OffPAD – Objet Personnel d'Authentification Hors-ligne appliqué aux hôpitaux et banques en ligne

Denis Migdal
Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen, France

Audun Jøsang
Ifi, University of Oslo, Norvège

September 19, 2017

Abstract

Les solutions d'identité et d'authentification manquent souvent d'ergonomie, de flexibilité, ou ne fournissent pas une fiabilité suffisante. Le concept Lucidman (Local User-Centric Identity Management) est une approche fournissant des fonctionnalités d'authentification et d'identité flexibles, sûres et ergonomiques. Dans ce cadre, nous présentons l'usage d'un OffPAD (Offline Personal Authentication Device) en tant qu'objet sécurisé supportant différentes formes d'authentification. L'approche Lucidman/OffPAD consiste à déporter la gestion d'identité et d'authentification, du serveur (ou du cloud) vers l'utilisateur. Cet article vise à montrer la manière dont l'OffPAD minimise les pré-requis, améliore l'ergonomie, et renforce la fiabilité des authentifications, tout en ayant l'avantage de permettre, même en présence de clients corrompus, des transactions en ligne sécurisées. L'objet sécurisé OffPAD a été conçu tel une coque de téléphone, ne constituant ainsi pas un objet supplémentaire porté par l'utilisateur. Nous nous concentrons sur 6 démonstrations, 3 dans un contexte de banques en ligne, et 3 dans un contexte hospitalier où les infirmiers, docteurs, et patients sont authentifiés via l'OffPAD afin de leur fournir, dans différentes situations, l'accès à des ressources.

1 Motivations et état de l'art

Nous présentons le concept OffPAD, une coque de téléphone avec un élément sécurisé, et ses composants logiciels. Le concept OffPAD a été proposé dans [5], cependant les prototypes matériel et logiciel ont été développés durant les deux dernières années dans le cadre du projet OffPAD[1].

[1]Financé par EUREKA et Eurostars, nr. E!8324.

Un des objectifs d'OffPAD est d'augmenter la sécurité et fiabilité sans réduire l'ergonomie, i.e., en ayant un minimum d'interférences avec les tâches usuelles de l'utilisateur, et en automatisant certains processus sous-jacents à l'authentification. OffPAD peut être vu comme un objet de gestion d'identité, considérant qu'une entité peut avoir simultanément plusieurs identités. OffPAD vise à améliorer le traditionnel "modèle silo", qui stocke les identités côté serveur, par la gestion locale des identités sous le contrôle exclusif de l'utilisateur. OffPAD conserve les données d'identifications des utilisateurs, mais aussi ceux des fournisseurs de services afin de permettre l'authentification du service par l'utilisateur. OffPAD améliore aussi la fédération d'identité (e.g. Shibboleth, OpenId, FacebookConnect, FIDO) qui sont gérées sur des serveurs ou dans le cloud, et sont "centré réseau" au lieu d'être "centré utilisateur".

Nous faisons la distinction entre les entités systèmes (navigateur ou serveur) et les entités légales/cognitives (personne ou organisation), multipliant ainsi les possibilités d'authentification mutuelles. Nous considérons aussi trois types d'authentification : (i) *syntaxique*, le plus simple, qui, e.g. vérifie la validité du certificat reçu, indifférent à l'identité du propriétaire du certificat ; (ii) *sémantique* incluant des vérifications syntaxiques, mais aussi, via un tiers de confiance, que l'entité distante respecte une politique de sécurité donnée ; et (iii) *cognitive*, le plus riche, nécessitant que l'entité de confiance soit capable de raisonnements cognitifs, comme les humains ou systèmes d'IA avancées afin de reconnaître l'identité du serveur. Avec OffPAD, nous nous intéressons à l'authentification cognitive impliquant un utilisateur humain. La norme X.800 (authentification syntaxique) met en jeu, au niveau de la couche réseau, une authentification de l'ordinateur client par le serveur (CS), et inversement (SC). Cette authentification est typiquement transparente pour un utilisateur humain. Cependant, pour des services en ligne, l'authentification de l'utilisateur par le serveur (US) et l'authentification cognitive du serveur par l'utilisateur (SU) sont plus pertinentes. L'importance de ces classes d'authentification vient du besoin d'une sécurité de bout-en-bout, i.e., entre l'utilisateur humain (U) et le serveur (S).

Il est communément admis que, lors d'une connexion TLS, l'authentification traditionnelle du serveur par les certificats PKIX[2] du navigateur fournit une authentification SU, ce qui n'est pas réellement le cas [3].

