Ceremony Analysis: Strengths and Weaknesses

Kenneth Radke, Colin Boyd, Juan Gonzalez Nieto, and Margot Brereton

Information Security Institute
and School of Design
Queensland University of Technology, Australia
{k.radke,c.boyd,j.gonzaleznieto,m.brereton}@qut.edu.au

Abstract. We investigate known security flaws in the context of security ceremonies to gain an understanding of the ceremony analysis process. The term security *ceremonies* is used to describe a system of protocols and humans which interact for a specific purpose. Security ceremonies and ceremony analysis is an area of research in its infancy, and we explore the basic principles involved to better understand the issues involved. We analyse three ceremonies, HTTPS, EMV and Opera Mini, and use the information gained from the experience to establish a list of typical flaws in ceremonies. Finally, we use that list to analyse a protocol proven secure for human use. This leads to a realisation of the strengths and weaknesses of ceremony analysis.

Keywords: Ceremony, EMV, HTTPS, Opera Mini, security, privacy, provable security, humans

1 Introduction

In 1993 Bellare and Rogaway introduced a model for reductionist security proofs for cryptographic key exchange protocols [4]. Since this time, many cryptographic protocols have been accompanied by a reductionist security proof.

A reductionist security proof means that the security of the protocol is reduced to a known *hard* mathematical problem, such that if an advantage is achieved over the protocol, then there will be some significant advantage over the known hard problem. If a protocol is proven secure in this way then, as long as the "hard" mathematical problems remain sufficiently hard, the protocol is unbreakable within the defined security model.

Unfortunately, many protocols so proven to be secure in theory, have been found to be insecure in practice, when deployed in the real world. This inequality between the theoretical security and the actual security can be traced back to a deficiency in the security proof model. The mathematical security models while useful, especially for examining the security of a protocol in isolation, do not take into account the wide range of side channel attacks, social engineering, and interfaces to other protocols and the environment, which occur in the real world.

In 2007, Ellison wrote that a more robust method for examining the security of a protocol was to consider a security *ceremony* [7]. Ellison wrote with reference

to network protocols, but we can extend that work to any group of protocols. A security *ceremony* may be described as *protocols in their context of use*. For example, the protocol HTTPS provides a connection secure from eavesdroppers between two nodes on a network. However a security *ceremony* would include a user, viewing a website on their computer, and using HTTPS via their web browser running on the computer to securely connect to another computer on the network. We will show that while ceremony analysis is powerful enough to capture known attacks, each use case of a given set of protocols is a new ceremony and requires its own ceremony analysis.

1.1 Related Work

The concept of a ceremony was developed earlier than 2007 [8]. In the years since 2007, there has been an increasing trend to meld information security with the social sciences, as indicated by conferences both in the U.S.A[1] and in Europe[2]. This multi-disciplinary approach brings into context the human usage of information security systems. As Shostack and Stewart state, "...our approach to information security is flawed" and "the way forward cannot be found solely in mathematics or technology" [18].

Although little progress has been made regarding ceremonies since 2007, a number of researchers in different areas have agreed that ceremony analysis is a promising research direction. These research areas include formal methods, network security, and applied cryptography.

In the formal methods' security community, there has been a call to include parts of ceremony analysis in the formal methods' analysis of protocols [12]. This work has been further developed in Martina et al's more recent work in the PKI context [13]. Martina et al used the verification method outlined by Ruksenas et al. [16,17], adapted using Bella's goal availability principles [2], to address the open question that Ellison posed as to how to model human behaviour.

In the network security community, the concept of a ceremony has been used to describe protocols which include humans, and thus to create more robust security ceremonies [11]. Karlof et al. describe a concept of *conditioned-safe ceremonies*, based on a *defence-in-depth* approach adapted from the human reliability community. Central to their approach is the use of *forcing functions* whose property is to prevent a user from proceeding, until a critical step is completed.

In the applied cryptography community, Ellison's ceremonies have been used as a basis for modelling authentication ceremonies involving humans [5]. In the authentication ceremony described by Brainard et al., a human who still has their primary authentication details intact, the *helper*, vouches for another personally known human who has lost their authentication details (the *asker*). This vouching process, an extra factor in identification of the asker, allows emergency authentication details to be provided.

