Indirect NFC-Login on a Non-NFC Device using an NFC-Smartphone

Bernd Borchert and Max Günther
Department of Computer Science, University of Tübingen, Germany

Abstract

Smartcard based authentication on web services stays a niche application because of the lack of smartcard readers on the vast majority of internet devices. In this paper we discuss a method that uses an NFC-enabled Smartphone in order to login via NFC-enabled smartcard on basically any internet device. We explain the details of this method and analyze its security, deployability, and usability aspects.

1. Introduction

User and password authentication is still the standard authentication method in the internet. For well-known security reasons a second factor ”what-you-have” like a smartcard is desirable. While it is well known how to design an authentication protocol for internet logins using smartcards the limiting factors for a large scale adoption of such a solution are missing smartcard readers and moreover the web browser which is not able to communicate with a reader even if present.

Grosse and Upadhyay suggest to embed a smartcard chip in a token with an USB interface or dual-interface (USB and NFC) and to enable the browser to communicate with that chip [1]. This way, a user can log into his account on his NFC-reader equipped internet client by holding the NFC-token close to it.

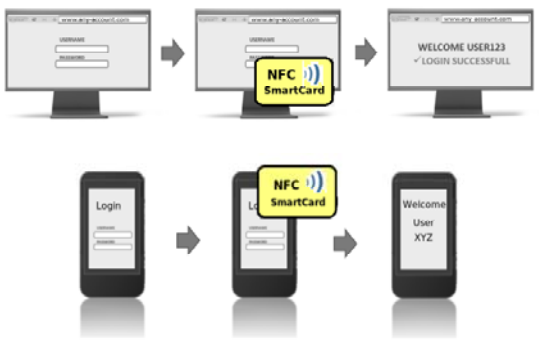


Fig. 1 Two examples of the direct NFC-Login method.

In this paper we will refer to this method as the *direct NFC-Login method* (we will focus on NFC connectivity).

The problem with this method is that clients need USB or NFC connectivity and - more restricting - the browser extension. Popular examples of unsupported clients would be non-NFC Smartphones and tablet computers including Apples iPhone/iPad and presumably the majority of PCs inside a company.

In this paper we try to overcome this limitation, still using NFC smartcards (or NFC tokens not in the shape of a smartcard) as a second factor. We suggest to use an NFC-enabled Smartphone to access a smartcard in order to authorize the login on an arbitrary internet client.

Our method works as follows: The server creates a challenge and encodes it as a 2D-code. The Smartphone scans the code and handles the challenge over to the smartcard where the response is computed and transmitted via the Smartphone and its mobile internet connection to the server. The server opens the session on the PC-browser in case the response is correct. We refer to this method as the *Indirect NFC-Login method* as opposed to the direct method described above. The Indirect NFC-Login method is shown from a users perspective in Fig. 2.

Fig. 2 Indirect NFC-Login

The Smartphone here acts as a mere communication device. It communicates with the web browser (via camera), with the smartcard via NFC (back and forth), with the server via mobile internet, and with the user via display, see Fig. 3.

Fig. 3 Indirect NFC-Login Information Flow

The login via Indirect NFC-Login is possible on every internet client given the user has an NFC-Smartphone.

The rest of this paper is structured as follows: In Sect. 2 we discuss related work like Google's 2-step-verification, smartcard-like tokens, and Smartphone based authentication solutions. We explain our suggested method in terms of architecture, information flow, and implementation in Sect. 3.

We evaluate its usability, deployability, and security in Sections 4 to 6 using the framework for comparative evaluation of Web authentication schemes by Bonneau et al. [13].

2. Related Work

Google 2sv (two-step-verification) [2]: In the Google 2-step verification method, or other similar methods, the user has to enter a verification code in addition to his username and password. The verification code may be sent to the user via voice message or via text/SMS or it may be created on the Smartphone. In the latter case, an individual key is stored on the Smartphone and an app computes the verification code from a timestamp or counter and this key (implementing IETF RFCs for one-time password algorithms [3], [4]). Two problems occuring here are phishing, server side authentication data leaks, and mobile malware that could tap voice/text messages or keys.