Par exemple, les attaques par hameçonnage débutent généralement par des pourriels qui trompent l'utilisateur de sorte à ce qu'il saisisse ses identifiants sur un site contrefait. D'un point de vue syntaxique, le certificat du serveur étant validé par le navigateur, il est correctement authentifié par TLS. Cependant, d'un point de vue cognitif, il n'y a pas authentification, l'identité du site étant différente de celle attendue par l'utilisateur. Le problème est dû à la pauvre ergonomie offerte par les implémentations actuelles de TLS [4] qui ne facilitent pas la reconnaissance d'identité. L'infection de la plate-forme cliente par des maliciels permet la modification de données que l'authentification d'entité seule ne peut détecter. Par exemple, lors de transactions bancaires en ligne,

[2]Public-Key Infrastructure based on X.509 certificates

un cheval de Troie peut changer le comportement du navigateur et ainsi modifier arbitrairement les entrées et sorties de données. Les données de la transaction sont ainsi modifiées, sans que l'utilisateur en soit conscient, et malgré l'authentification mutuelle des entités. Les maliciels SpyEye, Zeus, IceIX, TDL, Hiloti, et Carberp, sont des exemples concrets permettant de telles attaques. Les solutions actuelles d'authentification de l'origine des données, lors de transactions en ligne, sont inexistantes ou inadaptées, considérant le client à l'origine des données saisies par l'utilisateur, ici modifiées par le client avant leur envoi au serveur, brisant ainsi l'authentification de l'origine des données. Lors de transactions en ligne, la différence entre l'authentification des entités et l'authentification des données nécessite des mécanismes de sécurité dédiés à la vérification de l'intégrité des données. OffPAD permet une authentification, fiable et ergonomique, de l'origine des données, comme expliqué ci-dessus.

Les travaux existants sont présentés dans le journal [5] et le rapport technique [7] accompagnant cet article. Plusieurs solutions d'authentification reposant sur des objets externes sont présentées dans la littérature, incluant Pico de Stajano [11], MP-Auth de Mannan et van Oorschot [9], et Nebuchadnezzar de Singer et Laurie [10]. Cependant, ces objets supportent uniquement l'authentification des entités clientes (usuellement l'utilisateur) auprès des entité serveurs, contrairement à OffPAD qui supporte en sus l'authentification des données, et des entités serveurs auprès des entités clientes.

2 Description de l'objet OffPAD

OffPAD est un objet sécurisé, i.e. adéquatement protégé contre les attaques connues. Le premier prototype d'OffPAD est une coque de téléphone connectée à son hôte via une interface micro-USB standard. OffPAD est ainsi un objet portable sans constituer un objet électronique supplémentaire dans la poche de l'utilisateur. Débloquer l'OffPAD est actuellement effectué via empreinte. OffPAD est considéré hors ligne, signifiant que ses communications suivent un format contrôlé, durant des périodes de temps courtes et restreintes, et évitant le sans-fil, i.e., nous utilisons uniquement des interfaces micro-USB ou NFC.

Être hors ligne élimine l'exposition aux menaces venant d'Internet. Ainsi, nous considérons que les attaquants sont incapables d'exploiter les bugs du système d'exploitation et des applications d'OffPAD.

La première connexion à l'OffPAD requiert une confiance à la première utilisation (Trust-On-First-Use ou leap-of-faith). Au premier usage, il n'y a pas de moyen cryptographique de vérifier la connexion entre l'objet et la plate-forme cliente, la confiance doit être simplement basée sur l'observation de l'installation physique. Une vue schématique de l'OffPAD est illustrée Fig.1.

OffPAD intègre les composants matériels suivants : (i) *élément sécurisé Javacard/Global plate-forme* pour stockage et exécution sécurisés. (ii) *écran e-Ink* de 2.5 pouces, (iii) *DEL multicolore* pour la transmission d'informations simples, (iv) *émetteur-receveur NFC* et (v) *micro-USB* pour communiquer avec le client, (vi) *capteur d'empreintes*, (vii) *mémoire flash* de 4Go à 16Go. Nous

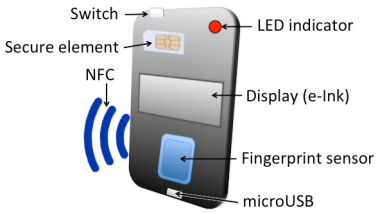

Figure 1: Schéma des composants OffPAD v.1

considérons sécurisés les capteurs intégrés à OffPAD qui utilise aussi le téléphone hôte pour d'autres acquisitions, e.g. la caméra, ou pour lui déléguer les calculs les plus lourds, e.g. ROC. Ainsi, un maliciel sur le téléphone peut communiquer de fausses informations à l'OffPAD. Cependant, ces entrées sont considérées non-sûres dans nos scénarii. Le microcode d'OffPAD supporte les fonctionnalités suivantes :
Authentification du porteur par sa biométrie.
Gestion de certificats sur l'OffPAD pour vérifier les signatures, e.g. vérifier l'identité du fournisseur de services.
Signature et vérification via la clé privée de l'OffPAD débloquée après authentification du porteur.
Afficher les informations sensibles via l'écran e-Ink ou la DEL multicolore.
Enrôlement biométrique du porteur via l'OffPAD d'après les modalités biométrique spécifiées.