[1] `http://weis2010.econinfosec.org`
[2] `http://www.cl.cam.ac.uk/~rja14/shb10/`

There is a large body of work on such topics as phishing on the internet, and social engineering in general [6,10]. This reflects the common understanding that many security decisions are based on trust, such as trust in a brand, rather than the mathematical assurances of a correctly executed protocol. For this reason, ceremony analysis provides a more complete understanding of the issues surrounding the use of a protocol by a human, than protocol analysis alone.

1.2 Contribution

We reinterpret recently identified security flaws in the context of ceremonies, and use this information to establish a list of typical flaws in ceremonies. We apply the knowledge learned from analysing previously identified security flaws to analyse a protocol including a human which has been proven secure. In doing so, we show that ceremony analysis is powerful, in that it can capture and describe all of the known issues investigated, and highlight flaws in a protocol proven secure. However, the process yielded the knowledge that ceremony analysis is analysis of one particular "use case" of a (set of) protocol(s). This knowledge leads to the realization of a limitation of ceremony analysis, which is that if the context of the set of protocols is changed then what was secure may no longer be secure (a different context, even for the same set of protocols, is a different ceremony).

1.3 Outline

In the next section we give an overview of ceremonies and reinvestigate the Hypertext Transfer Protocol Secure (HTTPS) ceremony from Ellison. After this introduction to ceremonies, the analysis of the Opera Mini ceremony is shown. We analyse three ceremony investigations, including an investigation of an EMV (Europay, MasterCard and VISA) ceremony not shown due to space constraints, and discuss the findings. We then use the lessons learned from the analysis of these known flaws to analyse a protocol which has been proven secure using an adversarial security model. Finally, we outline the strengths and weaknesses of ceremony analysis.

2 Ceremonies

Ellison wrote about security ceremonies in 2007 [7]. In this paper, he attributed the name *ceremony* as being coined for this purpose by Jesse Walker. Ellison provided several central ideas in a network security context, which can be directly applied to cryptographic protocols in general. The properties of a security ceremony that we distil from Ellison's work are as follows:

- a ceremony is a superset of protocols;
- there is nothing out-of-band; and
- humans, when part of the ceremony, are explicitly included.

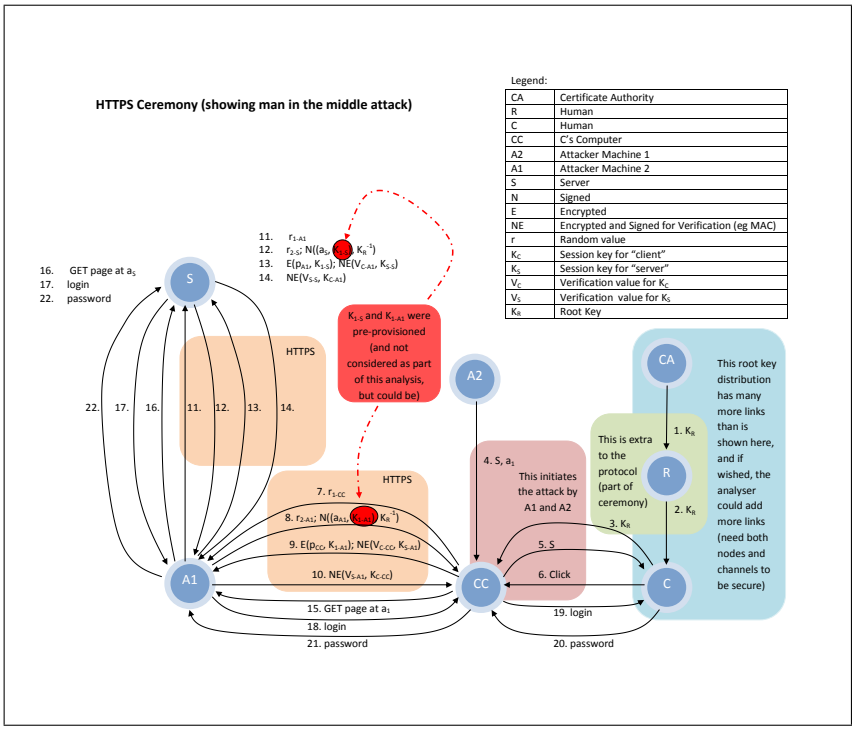

Fig. 1. HTTPS Ceremony, of Ellison (2007)