Smartcard-Like Token: Grosse and Upadhyay attempt to overcome some of the abovementioned problems with the 2sv approach by using smartcard-like tokens [1]. These tokens store cryptographic keys, execute asymmetric cryptography, and are equipped with a USB interface, an NFC-interface, or both. Grosse and Upadhyay suggest that the browser on the internet client provides two new APIs to web applications and facilitates the communication between a web application and the token for the purpose of key registration and authentication. They emphasize that no additional software other than a compliant web browser should be required for such a method.

We agree with the latter statement, nevertheless we think there will be still many clients without a compliant web browser. Examples are PCs in a company or an internet cafe, tablet computers, or - generally speaking - all clients that the user cannot update or modify. In addition, the requirement for a physical USB or NFC interface may not be satisfied in a number of cases.

Authentication with Smartphones and 2D codes: The Indirect NFC-Login method we propose here can be also seen as an extension of a class of login methods which uses the Smartphone in a similar way (to scan a 2D-code and to communicate with the server) but additionally as a credential store, i.e. the response is computed using a secret stored on the Smartphone. Various such Smartphone-as-key methods have been suggested [5]–[8] and product implementations already exist [9]–[11].

In their 1-factor variants, those methods aim at increased usability (neither username nor password have to be typed) and security (there is no password that could be phished).

Fig. 4 Smartphone-as-Key

One problem with the Smartphone-as-key methods is the growing threat from mobile malware which could steal the credential when stored on the Smartphone. This is one reason why the extension to the Indirect NFC-Login with a smartcard as secure credential store may be desirable. Another reason is that existing Smartcards like student cards, company cards or customer cards could be reused for this new application of web service authentication.

The Google Sesame method [9] is a different approach. It moves the place to enter the password from the PC to the user's Smartphone. It does not store a key-like credential on the Smartphone. The Indirect NFC-Login could be seen as a variation of Google Sesame in which via Smartphone not the password is supplied as a "what-you-know" factor but the smartcard as a "what-you-have" factor.

PhoneAuth [12]: Czeskis et al. present a Smartphone-as-key solution where the user's Smartphone is used in addition to traditional username/password authentication in order to provide additional security through a session-to-channel binding mechanism. PhoneAuth therefore can be seen as a kind of an "ignition key" for internet sessions, a direction in that Grosse and Upadhyay also want to evolve the usage of their smartcard-like solution.

3. Indirect-NFC-Login

In this chapter we describe the architecture and technical details of the Indirect NFC-Login method. We start with a brief overview of the generic procedure and explain our implementation afterwards. The method is shown from a users point of view in Fig. 2 and from an architectural perspective in Fig. 3.

3.1. Procedure

1) The user opens the login page that shows a dynamically created 2D code containing the server challenge.
2) The user scans this code with his Smartphone.
3) The Smartphone shows the URL and asks the user to bring the corresponding card close.
4) Once the NFC connection is established, the Smartphone sends the challenge to the Smartcard. The Smartcard computes the response and returns it.
5) The Smartphone then sends username, challenge and response to the account server via mobile internet.
6) The account server then opens the account on the PC browser in case the response is correct.

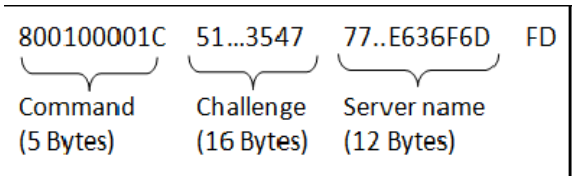
3.2. Implementation

We use public key cryptography to authenticate a user to an account server. The user has to sign a challenge with his private key. The account server knows the users public key so that it can verify a signature we assume that the key registration is already performed. The challenge basically consists of a randomly created session id.

