Towards a Secure Human-and-Computer Mutual Authentication Protocol

Kenneth Radke[1,2] **Colin Boyd**[1] **Juan Gonzalez Nieto**[1] **Margot Brereton**[2]

[1] Information Security Institute
[2] School of Design
Queensland University of Technology
Email: {k.radke, c.boyd, j.gonzaleznieto, m.brereton} @qut.edu.au

Abstract

We blend research from human-computer interface (HCI) design with computational based cryptographic provable security. We explore the notion of practice-oriented provable security (POPS), moving the focus to a higher level of abstraction (POPS+) for use in providing provable security for security ceremonies involving humans. In doing so we highlight some challenges and paradigm shifts required to achieve meaningful provable security for a protocol which includes a human. We move the focus of security ceremonies from being protocols in their context of use, to the protocols being cryptographic building blocks in a higher level protocol (the security ceremony), which POPS can be applied to. In order to illustrate the need for our approach, we analyse both a protocol proven secure in theory, and a similar protocol implemented by a financial institution, from both HCI and cryptographic perspectives.

Keywords: Ceremony, human, HTTPS, TLS, security, privacy, provable security, authentication

1 Introduction

Humans have had a need to communicate securely for thousands of years, with documented evidence of the use of a scytale (used for transposition ciphers) as early as 475BC (Mollin 2005). With the proliferation of computers over the last half century, and the capacity computers provide for cryptanalysis, calculations in confidentiality-ensuring cipher schemes have quickly become too complex for the general populace to complete by hand. This has led to a situation where trust is required by the general user with regards to whether their communication and assets remain private and secure. For example, typical cryptographic security solutions may include hash functions. The general populace is unaware what hash functions are, they certainly do not understand the functionality they provide, and they do not know which hash functions are being used on their behalf and which ones are known to be insecure. We focus on mutual authentication, using *browser-based*[1] protocols, particularly providing the human with assurance that they are communicating with the party they intend to be communicating with. We blend concepts from the provable security community, network security community, human computer interface (HCI) design community, and sociotechnical community. Our contribution includes a presentation of deficiencies in models and protocols previously published. We also outline a set of minimum guidelines that human-computer authentication protocols should have over and above computer-computer authentication protocols.

We blend HCI research and cryptographic research to create a useful protocol-including-a-human proving methodology. We highlight human-centred considerations which must be included when analysing such a protocol, and outline central issues in assessing security which requires a shift in thinking. To motivate our approach we analyse both a protocol proven secure in theory, and a similar protocol deployed in practice by a financial institution, from both HCI and cryptographic perspectives.

2 Background and Related Work

To create a mutual authentication protocol between a human and a computer, which is secure with respect to the common understanding of confidentiality and integrity[2], a number of fields of research need to be examined. The combination of an adversary having the capabilities of a computer and one of the parties being a computer, means that lessons learned in the non-computer world cannot be directly applied. For example, an attack in a physical environment (such as a robbery) may need a success rate of, at worst, one in ten to be worthwhile for the perpetrator; whereas in the cyberworld attacks that work one time in a million can be seen as successful (Shostack & Stewart 2008). So this suggests cryptography with enough security to withstand a computer attack is required, and yet humans are known to have neither the patience, nor the capacity, to compute the necessarily large numerical values required for modern cryptography. Further, if modern cryptography is used, then the human loses visibility, the process becomes non-transparent, and hence, for the general populace, blind trust is required that the data is secure.

Further, cryptography is no longer required only by nation states, the military, or secret lovers. Today, the general populace, in developing and first world countries, have huge amounts of data and communications they would like protected, and there are many real-world settings, such as smart phones, RFID tags, and e-commerce, that require protected communication. This means that, even if we could somehow remove the advantage that computers provide

Copyright ©2012, Australian Computer Society, Inc. This paper appeared at the 10th Australasian Information Security Conference (AISC 2012), Melbourne, Australia, January-February 2012. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 125, Josef Pieprzyk and Clark Thomborson, Ed. Reproduction for academic, not-for-profit purposes permitted provided this text is included.

[1] Defined by Gajek et al. as *protocols realizable within the constraints of commodity Web browsers* (Gajek et al. 2008).

[2] For definitions of authentication, integrity and confidentiality see (*ISO/IEC 27001 Information technology - Security techniques - Information security management systems - Requirements* 2005, Menezes et al. 1996).

the cryptanalyst over human capabilities, for example by using CAPTCHAs (eg in (Dziembowski 2011)) or POSHs (Daher & Canetti 2008), the amount of encrypting and decrypting required makes anything more than human involvement in the cryptosystem at critical authentication steps unrealistic.

In this background section, with our goal being a secure authentication protocol which is not only usable by humans, but also understandable by humans in such a way that blind trust is not required, we will cover the ideals of provable security, security *ceremonies*[3], and we will examine some HCI design and sociotechnical considerations.

