The OFFPAD : Requirements and Usage

Kent Varmedal, Henning Klevger, Joakim Hovlandsvag, Andun Josang, Johanne Vincent, Laurent Miralabé

The OffPAD: Requirements and Usage*

Kent Are Varmedal[1], Henning Klevjer[1], Joakim Hovlandsvåg[1], Audun Jøsang[1], Johann Vincent[2], and Laurent Miralabé[3]

[1] Department of Informatics, University of Oslo
[2] ENSICAEN, GREYC, F-14032 Caen, France
[3] TazTag, 2 Allée Gustave Eiffel, Campus de Ker Lann, 35170 Bruz, France
{kentav, hennikl, joakimsh, josang}@ifi.uio.no
johann.vincent@ensicaen.fr, lm@taztag.com

Abstract. Strong authentication for online service access typically requires some kind of hardware device for generating dynamic access credentials that are often used in combination with static passwords. This practice have the side effect that users fill up their pockets with more and more devices and their heads with more and more passwords. This situation becomes increasinlgy difficult to manage which in turn degrades the usability of online services. In order to cope with this situation users often adopt insecure *ad hoc* practices that enable them to practically manage their different identities and credentials. This paper explores how one single device can be used for authentication of user to service providers and server to users, as well as provide a range of other security services.

1 Introduction

Over the last decade there has been a radical evolution in procedures for authentication. Before the Internet revolution and the invention of the World Wide Web, system passwords were typically simple, physical keys were still the most widely used method to access offices, and bills were still being paid by signing pieces of paper. Today, computer systems are globally interconnected and exposed to millions of host, physical access control to offices is typically based on electronic access cards, and sensitive documents e.g. for financial transactions are now digital, which require digital signatures instead of hand-written signatures. In order for this evolution to remain secure and sustainable there are strong requirements for authentication of entities. By taking into account the distinction between system entity (client or server) and legal/cognitive entity (person or organisation) there are in fact two entities on each side of a communication session, as illustrated in Fig.1.

The distinction between the human user and the client system on the user side, as well as between the SP organisation and the server system on the server side, leads to the conclusion that each of the 4 entities can be authenticated in 2 different ways leading to 8 different classes of peer entity authentication between the two sides, as illustrated in Fig.1 and described in Table 2 and Table 1 below.

* 7th International Conference on Network and System Security (NSS 2013). Madrid, June 2013.

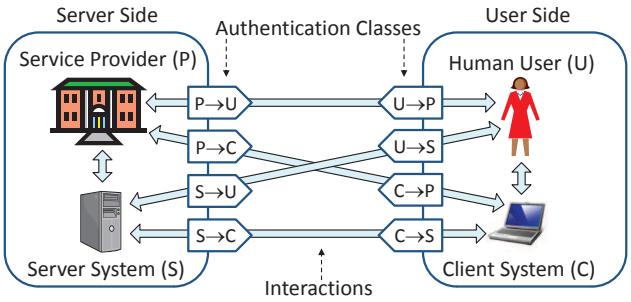

Fig. 1. General entity authentication classes

Class	Authentication of server 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 System (S) authentication by the Human User (U) (which can be called *Cognitive Server Authentication*)
[S → C]	Server System (S) authentication by the User Client (C)

Table 1. Classes of authentication of server-side entities by user-side entities

Class	Authentication of user side entities
[U → P]	Human User (U) authentication by the Service Provider (P)
[U → S]	Uuman User (U) authentication by the Server System (S) (commonly called *user authentication*)
[C → P]	User Client (C) authentication by the Service Provider (P)
[C → S]	User Client (C) authentication by the Server System (S)

Table 2. Classes of authentication of user-side entities by server-side entities

For online services applications the entity authentication classes [S → U] and [U → S] are the most relevant because of the need for end-to-end security. In the typical case where a human user accesses an online service, semantic end-to-end communication takes place between the human user (U) and the server system (S). It is therefore pragmatic to require mutual authentication between those two entities. Traffic encryption and authentication between the server system (S) and user client (C) typically provides communication confidentiality, but can not provide cognitive server authentication in a meaningful way.