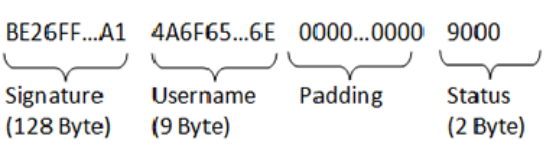
The components involved are server, PC, Smartphone, Smartcard. While the PC is only required to run a normal web browser, all other components are required to run custom software. In Fig. 3 these components and their communication are shown and are explained more comprehensively in the following.

1) The 2D code encodes an ASCII text show on the right. The first four lines in this example are administration information. The fifth line contains the challenge (created by the server, randomly and dynamically) and the last line contains the server name.
   Invisible to the user and not encoded in the 2D-code, the server also provides a browser-identificaiton-code'.

2) The Smartphone app scans the 2D code and extracts the ASCII text.
3) The Smartphone shows the URL to the user and ask him to bring his Smartphone close.
4) Once the NFC connection between Smartphone and Smartcard is established, the Smartphone app sends server name and challenge to the Smartcard.



If the smartcard has a user and a private key for the given server, it computes the signature for the challenge with the key and returns username and signature.



5) The Smartphone app then sends a message to the server including username, challenge and signature by calling a URL with these parameters. Needed URL information may be stored on the Smartphone, or on the Smartcard, or may be transmitted with the 2D code.

```
https://www.test.com/immediate/ReceiveFromMobi
le.php?challenge=QFOGmDbzHzbEYx5G&username=Joe
.Brown&signature=EE26FF509871AEFB0D0411D21B496
8AB0FA379BE94F0FB7FC9E7D8A29AB4253221671D6366E
0DFCFA6C2B0E4A2F3785349B739672BE25CF9D7B5D8D7C
F78DE49D8FFA63A8AC7BFC4429C0B940EFB7EB7B5E1C6B
15F15F71ACD1DE4C3AE01565316F36508025178F8A7D20
F94A53FFCAB7335ABE0306D2D0C548A1A4E1C6D06A1
```

6) After receiving this information, the server activates the session if the user exists, the challenge has not expired, and the signature is valid. To do so, the server redirects the PC browser to the authenticated users account page by responding accordingly to a polling request initiated by the browser and including the 'browser-identification-code'.

3.3. Key Registration

When using asymmetric cryptography with smartcards, the smartcard typically generates a public/private key pair and provides the public key whereas the private key never leaves the smartcard. The question is how the server gets to know the users public key. There are three cases to distinguish:

Direct Registration. Grosse and Upadhyay require a browser API for the token registration. A smartcard or token registered with this mechanism can be used for the Indirect NFC-Login afterwards because the indirect method is designed to be an extension of the direct method.

Indirect Registration. In cases when the user cannot access his account even once from a capable internet client (e.g. a company account) an indirect process is required. After the user is logged in on a PC he navigates to a subpage where a 2D-code is shown which contains portal name, username and a registration session id. The user scans this 2D-code with his Smartphone and is asked to hold the smartcard close. The Smartphone sends server name and user name to the smartcard. The smartcard creates a public/private key pair. It stores (server name, user name, private key) and returns the public key to the Smartphone. The Smartphone sends the public key together with the registration session id via tls-internet to the server.

Issuer-Side Registration. If special smartcards like customer cards are used the key registration can be performed in a safe environment. This avoids attacks by client-side malware on the registration process which are possible in the first two cases (e.g. malware could create a key pair on its own and register its public key with the server).

3.4. Variants

Symmetric vs. Asymmetric Keys: The described method works with symmetric as well as asymmetric keys. Nevertheless the main advantage of asymmetric keys is that the server stores only the public keys and therefore data leaks are not dangerous [1]. Asymmetric algorithms tend to be slower than symmetric ones, but our implementation did not show a relevant difference - smartcards are quite fast nowadays.

Offline Fallback with Asymmetric Keys still Possible: In the case of asymmetric keys there are two possibilities regarding the encryption: In the first, which we chose above, the server generates a nonce and expects to receive the private-key encryption as a response which the server is able to check with the public key. In this variant the response is too long to be typed manually what makes the internet access of the mobile phone mandatory. An alternative implementation lets the server encrypt a short passcode with the public key. The private key is able to decrypt the message and show the passcode to the user on the mobile phone. This allows for an offline fallback mechanism. As a drawback of this solution, the 2D-code becomes quite large which makes the scanning slower or impossible.