3 Démonstrations de l'OffPAD

Les applications suivantes d'OffPAD sont présentées.
Data-US : Authentification des données utilisateur par le fournisseur de services, basé sur ROC (Reconnaissance Optique de Caractères), aussi affichées sur l'écran e-Ink de l'OffPAD.
SU : Authentification du serveur par l'utilisateur, basé sur le système de petnames [2] gérés par OffPAD.
US: Authentification de l'utilisateur par le fournisseur de services, basé sur l'extension d'un protocole challenge-réponse, XDAA [6].
Auto-connexion : (dé)connexion contextuelle et automatique basée sur la position en intérieur de l'OffPAD, détectée par le système de Sonitor.
Multi-connexion : accès automatique à une ressource conditionné par l'authentification simultanée de plusieurs utilisateurs, utilisant aussi le système TellU Smarttracker.
Auth. forte : L'authentification forte est requise pour accéder à des informations ou services sensibles, l'utilisateur s'authentifie via l'OffPAD.

Nous présentons la manière dont OffPAD assure l'authentification mutuelle des entités utilisateur et serveur, ainsi que l'authentification des données. Chaque cas d'utilisation est illustré par une cérémonie [1], protocole où l'environnement

et les actions utilisateurs sont inclus. Le but de nos solutions est de permettre, même en présence de plate-formes clientes infectées par un maliciel, des interactions sécurisées. Nous illustrons ici les cérémonies SU, Auto-connexion, Multi-connexion, et Auth. forte et motivons Auto-connexion pour les hôpitaux. Afin de supporter l'authentification cognitive du serveur, le nom de domaine inclus dans son certificat est associé à un surnom défini par l'utilisateur (petname) et représentant le fournisseur de services, facilitant ainsi la reconnaissance du fournisseur de services par l'utilisateur. Le certificat du serveur est aussi validé de manière traditionnelle, fournissant une authentification syntaxique du serveur.

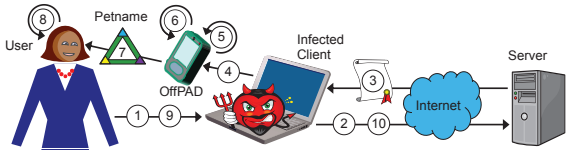

Figure 2: Authentification du serveur par l'utilisateur via les petname gérés par OffPAD

Les actions/messages de la démonstration SU sont décrits ci-dessous : (1) L'utilisateur initie une connexion TLS sécurisée à travers la plate-forme cliente (2) La plate-forme cliente contacte le serveur (3) Le serveur envoie sa clé publique via son certificat (4) Le certificat du serveur est transmis à l'OffPAD (5) Le certificat du serveur est validé (authentification syntaxique du serveur) (6) Le certificat du serveur est associé à un petname (7) Le petname est présenté à l'utilisateur (8) L'utilisateur authentifie cognitivement le serveur (9) L'utilisateur approuve l'authentification du serveur (10) La connexion TLS est établie entre le client et le serveur.

Les hôpitaux sont un environnement de travail mouvementé où de multiples utilisateurs ayant diverses fonctions interagissent avec le système informatique de l'hôpital pour diverses tâches comme l'enregistrement des patients, les informations routinières, ou la journalisation des tâches médicales. Cependant, la sécurité et la confidentialité des informations des patients doivent être garanties. Cela implique que l'équipe de soin doit se connecter et être autorisée à chaque fois qu'ils interagissent avec le système informatique. Cela est chronophage et distrait des tâches principales. L'inadéquation de l'authentification classique via nom d'utilisateur/mot de passe est dû aux observations suivantes : (i) le travail médical survient rapidement alors que la connexion détourne l'attention ; (ii) le travail médical est nomade avec des interruptions constantes alors que la connexion est restreinte à un ordinateur ; (iii) le travail médical est collaboratif, partageant du matériel alors que la connexion est prévue pour les activités d'un seul utilisateur.

Les démonstrations de l'OffPAD se concentrent sur des mécanismes d'authentification continues, contextuelles et ergonomiques afin de soulager l'utilisateur du poids d'un processus de (dé)connexions fréquentes. Nous présentons un mécanisme d'authentification basé sur la position où l'utilisateur sera automatiquement

connecté à un terminal lorsqu'il s'en approchera, et déconnecté quand il s'en éloignera.