2.1 Ceremonies Example: HTTPS with MITM Attack

HTTPS is a protocol used on the internet to provide confidentiality and integrity to messages between two parties. An example HTTPS ceremony derived from Ellison's paper is shown in Figure 1. This ceremony has a number of parts, between multiple "nodes" or parties. First, on the right hand side of Figure 1, is the root key distribution part of the ceremony. The nodes in this key distribution process have been denoted by CA, R and C. Here the certificate authority is represented by CA, and R represents the registration authority which involves a number of human steps between the CA and the human party C. The human C will use the key from the CA on C's computer CC. The messages for placing the key on C's computer CC are shown in messages 1 to 3. Notice that there is no time scale on the ceremony.

The attack is shown between the user C, and the user's computer CC and the server S, in messages 4 to 6. The attack is carried out via two adversaries, $A1$ and $A2$. At some time after the user's computer CC is set up ready to take part in HTTPS, adversary $A2$ sends a name (server S's name) and an address (adversary $A1$'s address) to the computer CC. User C decides whether or not to proceed to the server based on the server's name alone, because the software

running on CC does not present both the name and address to user C, only the name.

From here, the ceremony proceeds as expected through messages 7 to 22, and hence the attack. User C's computer, CC, securely connects to adversary $A1$ (messages 7 to 10) using HTTPS, adversary $A1$ securely connects to server S (messages 11 to 14) using HTTPS, and then the adversary $A1$ faithfully relays communication between the user's computer CC and the server S. Specifically $A1$ passes on the login and password information, which adversary $A1$ now has in plaintext form for the future (note the decryption and re-encryption between messages 21 and 22 for the password, and similarly for the login). After message 22, adversary $A1$ is securely logged into server S, and is free to proceed as desired.

Ellison's example ceremony presumes that only the name of the target, and not the target's web address, is passed on to the human through the web browser in message 5, for the human to make their decision on. If this is the case, then this is clearly an issue that will result in the security of the ceremony being compromised. Some readers may suggest that this should not be the case any longer, due to such advances as extended certificates which have been introduced since 2007 (`http://www.cabforum.org/`). However, in a recent study by the authors which asked the participants to log their web usage security decisions for a week, not one participant based any of their security decisions in a week of web use on any of the information made available by the extended certificate enhancements [15]. Also, extended certificates are not yet mandated for use in HTTPS. Hence the issue remains current. Further, even if the address, as well as the name, is displayed to the user to base their security decision on, Ellison asks whether the human user will be provisioned ahead of time with the association between the address of the server and the name of the server, and the correctness of the name [7].

The above means that, in the ceremony shown in Figure 1, the user (C) believes that their computer (CC) is securely connected to the server (S). Indeed, CC is securely connected to *something*, just not the *intended* server. The point is that the HTTPS protocol is not broken, there are successful usages of the protocol between CC and $A1$, and between $A1$ and S. But the security ceremony, which includes the HTTPS protocol, is fatally flawed.

3 Opera Mini Ceremony

Opera Software ASA is a company which develops a suite of multi-platform web browsing software programs (`http://www.opera.com/company/`). Opera has had the greatest market share of any mobile web browser in the world, for the last 12 months [3]. There are different versions of Opera web browsers for different purposes. The three main variations of the browser being:

- standard Opera for PC/Mac
- Opera Mini for mobile telephones
- Opera Mobile for devices such as PDAs

[3] `http://gs.statcounter.com/#mobile_browser-ww-monthly-200911-201010`

3.1 Opera Mini Design

Opera Mini is the version for devices such as mobile telephones, which have restricted computing power and resources. Opera Mini has no full rendering engine on the device (`http://www.opera.com/mobile/specs/`). Instead, Opera has proprietary servers which handle the internet requests made on the mobile.