No-Javascript Fallback: Javascript is needed for the polling connection between browser and server. In case Javascript is disabled on the client computer the user has to click a button to redirect the browser to the open account in step 6.

Chipcards instead of Smartcards: The method could be built with chipcards like Mifare instead of smartcards. Nevertheless, because chipcards cannot execute challenge/response the credential has to leave the card and could be object to theft. Therefore, for security reasons it is not recommended to use chipcards.

4. Usability

The Indirect NFC-Login method is obviously less convenient than the direct one and requires an NFC-enabled Smartphone. It is only a fall back in case the direct login is not possible. For a more detailed evaluation of the usability, we use the framework by Bonneau et al. for comparative evaluation of Web authentication schemes [13].

- Memorywise-Effortless: No
  The method does not add effort but inherits that from the password.
- Scalable-for-Users: No/Somewhat
  A user would need a different password for each account, but could use the same smartcard for all of them.
- Nothing-to-carry: No
  The user has to carry his Smartphone and a smartcard.
- Quasi-Nothing-to-carry: Yes
  It can be argued, that the user carries his Smartphone and smartcard/wallet anyway.
- Physically-Effortless: No
  The user has to scan the 2D-code and hold the smartcard close to the Smartphone.
- Easy-to-Learn: Yes
- Easy-to-Use: Yes
- Infrequent-Errors: Somewhat
  Possible problems are the 2D-scan process, accessing the NFC-smartcard, and mobile internet connection reliability.
- Easy-Recovery-from-Loss: No
  In case the user loses his smartcard, he has to revoke it and to order a new one.

Note that the direct NFC-Login has slightly better evaluations for the criteria Quasi-Nothing-to-carry, Physically-Effortless, Easy-to-Learn/Use, and Infrequent-Errors and has the same evaluation for the other criteria.

5. Deployability

Bonnau et al. use six criteria to evaluate deployability. We believe that Browser-Compatibility is the most important criterion because as long as it is not fulfilled a method will not be adopted at scale.

- Accessible: Somewhat
  Blind or one-handed people may have some difficulties scanning the 2D-code and handling the Smartphone and smartcard.
- Negligible-Cost-Per-User: No
  The smartcard adds cost and the NFC-Smartphone is a costly requirement.
- Server-Compatible: No
  Server side changes are required.
- Browser-Compatible: Yes
  Works on every internet client.
- Mature: No
  Only a prototype/demo exists.
- Non-Proprietary: No
  No open source project yet.

The direct NFC-Login method is more accessible and cheaper than the Indirect NFC-Login. Both methods do not fulfill the criteria Server-Compatible, Mature, and Non-Proprietary.

Browser-Compatible means, that any standard-browser can be used and no additional software on the internet client is required. The direct NFC-Login method relies on a special API provide to the web service by the browser and is therefore clearly not Browser-Compatible (at least until the supposed API is a standard for web browsers). A criterion Client-Compatible could also include the required hardware. In this sense the direct NFC-Login method is not Client-Compatible because USB and NFC are not ubiquitous among internet clients. The Indirect NFC-Login method, on the other hand, is Browser-Compatible and Client-Compatible as the browser only has to display the 2D-code.

The Indirect NFC-Login method requires that the user has an NFC-enabled Smartphone. The current status is that only a minority of Smartphones are equipped with NFC. Nevertheless, nearly all newly available Smartphone models are equipped with NFC nowadays, with the notable exception being the iPhone (market share of roughly 15%).

The Bonneau framework does not address requirements of the kind of ”the user has to have an NFC-Smartphone”. But it clearly is a deployability disadvantage.

6. Security

6.1. Identity Theft

The security of the Indirect NFC-Login method is mainly based on the challenge/response scheme executed between server and Smartcard and facilitated by the browser and the Smartphone. In the following, we evaluate the security of the Indirect NFC-Login method, again following the evaluation framework suggested by Bonneau et al.