Les actions/messages des démonstrations Auto-connexion, Multi-connexion, et Auth. forte sont décrits ci-dessous : (1) L'infirmière (i) associe, auprès de Smarttracker (s), l'OffPAD à un tag qu'elle portera, et dont la localisation est détectée par Sonitor (l) (2) (i) entre dans une zone, (l) notifie (s) qui en informe (i) (3) Optionnellement, (i) s'authentifie auprès de (s) via l'OffPAD (Auth. forte) (4) (s) connecte (i) auprès du système informatique de l'hôpital (Auto-connexion). (5) (i) accède à la ressource, variant selon les utilisateurs déjà connectés (Multi-connexion). (6) (i) sort de la zone, (l) notifie (s) qui déconnecte et notifie (i).

4 Discussion et conclusion

Plusieurs applications utilisant OffPAD peuvent être imaginées [8]. Nous en mentionnons ici quelques autres. La méthode pour les transactions bancaires peut être utilisée pour signer des prescriptions médicales. La méthode pour obtenir les données du patient peut être utilisée dans d'autres situations, e.g., quand une infirmière est autorisée à faire des changements sur une ressource, seulement sous la supervision d'un docteur (e.g. un spécialiste). L'Auto-connexion peut être utilisée pour un déplacement facilité des patients entre les chambres, où les divertissements, e.g. les préférences télévisuelles, sont immédiatement transférées au terminal le plus proche du patient. Les petnames peuvent être associés aux noms de domaine de n'importe quel type de services sensibles, comme les impôts, les magasins préférés, etc, l'utilisateur peut ainsi les authentifier cognitivement.

Remerciements

Nous tenons à remercier tous les membres du projet OffPAD qui ont soit participé à l'élaboration des démonstrations d'OffPAD, soit contribué par leurs discussions ou bonne idées ; particulièrement : L. Dallot, L. Miralabe, et G. Cornet (TazTag), K.E. Husa and S. Morka (TellU, fournisseur de plate-formes et services IoT), M.P. Haugen (U.Oslo), C. Rosenberger et E. Cherrier (Laboratoire GREYC), A. Taherkordi (Sonitor, concepteur de solutions de localisation en intérieur).

References

[1] C. Ellison. Ceremony Design and Analysis. Cryptology ePrint Archive, Report 2007/399, 2007.

[2] M. S. Ferdous and A. Jøsang. Entity Authentication & Trust Validation in PKI using Petname Systems. In *Theory and Practice of Cryptography Solutions for Secure Information Systems (CRYPSIS)*, pages 302–334. IGI Global, 2013.

[3] A. Jøsang. Trust Extortion on the Internet. In *7th Workshop on Security and Trust Management (STM)*, pages 6–21. LNCS 7170, Springer, 2012.

[4] A. Jøsang, B. AlFayyadh, T. Grandison, M. AlZomai, and J. McNamara. Security Usability Principles for Vulnerability Analysis and Risk Assessment. In *23rd Annual Computer Security Applications Conference (ACSAC)*, pages 269–278. IEEE, 2007.

[5] A. Jøsang, C. Rosenberger, L. Miralabé, H. Klevjer, K. A. Varmedal, J. Daveau, K. E. Husa, and P. Taugbøl. Local user-centric identity management. *Journal of Trust Management*, 2(1):1–28, 2015.

[6] H. Klevjer, K. A. Varmedal, and A. Jøsang. Extended HTTP digest access authentication. In *3rd IFIP WG 11.6 Working Conference on Policies & Research in Identity Management (IFIP IDMAN)*, volume 396 of *IFIP AICT*, pages 83–96. Springer, 2013.

[7] D. Migdal, C. Johansen, and A. Jøsang. Offpad: Offline personal authenticating device – implementations and applications. Technical Report 454, U. Oslo, Aug. 2016. (`http://heim.ifi.uio.no/~cristi/papers/TR454.pdf`).

[8] K. A. Varmedal, H. Klevjer, J. Hovlandsvåg, A. Jøsang, J. Vincent, and L. Miralabé. OffPAD: Requirements and Usage. In *Network and System Security (NSS)*, volume 7873 of *LNCS*, pages 80–93. Springer, 2013.

[9] M. Mannan and P. C. van Oorschot. Using a personal device to strengthen password authentication from an untrusted computer. In *Financial Cryptography and Data Security*, pages 88–103. Springer, 2007.

[10] B. Laurie and A. Singer. Choose the red pill and the blue pill: a position paper. In *Proceedings of the 2008 workshop on New security paradigms*, pages 127–133. ACM, 2009.

[11] F. Stajano. Pico: No more passwords!. In *International Workshop on Security Protocols*, pages 49-81. Springer, 2011.