how to authenticate unknown principals without trusted parties. In: Security Protocols Workshop. pp 5–19
20. Jøsang A, Varmedal KA, Rosenberger C, Kumar R (2012) Service provider authentication assurance. In: Proceedings of the 10th annual conference on privacy, security and trust (PST 2012)
21. Ferdous MS, Jøsang A (2013) Entity Authentication & Trust Validation in PKI using Petname Systems. In: Elçi A, et al. (eds). Theory and practice of cryptography solutions for secure information systems (CRYPSIS). IGI Global, Hershey, PA, USA
22. Ferdous MS, Jøsang A, Singh K, Borgaonkar R (2009) Security Usability of Petname Systems. In: Proceedings of the 14th Nordic workshop on secure IT systems (NordSec 2009)
23. Stiegler M (2005) Petname Systems. Technical Report HPL-2005-148, HP Laboratories Palo Alto. http://www.hpl.hp.com/techreports/2005/HPL-2005-148.pdf
24. Wilcox-O'Hearn BZ (2005) Names: Decentralized, secure, human-meaningful: Choose two. http://shoestringfoundation.org/~bauerm/names/distnames.html
25. Miller MS (2000) Lambda for Humans: The PetName Markup Language. Resources library for *E*, http://www.erights.org/elib/capability/pnml.html
26. Close T (2004) Trust Management for Humans. Waterken YURL, WaterkenInc. http://www.waterken.com/dev/YURL/Name/
27. Franks J, Hallam-Baker P, Hostetler J, Lawrence S, Leach P, Luotonen A, Sink E, Stewart L (1999) RFC 2617 – HTTP Authentication: Basic and Digest Access Authentication. IETF. http://www.ietf.org/rfc/rfc2617.txt
28. Klevjer H (2013) Requirements and Analysis of Extended HTTP Digest Access Authentication. Master's thesis, University of Oslo, Norway
29. Berners-Lee T, Fielding R, Frystyk H (1996) RFC 1945 – Hypertext Transfer Protocol – HTTP/1.0. IETF. http://www.ietf.org/rfc/rfc1945.txt
30. Franks J, Hallam-Baker P, Hostetler J, Leach P, Luotonen A, Sink E, Stewart L (1997) RFC 2069 – An Extension to HTTP : Digest Access Authentication. IETF. http://www.ietf.org/rfc/rfc2069.txt
31. Alzomai M, Alfayyadh B, Jøsang A (2010) Display security for online transactions. In: The 5th International conference for internet technology and secured transactions (ICITST-2010)
32. Jøsang A, Povey D, Ho A (2002) What you see is not always what you sign. In: Proceedings of the Australian UNIX and Open Systems Users Group Conference (AUUG2002)

33. Kain K, Smith SW, Asokanm R (2002) Digital signatures and electronic documents: a cautionary tale. In: Proceedings of IFIP conference on communications and multimedia security: advanced communications and multimedia security
34. Spalka A, Cremers AB, Langweg H (2001) The fairy tale of 'What You See Is What You Sign - Trojan Horse attacks on software for digital signatures. In: IFIP Working conference on security and control of IT in society-II (SCITS-II)
35. Weber A (1998) See What You Sign: Secure Implementations of Digital Signatures. In: Proceedings of the International Conference on Intelligence and Services in Networks (ISN 1998). pp 509–520
36. Varmedal KA (2013) Cognitive Entity Authentication with Petname Systems. Master's thesis, University of Oslo, Norway
37. Online QDA Project (2011) Methodologies. http://onlineqda.hud.ac.uk/methodologies.php (visited 13.03.2013)
38. Nielsen J (2000) Why you only need to test with 5 users. http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/ (visited 25.02.2013)
39. Baker SE, Edwards R (2012) How many qualitative interviews is enough? Online. http://eprints.ncrm.ac.uk/2273/#.UF7md_-Mp-E.citeulike (visited 11.12.2013)
40. Boyce C, Neale P (2006) Conducting in Depth Interviews: A Guide for Designing and Conducting In-Depth Interviews for Evaluation Input. Pathfinder International. http://www.pathfind.org/site/DocServer/m_e_tool_series_indepth_interviews.pdf?docID=6301, (visited 11.12.2013)
41. McCarney R, Warner J, Iliffe S, van Haselen R, Griffin M, Fisher P (2007) The Hawthorne effect: a randomised, controlled trial. BMC Med Res Methodol 7(1):30
42. Juniper (2011) Juniper mobile threat report 2011. Technical report, Juniper Networks, Inc.
43. Burr WE, Dodson DF, Newton EM, Perlner RA, Polk WT, Gupta S, Nabbus EA (2013) Electronic Authentication Guideline – NIST Special Publication 800-63 Rev. 2. Technical report, National Institute of Standards and Technology
44. Ministry of Government Administration Reform M (2008) Framework for Authentication and Non-Repudiation in Electronic Communication with and within the Public Sector (in Norwegian: Rammeverk for autentisering og uavviselighet i elektronisk kommunikasjon med og i offentlig sektor). Technical report, Norwegian Gov.
45. Department of Finance and Deregulation (2009) National e-Authentication Framework (NeAF). Australian Government Information Management Office, Canberra
46. Hulsebosch B, Lenzini G, Eertink H (2009) Deliverable D2.3 - STORK Quality authenticator scheme. Technical report, STORK eID Consortium.)
47. Ministry of Communications and Information Technology (2012) e-Pramaan: Framework for e-Authentication, Government of India, Delhi
48. ISO (2013) ISO/IEC 29115:2013. Entity Authentication Assurance Framework. ISO, Geneva, Switzerland
49. Jøsang A (2014) Assurance requirements for mutual user and service provider authentication. In: 3rd International workshop on quantitative aspects in security assurance (QASA 2014)
50. ITU (1997) Recommendation X.509 V3, The Directory: Authentication Framework (also Known as ISO/IEC 9594-8). International Telecommunications Union, Telecommunication Standards Sector(ITU-T), Geneva