This process of sending requests to the internet via a server which handles the rendering and compresses the data before sending the resulting page back to the mobile telephone, has benefits both in a reduction of the computing power required on the device, and also reduced bandwidth requirements to the device which is running Opera Mini. The issue from a security point of view is that there is no *end-to-end* security. The requests from the mobile telephone to Opera's server are encrypted using Opera's proprietary encryption, but the messages are decrypted from Opera's proprietary encryption at the Opera server, and then the data is re-encrypted using standard HTTPS and the certificate of the actual target website (`http://www.opera.com/mobile/help/faq/#security`). As the Opera Mini FAQ on security reads:

> "To be able to do this translation, the Opera Mini server needs to have access to the unencrypted version of the webpage. Therefore no end-to-end encryption between the client and the remote web server is possible. If you need full end-to-end encryption, you should use a full web browser..."
> (`http://www.opera.com/mobile/help/faq/#security`)

3.2 Opera Mini Ceremony Analysis

Opera mini's use in a mobile phone is a quintessential security ceremony. There is one protocol between Opera's server and the internet, another protocol between the mobile telephone and Opera's server, and finally there is a human user making security decisions based on what they see on the browser on their mobile telephone.

Of particular interest in the Opera Mini ceremony is the use of standard icons to indicate security to the user. In, for example, Internet Explorer, which almost one in two desktop users currently use worldwide[4], the use of the *padlock* symbol means that the connection between the user and the website the user is interacting with is secure via use of HTTPS. By secure, we mean that confidentiality and integrity are assured such that no computer on the path between the user and the website can decrypt any of the information or change the message that is sent by the user or the website. The padlock icon is used similarly in all other major browsers.

However, as shown in Figure 2, Opera Mini displays a padlock symbol (top right of picture) when there is not end-to-end security. This means that Opera Mini users, who know what the padlock symbol means in other browsers, are led to believe that they have a confidential connection to the website they are viewing, when they do not.

[4] `http://gs.statcounter.com/#browser-ww-monthly-200907-201008`

Fig. 2. Opera Mini Secure Connection (`http://www.opera.com/mobile/demo/` viewing NAB's secure logon page)

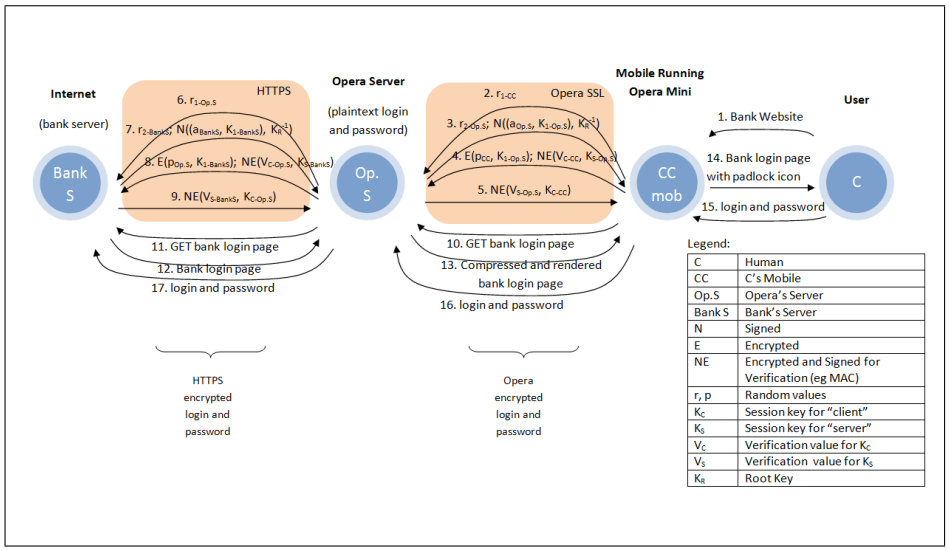

Fig. 3. Opera Mini Ceremony

Figure 3 describes the Opera Mini ceremony. The ceremony begins with the user of a mobile telephone typing the address of their bank's website into the Opera Mini web browser (message 1). A process similar to HTTPS then occurs

between the mobile telephone and Opera's Server (approximated by messages 2 to 5). As Opera ASA states:

> The communication is protected by 256-bit RC4 and the key exchange is done by 1280-bit RSA. All hashes are created using SHA-256. These are the algorithms used by most SSL sites today. (`http://www.opera.com/mobile/help/faq/#security`)

A HTTPS connection is also formed between Opera's server and the bank's server (messages 6 to 9). Once this is complete, the request for the page is passed through to the bank (messages 10 and 11), and the bank replies with its customer login page (message 12). The Opera server renders this page, and sends the compressed output to the user's mobile telephone device (message 13). On the mobile telephone, Opera Mini then displays the webpage, including the padlock symbol (message 14). The user sees the padlock symbol, and chooses whether to input their login information and password. If the user does enter their login and password (message 15), then this is sent back to the bank's server via the Opera encrypted channel (message 16), decrypted at the Opera Server, and then re-encrypted and sent on to the bank's server via the HTTPS encrypted channel (message 17).

In a recent study, our research team investigated security decisions made by users in a week of standard web usage. We found that most users made the choice of whether or not to interact with websites that had direct financial interfaces, such as banks or online retail, based on whether or not the padlock symbol was shown [15]. Users presumed that a padlock meant that no one, apart from the website they were communicating with, could see their financial details and confidential information, such as login and password, in plaintext form. Opera's intimation of confidentiality by the depiction of the padlock symbol is not in keeping with Opera's statement in the Opera Mini FAQ which says "if you need full end-to-end encryption, you should use a full web browser..." (`http://www.opera.com/mobile/help/faq/#security`).

Interestingly, while the plaintext state of messages through the Opera Server clearly is a security issue and probably not realised by most Opera Mini users, the design has some security benefits. If the user trusts Opera Mini with all their communication with every party they communicate with on the internet, then this design of accessing the internet through a proxy provides essentially anonymous internet usage, as well as protection against various JavaScript-based malicious software (malware).

4 Lessons Learned

By re-investigating known security flaws from a ceremony point of view, we identified a set of common flaws. These ceremonies included the EMV ceremony described by Murdoch et al., but these were left out of this paper due to space constraints [14]. This list included:

– each individual protocol remained secure, but the critical security information was not passed from one protocol to the next;
– the information passed on to the human was inadequate for the human to have any chance of making a correct decision;
– it is clear that lessons long since learned for protocols, have not been transferred into security ceremony knowledge.

While ceremony analysis has been demonstrated to capture known flaws, and therefore is useful, the technique is not without pitfalls. The most significant flaw is highlighted by our definition for a ceremony, stated in Section 1, which was that security ceremonies were *protocols in their context of use*. This means that, even if the underlying protocols are found to be secure for a given context, they may well not be secure in even a slightly different context, leading to the situation of requiring a new ceremony analysis for the same set of protocols in each new context.

All of the ceremonies examined have been *use cases*, the *context of use*, of the underlying protocols, and therefore the first job of the ceremony analyser is to create a list of use cases to create a rigorous security proof for. Of particular concern for the ceremony analysis technique are areas where the context of use for the protocols, for a specific ceremony, do not yet exist. Ceremony analysis will therefore, by necessity, trail behind users' use of any given system. For example, the people responsible for the security of new smart card driver licenses will only be able to analyse certain security ceremonies once users of the smart card have been interacting with (potentially previously unknown) third parties. This interaction with new third parties may be a new context, and hence a new ceremony will be created which will be able to be analysed only in retrospect. This is a significant step down from the ideals of provable security, which aims to ensure that, once a protocol is proven secure, it will be secure regardless of how it is used.

Therefore the common flaws revealed in the ceremonies analysed to date suggest these assessments which should be completed on security ceremonies prior to deployment.