- Resilient-to-Physical-Observation: Yes
  The 2D-code contains administration data and a challenge but no secret that could be observed.
- Resilient-to-Targeted-Impersonation: Yes
  The users private key is not related to his personal details.
- Resilient-to-Throttled-Guessing: Yes
- Resilient-to-Unthrottled-Guessing: Yes
  The credential space is large enough and credentials are chosen randomly.
- Resilient-to-Internal-Observation: Yes
  The smartcard will never expose the private key and is malware-proof and tamper-proof.
- Resilient-to-Leaks-from-Other-Verifiers: Yes
  A unique key-pair is used for every account.
- Resilient-to-Phishing: Yes
  The private key cannot be phished.
- Resilient-to-Theft: Yes/Somewhat
  Yes, if used in addition to a password.
- No-Trusted-Third-Party: No/Somewhat
- Requiring-Explicit-Consent: Yes
  The user has to bring the smartcard close to the Smartphone.
- Unlinkable: Yes
  Different public keys are used for each account.

Not surprisingly, the direct and the Indirect NFC-Login methods have the same security properties.

6.2. Session Theft

In the direct NFC-Login method, the internet client is the obvious place to steal the session. In the Indirect NFC-Login method, the Smartphone is another place where the session could be stolen. Note that Smartphone malware observes a challenge/response pair during the authentication. This threat was the reason to introduce the browser identification token described above - it prevents malware on the Smartphone (or a possible control and command server) to gain a session with the tapped challenge/response pair and session id. Another way malware on the Smartphone could steal a session is by opening a session via mobile internet on its own and abusing the smartcard when the user

holds the smartcard close to the Smartphone. The malware will still lack the password.

As the session cannot be stolen on the Smartphone, the Indirect NFC-Login method has the same security level regarding session theft as the direct variant.

7. Variants and Extensions

7.1. Form Factors of the NFC-Token

Various variants of the format and the position of the NFCSmartcard are possible.

- **Smartcard**. This may be the most obvious technology for companies or other organizations, as company cards or student cards already have this form factor.
- **Key fob**. Grosse and Upadhyay suggest a key fob containing the NFC-chip [1].
- **Ring**. As another example the NFC-Chip may be integrated into a ring owned by the user. The Smartphone and the Smartcard automatically have an NFC-connection established once the user takes the Smartphone in his hand [14].
- **Implant**. The same usability gain is obtained in case the NFC-Chip is implanted into the user's hand. Moreover, in that scenario the risk of loss or theft of the NFC-Chip would be zero.

In case the first factor password is abandoned the gain of usability for the variants ring and implant comes along with the following severe security disadvantage: the continuous resp. frequent proximity of Smartphone and NFC-Chip may give Smartphone Trojans the chance to open a session secretly not only when the user establishes proximity but virtually anytime resp. very often. In other words, the security benefit Requiring-Explicit-Consent is no longer given.

7.2. "What-you-have" and/or "What-you-know"

The direct NFC-Login method and the Indirect NFC-Login method add another factor "what-you-have" (smartcard) to the factor "what-you-know" (password).

Instead of adding the factor smartcard to the factor password it would be desirable for some accounts to replace the password by the smartcard. In comparison to the two factor solution this would increase the usability (e.g. nothing to remember or type) but decrease security (e.g. physical theft).

In comparison to the 1-factor password method the resulting 1-factor "what-you-have" authentication maybe better in some cases because the user does not have to remember nor type anything and no phishing is possible. On the other hand the user always has to carry the smartcard/token that could be stolen.

7.2. Transaction Signing instead of logging in

In the discussed method we sign random texts (session id, nonce) to prove the possession of a private key. A more critical application of signatures is to use them to sign actually meaningful texts like banking transactions, orderings, or payment confirmations. However, we then have to deal with the trusted interface problem which was not particularly relevant in the context of logins. A Smartphone or a PC alone cannot solve this problem because both have to be considered as potentially malware-infected. Nevertheless an approach of splitting trust becomes possible with a variant of the Indirect NFC-Login method: the transaction details are shown on both displays. For more details about transaction signing with NFC-Smartphone and NFC-smartcard see the recent paper [15].