In case of one-factor user authentication based on static passwords, corporate computer networks and online service providers typically require long, complex and unique passwords. Two-factor authentication is often required for access to sensitive online services and for physical access to buildings, where one of the factors is a dynamic credential generated by a device, such as an OTP (One-

Time Password). Two-factor authentication typically combines "something you know" (the password) with "something you have" (the authentication device).

Personal authentication devices combined with static passwords have been used by online banks, e-governments and other online service providers for several years. The device is usually an OTP-generator, or a smart card (with reader) of varying complexity. The reason for service providers to use two-factor authentication is to have a higher level of authentication assurance. Some banks offer smart phone authentication applications to log into their services.

In [14], Jøsang and Pope describe the Personal Authentication Device (PAD), a secure device external to the computer. The PAD is used as an identity management system to which the user authenticates once (with a PIN, password or similar), and for one *session*[4]. The user can authenticate to every supported service automatically using the PAD as his identity manager. This is done by transient (replay protected) challenge-response communication between the PAD and the remote server, through the user's computer.

Service providers have identities that also need adequate management. Interestingly, technologies for service provider authentication are very different from those of user authentication, because e.g. user authentication (class [U $\rightarrow$ S]) mainly takes place on the application layer, whereas traditional server authentication (class [S $\rightarrow$ C]) mainly takes place on the transport layer.

In [17], Klevjer *et al.* describe a more secure PAD, the physically decoupled OffPAD, which supports mutual authentication between user and server, as well as user-centric identity management, i.e. secure and usable management of digital identities and credentials on the OffPAD rather than in the user's brain. The OffPAD supports management and authentication of both user and service provider identities. It should be (mostly) offline and contain a secure element, to protect its contents and the privacy of the user.

The idea of having a secure device to do different kinds of authenticated operations is also proposed in a position paper by Laurie and Singer [18]. The so-called "Nebuchadnezzar" is a device that can run multiple security applications such as authentication and transaction signing. Another device similar to the OffPAD is the Pico by Frank Stajano, which is designed to replace passwords everywhere [21]. Stajano describes a number of different solutions where the Pico can be used instead of a password or PIN, such as client authentication to websites, logging into one's home computer or unlocking a screen saver.

In this paper we will first present some requirements for an OffPAD, followed by descriptions of different applications that can be implemented with contemporary technology. Finally, the limitations of the device are addressed.

2 Requirements

The OffPAD is an Offline Personal Authentication Device. The security requirements specified for the Nebuchadnezzar device [18] are aimed at the operating

[4] Limited to either time or connection.

system on the device and can easily be transferred to the concept of an OffPAD. The system requirements they propose to the device is to have a securely built operating system, with a bullet-proof kernel that can run multiple applications which can interact with untrusted systems. The user interface of such a device must be non-spoofable and it should be able to attest to the software is running. The device is not for general purpose (e.g. it does not run a web browser). The device has to support cryptographic functions and being updateable.

Klevjer *et al.* [17] describe that the OffPAD should also have limited connectivity, a secure element and access control. The requirement of limited connectivity can be met by using NFC or other physically activated (contactless) communication. Other (live) means of communications may be appropriate, depending on the required assurance level. The infrastructure for secure messaging and storage in a secure element is described in ISO 7816-4 [13]. For access to the OffPAD, the user must unlock the device by using a PIN, pass phrase, biometrics or other adequate authentication credentials, which prevents unauthorized users from activating the device.

The OffPAD must also be tamper resistant, so that an attacker with physical access to the device cannot easily access information stored on the OffPAD or alter any of the OffPADs characteristics. A possible design of the OffPAD is illustrated in Fig.2 below.