– Look for protocol-like deficiencies, such as outlined in [1]. Treat each constituent protocol as a node in the ceremony, and check that nonces and identification are being passed between nodes.
– Ensure that key cryptographic information is being transferred between nodes in the ceremony.
– If the ceremony includes a protocol including a human as part of the protocol, and if the protocol comes with a proof of security, re-examine the proof of security for the assumptions that were made concerning the human.
– Examine the human's role in the ceremony. If the only way for the human to accomplish their goal is via a particular route through a security decision point, the human will take that route.
– Examine the human-factor considerations of the ceremony. These issues include how many items a human can remember (for example, web address and store name pairs, as per the HTTPS ceremony) and the prior knowledge and

education required. For example, in approving the usage of a HTTPS ceremony, do humans realise that the most critical information is the address? Our recent study indicated that they did not.

5 Investigation of a Provably Secure Protocol

In 2008, Gajek et al. expanded on Bellare and Rogaway's concept of *practice oriented provable security* [3]. The significant enhancement that Gajek et al. made to previous security models was that they proved a protocol including a human to be secure [9]. They achieved this by adding formal actions *render* and *recognise* to a security model. *Render* is the process of a web browser rendering a HTML page, based on the browser's state, and presenting that page to the user. *Recognise* is the process of a user viewing the webpage, judging if the *Human-Perceptable Authenticator* (HPA) is correct, and outputting either *true* or *false* depending on the results of that test.

The protocol that Gajek et al. proved to be secure, what they called *browser-based user-aware mutual authentication over TLS*, is a non-trivial security ceremony. In the protocol, there is a user who has a computer, a browser running on the user's computer, and the user is interacting with a server via their computer's browser. Gajek et al. take the important step of extending the definition of the underlying TLS (Transport Layer Security) protocol to include the human user. In the protocol, the user types the address of the server into their browser, the TLS HTTPS connection is created between the server and the browser, the server then sends a HPA to the user via the browser (which renders the HPA). If the user recognises the HPA, then they type in their login credentials. In this way, the server is authenticated to the user (via the HPA) and the user is authenticated to the server (via the traditional login and password technique). For the full Gajek et al protocol, see [9].

We analysed the Gajek protocol using the lessons learned via our analysis of the previously outlined security ceremonies. We make the following observations.

- The protocol begins with the human typing in the web address of the server. This immediately removes one significant source of failing by the user (web address of target incorrect), which was specifically outlined in the HTTPS Ceremony shown in section 2.1. So the question is only, "Could an adversary, who is not the server, supply the user with a HPA which will cause the user to enter their username and password?"
- Many assumptions are rolled into the browser's *render* and the human's *recognise* capabilities. For example, since it is not specified in the protocol, there is every chance that a website (and browser) designer implementing this protocol would not put in any check to ensure that the HPA (typically a picture) is fully shown *before* the user can type in their user name and password. On a slow connection, users may well type in their details prior to seeing some, or all, of the HPA. Further, even if such a check was put in place (picture fully downloaded and shown prior to displaying login details entry form), the protocol could still be broken via sending an *all black* or

all grey with a red cross in the middle picture. Many users may view these pictures as a download or rendering fault, and still enter their user name and password.
- Another potential attack is to degrade or pixelate the picture. There will be storage space and bandwidth decisions made concerning the file format, size, and resolution of the HPA, by at least all three of the owners of the server, the webpage developer, and the web browser developer. As written, these decisions are left to the individual developers with no necessity for a common technique. The essential message, from both this and the prior point, is the need for protocol developers to include in their protocol design, and hence protocol proof, the specification of the critical elements of the designs of the interface to the human.
- How does the user know that *this is the protocol*? The user does not know the algorithm, does not know that suddenly they should be waiting for a HPA. This suggests that there is no need to attack this protocol at all, and the adversary should create a different protocol. Therefore, once this issue is realised, as part of a security *ceremony* potential solutions such as side-channel instructions to the user about the protocol may be necessary.

6 Conclusion

We have shown that security flaws in complex systems of protocols, with human interaction, can be analysed using security ceremonies. The analysis of the EMV smart card ceremony (omitted due to space constraints) and the Opera Mini ceremony, followed by the analysis of the TLS protocol which had been proven secure for human use, shows that a ceremony analysis is capable of capturing a greater range of security flaws than protocol analysis alone.