7.2. Two Factors at once

Given the fact that there are several channels involved in the Indirect NFC-Login method it would be possible to apply the two-channel PIN method described in [16]. In addition to the response, the smartcard would compute a random permutation of the digits 0-9 and send it to the Smartphone.

The Smartphone sends the response and the permutation to the server and displays the permutation to the user. The user enters his PIN on the PC by clicking onto a virtual number pad with empty fields according to this permutation, and finally the PC submits the click positions to the server. Knowing the permutation and the message from the user, the server can retrieve the entered PIN while malware on the Smartphone or on the PC cannot.

8. Conclusion

We proposed the Indirect NFC-Login as an extension to what we call the direct NFC-Login proposed by Grosse et al.

Comparing the two methods we noticed that security properties are essentially the same. Regarding usability the direct method is preferable. The comparison of the deployability is ambivalent: On the one hand, the user needs an NFC-Smartphone for the indirect method. On the other hand we claim that the indirect method makes NFC-Login ubiquitous in the sense that every internet client is able to execute the login.

9. References

[1] E. Grosse and M. Upadhyay, Authentication at Scale. IEEE Security and Privacy Vol. 11, pp. 15-22, 2013.

[2] (2013) '2 Step Verification', Google Inc.; http://www.google.com/landing/2step/index.html (27 Nov 2013).

[3] (2011) 'TOTP: Time-based one-time password algorithm', Internet Engineering Task Force; http://tools.ietf.org/html/rfc6238 (27 Nov 2013).

[4] (2005) 'HOTP: An hmac-based one-time password algorithm', Network Working Group; http://tools.ietf.org/html/rfc4226 (27 Nov 2013).

[5] M. Tanaka and Y. Teshigawara. (2007) 'A method and its usability for user authentication by utilizing a matrix code reader on mobile phones', Information Security Applications, volume 4298 of Lecture Notes in Computer Science. Springer Berlin Heidelberg.

[6] J. J. L. Cobos and P. C. De La Hoz. (2008) 'Method and system for authenticating a user by means of a mobile device', Patent US20100070759 A1.

[7] K. Reinhardt and B. Borchert. (2009) 'Method and computer program product for providing authorized access to online accounts, Patent, WO/2011/069492.

[8] B. Dodson, D. Sengupta, D. Boneh, and M. Lam. (2010) 'Secure, consumerfriendly web authentication and payments with a phone', Conference on Mobile Computing, Applications, and Services (MobiCASE 10).

[9] (2011) 'eKaay – Smart Login', eKaay UG; http://ekaay.com/?lang=en (27 Nov 2013).

[10] (2012) 'tiqr', SURFnet BV; http://tiqr.org. (27 Nov 2013).

[11] (2011) 'mydigipass', VASCO Data Security International GmbH; https://www.mydigipass.com/ (27 Nov 2013).

[12] A. Czeskis, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz. (2012) 'Strengthening User Authentication through Opportunistic Cryptographic Identity Assertions', In the Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS). October 16-18, 2012.

[13] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano (2012) 'The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes', IEEE Symposium on Security and Privacy. San Francisco, CA, USA, May 21 2012.

[14] (2013) igeak, http://www.igeak.com/Product/Feature/188 (27 Nov 2013).

[15] M. Günther and B. Borchert. (2013) 'Online Banking with NFC-Enabled Bank Card and NFC-Enabled Smartphone', Information Security Theory and Practice. Security of Mobile and Cyber-Physical Systems. Lecture Notes in Computer Science 7886, Springer Berlin Heidelberg, 2013.

[16] B. Borchert and K. Reinhardt. (2007) 'Vorrichtung und Verfahren zur abhör- und manipulationssicheren Verschlüsselung für Online-Accounts, Patent DE2007052734B4.