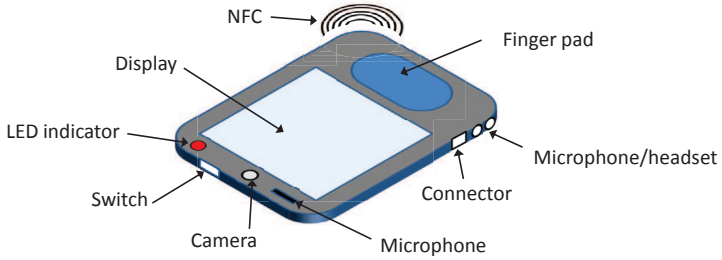

Fig. 2. OffPAD design.

The OffPAD may have several interfaces for communication. Microphone 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.

Communication

Some form of electronic communication is essential for practical integration of the OffPAD into online authentication. Options for electronic communication technologies are listed in table 3. The OffPAD must be restricted with regard to connectivity, and should remain *offline* as much as possible, meaning that it should only be able to communicate securely, in controlled formats and in short, restricted time periods. This decoupling from networks improves security on the

NFC	Short point-to-point connections over limited range.
Bluetooth	Medium range point-to-point communication.
ZigBee	Longer point-to-point connections with low power consumption and transmission range up to 100 m [5].
WiFi	Communication that over the Internet.
USB	Wired connection, can also be used to charge the device's battery (but this will make the OffPAD online).

Table 3. OffPAD communication technologies

device, as it is less vulnerable to outside attacks. Any specific electronic communication technology described in the list above should normally be disconnected, and should only be connected whenever it is needed for authentication or for management of the device. However, the connection should be fast and easy to set up, which might exclude WiFi and Bluetooth.

NFC with a backup USB connection is probably the most suitable communication technology 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 builds upon the concept of Trust-On-First-Use (TOFU), also known as leap-of-faith. On first use there is no cryptographic way to verify that the connection is only between the device and the software, this must be based on trust (or faith) in the physically observed set-up. On the first connection some kind of pairing between the device and computer occurs, so that the subsequent connections can be verified to be between the same device and computer.

3 Services

One OffPAD may be used for a number of different security services simultaneously. With a simple menu system the user can select which service she wants to use. Each service can also be used in different environments (e.g. locations), where either the OffPAD can detect the environment or the user can select it depending on the type of application and communication protocol.

3.1 Digitally Signed Bank Transactions

Normally, when doing online banking, the user can review information on a transaction in the web browser window before confirming and submitting the transaction. The data shown on the screen in such a scenario is vulnerable to change by a man-in-the-browser (MITB). The integrity of the transaction data may be broken (e.g. the amount and receiver account may be changed) before being submitted to the bank's server. With an OffPAD, the transaction data may be signed with the bank's private key and be presented on the device's screen.

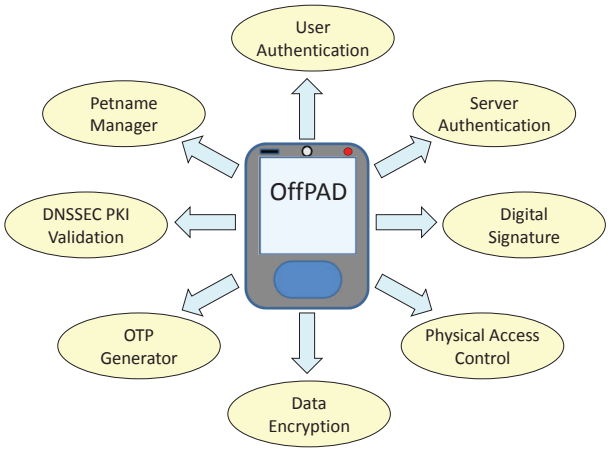

Fig. 3. Multi usage device.

This data may be verified on the trusted device by validating the signature using the bank's public key. Alternatively, a photo of the browser window may be taken with the OffPAD camera, and the text from the photo analysed with OCR (Optical Character Recognition) software. The resulting text may be compared with the expected transaction details. This ensures that the specified transaction data received by the server is exactly as specified by the user, and that it is consistent with what is presented in the client's browser window [1].