In the process of analysing these ceremonies, we have constructed an approach for analysing further security ceremonies. We also highlight the role that the designer plays in ensuring that the ceremony is secure. This role necessitates a grounding in security considerations, and similarly that creators of protocols are aware of typical design considerations at the human-computer interface.

Finally, the realisation that security ceremonies are essentially *use cases* of the underlying protocols, warns against the presumption that a protocol shown secure in one ceremony will mean that the same protocol is secure in another ceremony. The development of a list of use cases for the protocol, or device such as a smart card, becomes critical, as is the use standardized protocols that either are provably secure or have been rigorously scrutinized. This work may be similar to the construction of a safety case for mission critical systems.

Acknowledgments. The authors acknowledge and appreciate the suggestions by Jason Reid and Douglas Stebila as to appropriate real-world groups of protocols we could conduct ceremony analysis on. The authors also appreciate the quality and differing viewpoints exhibited by the blind reviewers, which have directly lead to improvements in this paper.

References

1. Abadi, M., Needham, R.M.: Prudent engineering practice for cryptographic protocols. IEEE Trans. Software Eng. 22(1), 6–15 (1996)
2. Bella, G.: Formal correctness of security protocols. Springer Verlag (2007)
3. Bellare, M.: Practice-oriented provable security. In: Damgård, I. (ed.) Lectures on Data Security. Lecture Notes in Computer Science, vol. 1561, pp. 1–15. Springer (1999)
4. Bellare, M., Rogaway, P.: Entity Authentication and Key Distribution. In: Stinson, D.R. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 773, pp. 232–249. Springer (1993)
5. Brainard, J.G., Juels, A., Rivest, R.L., Szydlo, M., Yung, M.: Fourth-factor authentication: somebody you know. In: ACM Conference on Computer and Communications Security. pp. 168–178. ACM (2006)
6. Dhamija, R., Tygar, J., Hearst, M.: Why phishing works. In: Proceedings of the SIGCHI conference on Human Factors in computing systems. p. 590. ACM (2006)
7. Ellison, C.: Ceremony Design and Analysis. Cryptology ePrint Archive, Report 2007/399 (2007), `http://eprint.iacr.org/`
8. Ellison, C., Dohrmann, S.: Public-key support for group collaboration. ACM Trans. Inf. Syst. Secur. 6(4), 547–565 (2003)
9. Gajek, S., Manulis, M., Sadeghi, A.R., Schwenk, J.: Provably Secure Browser-Based User-Aware Mutual Authentication over TLS. In: Abe, M., Gligor, V.D. (eds.) ASIACCS. pp. 300–311. ACM (2008)
10. Herzberg, A.: Why Johnny can't surf (safely)? Attacks and defenses for web users. Computers & Security 28(1-2), 63–71 (2009)
11. Karlof, C., Tygar, J.D., Wagner, D.: Conditioned-safe ceremonies and a user study of an application to web authentication. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2009, San Diego, California, USA. The Internet Society (2009)
12. Martina, J., Carlos, M.: Why should we analyze security ceremonies. In: Applications of Logic in Computer Security. The 15th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (2008)
13. Martina, J.E., de Souza, T.C.S., Custodio, R.F.: Ceremonies Formal Analysis in PKI's Context. In: CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering. pp. 392–398. IEEE Computer Society, Washington, DC, USA (2009)
14. Murdoch, S.J., Drimer, S., Anderson, R.J., Bond, M.: Chip and pin is broken. In: IEEE Symposium on Security and Privacy. pp. 433–446. IEEE Computer Society (2010)
15. Radke, K., Boyd, C., Brereton, M., Nieto, J.G.: How HCI Design Influences Web Security Decisions. In: OzCHI. ACM (2010)
16. Ruksenas, R., Curzon, P., Blandford, A.: Detecting cognitive causes of confidentiality leaks. Electr. Notes Theor. Comput. Sci. 183, 21–38 (2007)
17. Ruksenas, R., Curzon, P., Blandford, A.: Modelling and analysing cognitive causes of security breaches. ISSE 4(2), 143–160 (2008)
18. Shostack, A., Stewart, A.: The New School of Information Security. Addison-Wesley Professional, Upper Saddle River, N.J. (2008)