Another way to keep the user from signing malformed transaction data, is by having the OffPAD display all the information that is to be signed, e.g. the amount and destination of a bank transfer. Studies have shown that users do not check all the details before confirming such transactions, especially when long rows of numbers are involved, and only a few of them are wrong [2]. The OffPAD would then require a simple keyboard, and the protocol would need to support the OffPAD to be able to not only sign data, but also to generate and modify transactions.

3.2 User authentication

People who frequently use online services will typically accumulate a large number of online identities and related passwords, and managing these quickly becomes impossible. 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[5]), the password is exposed in the computer's memory and is vulnerable to attacks such as key logging or

[5] http://lastpass.com

memory inspection. A solution for password authentication using an OffPAD is proposed by Klevjer *et al.* in [17], consisting of an extension to the original *HTTP Digest Access Authentication* scheme specified as a part of the HTTP standard in [9]. User credentials are stored in a hashed format on both the server and the OffPAD. When 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 plugin or extension.

The OffPAD may be used as an authenticator to servers that supports challenge-response authentication protocols. For existing systems that do not want to change their authentication system, the OffPAD may still be able to provide the server with the credentials. This, however, would require the use of HTTPS or another protection mechanism, as the username and password would be sent in plaintext from the OffPAD to the user's computer.

3.3 Validation of DNSSEC

DNSSEC is a mechanism for confirming the validity of a domain name system (DNS) record using asymmetric cryptography. DNSSEC rests on a hierarchic PKI where the public key of the DNSSEC root is globally known and acknowledged. The OffPAD can store a copy of the DNSSEC root public key, and get all the other required DNS-records from the computer. Then the OffPAD can validate each record and alert the user if validation fails.

The validation process of DNSSEC is quite simple and described in [4, 3]. Each resource record (RR) in DNS has a Resource Record Signature (RRSIG) containing the signature for the current record. This signature can be authenticated with the public key stored in the DNSKEY RR. A digest of the DNSKEY is stored in a Delegation Signer (DS) RR in the parent DNS zone[6], which in turn has a RRSIG. The validation process propagates all the way to the root.

For services that take the advantage of DNSSEC and stores their server certificate in the new TLSA resource record [10], would also make the OffPAD able to validate the server certificate through DNSSEC instead of or in addition to X.509. The TLSA RR is a resource record that gives you enough information to be able to validate the targeted server certificate, and it also tells you if you should validate it through X.509 or not. Public keys for other usage could also be stored in DNSSEC, in other RR types.

To be able to validate a server through DNSSEC, the OffPAD must be fed the chain of RRs, keys and their respective signatures. This must come from

[6] A DNS zone is a collection of RR that is administrated by the same entity, thus signed with the same DNSKEY

the untrusted computer when the authentication should take place. Man-in-the-middle attacks are still not a threat, since the whole chain of data is indirectly signed by the root node's key, which must be pre installed on the OffPAD.

3.4 Server authentication

There are multiple ways for an attacker to lure a victim to access a fake website with phishing attacks. Users are normally not well aware of the possible threats, and will in many cases not notice that the fake website is not the intended website, even if the user tries to inspect the server certificate [15]. To make it easy for the user to verify that she is connected to the correct service, the user can use a *petname system* [22, 7]. The petname system allows the user to associate server identities with personally recognisable identifiers such as a logo, name or tune, which are called petnames. Having a personally recognisable petname enables the user to easily validate the identity of the service. If the user navigates to a web site and there is a mismatch between its identifier and the one stored, the system should alert the user, or ask the user to add a new petname for the new service. Petname systems protects users from falling victim to phishing attacks [7].

As proposed by Ferdous *et al.* [8], the Petname system can be implemented on an OffPAD, validating the service being accessed. This will make the Petname system more user-friendly, since the user only needs to manage one central collection of petnames. DNSSEC combined with a petname system can give a quite strong service authentication, which gives a good base for the Server Authentication Assurance as proposed by Jøsang *et al.* [16].

3.5 Generation of One-Time Passwords

Several service providers that offer two factor authentication use one-time passwords (OTP) generated by a device. An OTP is considered to be a dynamic password, and is typically combined with a static password for authentication to an online service. The OTP is generated as a function of a secret string of bytes and either a timestamp or a counter value. Standards exist for how these functions can be implemented, e.g. Time-Based One-Time Password Algorithm (TOTP) [20] and HMAC-Based One-Time Password Algorithm (HOTP) [19].

Both TOTP and HOTP works by taking pseudorandom bytes from a HMAC[7] using a shared secret key and the time or counter value as input. The resulting string of pseudorandom bytes is the OTP. The key and the expected time or counter value are known to the service provider so that it can perform the same calculations and compare the received OTP with the locally computed OTP. The simplicity of the OTP mechanism makes it possible to install any practical number of different OTP services on the same OffPAD, to manage OTP-based authentication for access to different service providers.

[7] Hash-based Message Authentication Code

3.6 Encryption and Decryption of Messages

E-mail is still widely used for exchanging private and confidential information between people. While the user's connection to the mail server might be encrypted, the connection between different mail servers is not. This is problematic when it comes to "forgotten passwords" request, where some service providers send a temporary password in plain text, or a time limited link to reset the password. Sometimes the forgotten current password is even resent in clear. A solution is to let the service provider encrypt a message containing a password reset code with the public key for the OffPAD, so that it can only be decrypted by the OffPAD. This ensures that only the correct user can reset his password.

This can also be used to encrypt and decrypt other messages for the user, e.g. notifications from a bank. It is particularly useful were the user's computer is considered compromised.

3.7 Physical Access Control

Physical access control based on NFC technology is increasing in popularity. An OffPAD with passive NFC capability can support physical access control in the same way that standard NFC-enabled identity cards do, but can also support more advanced functionality, e.g. using personal stored fingerprints or other biometric information. A fingerprint can be scanned and validated on the OffPAD instead of on a central system, which enhances convenience, hygiene and privacy for the user. After matching the fingerprint the OffPAD can send an assertion to the access control system. If the user is authenticated on the OffPAD, she might not need to enter her PIN-code on the keypad by the door, limiting the possibility for an attacker to observe the PIN-code.

4 Limitations

The OffPAD is primarily intended to be a security device. Since complexity is the enemy of security it implies that 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. This design principle is aimed at reducing the attack surface. The challenge is to offer adequate usability despite these limitations.

4.1 Deployment and updating applications on the OffPAD

A challenge for the OffPAD is to upgrade the software on the device itself, as known bugs can make the OffPAD vulnerable, and even the process of updating the software might create vulnerabilities. There is a number of ways to update software, including the use of physical service stations, but this is quite impractical and it is hard for a user to build trust relations with such stations even if they are completely trustworthy.

The easiest, and still secure method for updates, is through the user's computer. Where the user (or OffPAD driver) downloads update files and transfer them to the OffPAD. If these files are signed, the OffPAD can validate the files and their source before running them. This is somewhat similar to how application distribution systems for smart phones work.

4.2 Controlled Connectivity

One of the requirements for the OffPAD is to enforce 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 short connection time requires either high transmission rate, or small amounts of data, or a combination of both. The standard for NFC [12] describes two communication modes: passive and active. Passive means that the target entity does not have a power source itself. It gets its power from a radio frequency (RF) field generated by the other part in the communication (the initiator). In active communication, both entities have a power source and the initiator and target alternate between making the RF field. The top speed is defined to be up to 424 kbit/s for passive mode and up to 6780 kbit/s for active mode. This limits the amount of data to transferred, but will probably not introduce any practical constraints for the security services mentioned here. Services that need to send or receive relatively large amounts of data might need to use active mode.

4.3 Driver Software and Browser Plug-ins

The OffPAD is intended to communicate with a variety of computer systems using one or several of the communication technologies listed in Table3. For each communication modality, specific software driver is needed, which *a priori* will be installed by the OffPAD hardware manufacturer. In case software update of any of the drivers is needed it is important that the driver can be securely obtained, and that it is easy to install.

Communication between the client computer and the OffPAD requires drivers and software installed on the client computer. It is important for the software to not introduce new security vulnerabilities to the host computer. A client computer must be used to transfer data to the OffPAD, and in case the client computer has been infected with malware the data to be transferred to the OffPAD could give attackers some information about the OffPAD, and potentially an opportunity to compromise the OffPAD itself. This can be prevented with pairing, where the user physically operates the client computer to initialise the pairing process. The pairing process can be done using Diffie-Hellman key exchange [6],

where both the client computer and the OffPAD select unique keys for each other. As the user starts the pairing process and observes that there is only the host and the OffPAD, this gives a small chance for a *man-in-the-middle-attack* (unless the *man* is already in the host system).

If there is malicious code on the host computer (in the drivers, browser plug-in or other places), applications that base themselves on information from the host may take wrong actions. Applications where the information is validated with a public key, with DNSSEC for instance, can be secure even if the host computer is compromised.

4.4 Protecting the OffPAD in Case of Theft

Even if there is strong access control on the device it is challenging to protect against all forms of attacks when the device is in the hands of the attacker, but some security measures can provide relatively robust protection against compromise. It should not be possible to rapidly do an exhaustive search through the entire PIN code space, or rapidly try many different fingerprints. A limit of e.g. 3 attempts can e.g. trigger a delay of e.g. 10 minutes before the OffPAD can be accessed again.

5 Commercial Adoption

The OffPAD might be relatively expensive to produce compared to a simple OTP-calculator, but as mentioned already some banks and institutions are already deploying relatively advanced devices to the public, such as card readers with a display and keypad. With an OffPAD as a general purpose security device many more security services can be supported and the resulting security for online transactions will be higher than that which can be achieved with most of the calculators used today.

The challenge is to get the average Internet users to discover the advantages of adopting this device for all their security solutions, and therefore to be willing to pay for it. Since the OffPAD is not a web-browser, an MP3-player or gaming console, people would probably not carry it with them everywhere, as they typically do for mobile phones. Adoption would certainly be increased in case different service providers and system developers (e.g. Facebook, Google, Microsoft and Apple) decided to promote an OffPAD, or a protocol supported by the OffPAD. But initially, it will most likely be companies with high security requirements that will see the need for having people use a device like the OffPAD.

5.1 Using a Mobile Phone as the OffPAD

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 [11]. 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.

It is important that the user can be assured that she is interacting with the operating system and not an application pretending to be the operating system. Windows Phone, iOS and Android[8] have a "trusted path" button that ensures that the user is directed to the home screen and not to an application.

The French company TazTag has introduced a mobile phone (TPH-ONE) [23] based on Android, and also integrates a secure element which can be accessed in a secure state. The secure element can potentially offer some of an OffPAD's functionality, by providing identity management and security services.

6 Conclusion

In this paper we describe the OffPAD as a general purpose security device for all types of users, which can replace or complement the different multi-factor authentication devices that already exist. This paper also describes specific security services that can be implemented on the OffPAD. It can be integrated with many of the existing systems and can offer a general security solution for the client and user side, both in enterprise and private settings.

7 Future work

Prototypes for the different security services mentioned in this paper are currently being developed. User experiments are planned to test if security services offered by the OffPAD are superior to solutions that already exist. Still to be developed are solutions for updating software and the operating system. The security enhanced smart phone produced by TazTag will be tested to see if the security of the operating systems and hardware can meet the requirements for an OffPAD. For instance, it must be possible to switch between the OffPAD functionality and the normal smart phone context using a physical switch on the device.

Acknowledgements. The work reported in this paper has been partially funded by Franco-Norwegian Foundation Project 1/11-FNS, and by the EUREKA Project 7161 Lucidman.

[8] It is possible to change the workings of the "Home" button in the settings for Android, so it might be possible for an application to change the function of this button.

References

1. Mohammed Alzomai, Bander Alfayyadh, and Audun Jøsang. Display security for online transactions. In *The 5th International Conference for Internet Technology and Secured Transactions (ICITST-2010)*, London, November 2010.
2. Mohammed AlZomai, Bander AlFayyadh, Audun Jøsang, and Adrian McCullag. An Exprimental Investigation of the Usability of Transaction Authorization in Online Bank Security Systems. In *The Proceedings of the Australasian Informatin Security Conference (AISC2008)*, Wollongong, Australia, January 2008.
3. R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the DNS Security Extensions. RFC 4035 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014.
4. R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource Records for the DNS Security Extensions. RFC 4034 (Proposed Standard), March 2005. Updated by RFCs 4470, 6014.
5. Nick Baker. ZigBee and Bluetooth strengths and weaknesses for industrial applications. *Computing Control Engineering Journal*, 16(2):20 –25, April-May 2005.
6. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976.
7. Md. Sadek Ferdous and Audun Jøsang. Entity Authentication & Trust Validation in PKI using Petname Systems. In Atilla Elçi et al., editors, *Theory and Practice of Cryptography Solutions for Secure Information Systems (CRYPSIS)*. IGI Global, May 2013.
8. Md Sadek Ferdous, Audun Jøsang, Kuldeep Singh, and Ravi Borgaonkar. Security Usability of Petname Systems. In *Proceedings of the 14th Nordic Workshop on Secure IT systems (NordSec 2009)*, Oslo, October 2009.
9. John Franks et al. *RFC 2617 – HTTP Authentication: Basic and Digest Access Authentication*. IETF, June 1999. URL: http://www.ietf.org/rfc/rfc2617.txt.
10. P Hoffman and J. Schlyter. *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*. IETF, August 2012. URL: http://www.ietf.org/rfc/rfc6698.txt.
11. Juniper Networks Inc. Juniper mobile threat report 2011. Technical report, Juniper Networks, Inc., 2011.
12. ISO. *IS18092. Near Field Communication – Interface and Protocol (NFCIP-1)*. ISO, Geneva, Switzerland, 2004.
13. ISO. *IS7816-4. Identification cards - Integrated circuit cards - Part 4: Organization, security and commands for interchange*. ISO, Geneva, Switzerland, 2005.
14. A. Jøsang and S. Pope. User-Centric Identity Management. In Andrew Clark., editor, *Proceedings of AusCERT 2005*, Brisbane, Australia, May 2005.
15. Audun Jøsang. Trust Extortion on the Internet. In *Proceedings of the 7th International Workshop on Security and Trust Management (STM 2011)*, Copenhagen, 2012. Springer LNCS 7170.
16. Audun Jøsang, Kent A. Varmedal, Christophe Rosenberger, and Rajendra Kumar. Service Provider Authentication Assurance. In *Proceedings of the 10th Annual Conference on Privacy, Security and Trust (PST 2012)*, Paris, July 2012.
17. Henning Klevjer, Audun Jøsang, and Kent A. Varmedal. 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, 8-9 April 2013.

18. B. Laurie and A. Singer. Choose the Red Pill and the Blue Pill: A Position Paper. In *Proceedings of the 2008 New Wecurity Paradigms Workshop*, pages 127–133. ACM, 2009.
19. D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen. HOTP: An HMAC-Based One-Time Password Algorithm. RFC 4226 (Informational), December 2005.
20. D. M'Raihi, S. Machani, M. Pei, and J. Rydell. TOTP: Time-Based One-Time Password Algorithm. RFC 6238 (Informational), May 2011.
21. Frank Stajano. Pico: No more passwords! In *Proceedings of the 19th International Workshop Security Protocols*, pages 49–81, Cambridge, March 2011.
22. Marc Stiegler. Petname Systems. Technical Report HPL-2005-148, HP Laboratories Palo Alto, 15 August 2005.
23. TazTag. Mobility Products, 2012.
URL: http://taztag.com/index.php?option=com_content&view=article&id=104.