2.1 Provable Security

In 1993, Bellare and Rogaway responded to a need to add more rigour to authentication protocol analysis (Bellare & Rogaway 1993*a*). They applied reduction techniques for proving algorithms[4] to authentication and key distribution protocols. These techniques had been previously used by Goldwasser, Micali, Rivest, Blum and Yao in other cryptographic primitive settings ((Yao 1982, Goldwasser & Micali 1984, Blum & Micali 1984, Goldwasser et al. 1988) cited in (Bellare & Rogaway 1993*a*)). The critical concept of a reductionist proof of security is that, if an adversary can break the protocol, then the adversary can also break the underlying cryptographic primitive.

Perhaps a more significant contribution of Bellare and Rogaway's 1993 work was the concept of *practice-oriented* provable security (POPS). Provable security research prior to this had been based on only theoretical primitives (Bellare 1999), such that at the time of Bellare and Rogaway's 1993 papers provably secure cryptographic primitives tended to be much less efficient than primitives used in practice (Bellare & Rogaway 1993*a*,*b*). Since there was no intersection between provably secure cryptographic primitives and the primitives used in practice, provable security pre-1993 was just theory. With the addition of an idealised model, the random oracle, protocols using the primitives used in practice could have security proofs developed.

Unfortunately, the concept behind POPS has not extended as far as required into protocol design, particularly in the area of protocols which involve humans. A fundamental ideal of POPS is that at no point should a protocol be able to be broken without breaking the underlying cryptographic primitive, and hence the protocol should not be weaker than the underlying primitive. In reality, particularly with respect to humans [5], this is not the case. For example, humans have shown themselves to be susceptible to many social engineering attacks, which allow the theoretically secure protocols (which have a reductionist proof) to be broken in practice. Further, humans do not execute a protocol as the protocol designer thought they would. The reasons protocols, proven secure mathematically, are broken when humans use them, can be summarised to *the model used for the security proof was insufficient*. Such a statement masks a variety of sources of deficiency, some of which include:

- modelling a human is too difficult, and hence humans are either left out of the model and security proof, or else humans are given unrealistic powers such as being expected to follow the protocol 100% correctly, 100% of the time; or they are expected to completely forget previous actions.
- The model, and hence the security proof of the protocol, does not include critical *out-of-band* (OOB)[6] communication and necessary setup steps prior to the protocol running.
- The protocol definition, and hence the security proof based on the model, does not include the complete design (for one example, see Section 3.1). Most particularly, decisions that affect security, particularly HCI decisions, are left out of the protocol definition and are hence being made by non-security-aware practitioners.

Two promising directions in the provable security community have been made by Hopper and Blum (Hopper & Blum 2001), and by Gajek et al. (Gajek et al. 2008). Hopper and Blum's contribution was to provide a goal of creating (α, β, t) protocols for use by humans, in which at least $(1 - \alpha)$ of the human population can do what they need to do, in at most t seconds, with probability of correct execution of the protocol greater than $(1 - \beta)$. This data could be collected empirically, and their idea was to create light-weight cryptographic protocols that would have a mathematical proof of security, with ideally 90% of the population executing the protocol correctly inside 10 seconds, 90% of the time (Hopper & Blum 2001). Unfortunately, the protocol they suggested resulted in 10% of the population executing the protocol correctly inside 300 seconds, 80% of the time, and has gone on to become the basis of light-weight protocols for constrained devices, such as RFID, rather than human executable protocols (for example, see (Juels & Weis 2005, Hammouri & Sunar 2008, Bringer et al. 2006)). However, the concept of combining empirical evidence of usability with a security proof is a promising direction.

Secondly, Gajek et al. presented a protocol for mutual authentication between a human and an online institution, via the web (Gajek et al. 2008). This protocol, discussed in depth in section 3.1 and the basis of our proposal, has a number of innovative and useful features. Firstly, for the purposes of the security proof, the human is separated from their computer and web browser, so that the authentication between the human and a server has three parties, being the human, the human's computer with a web-browser, and the server. Secondly, the human and the human's computer are given specific functions in the security proof model. These functions were that the browser on the human's computer *renders* a webpage (based on browser state), and the human must be able to *recognise* what Gajek et al. called a *human perceptible authenticator* (HPA) (Gajek et al. 2008). The HPA can be anything, but in the Gajek et al. protocol the HPA was an image which was previously selected by the user and sent to the server. By adding these functions, the human's involvement is partitioned from the non-human protocol messages, and a formal proof of security is created. The proof concludes with the security of the protocol being bounded by the probability of a human to recognise their previously chosen HPA from the set of all possible values (specifically

[3] *Ceremony* (a term coined by Jesse Walker) analysis is protocol analysis with the human interaction explicitly included (Ellison 2007).

[4] These reductionist proof techniques were collectively called *provable security*.

[5] Beyond human involvement and potential social attacks, information is leaked concerning otherwise secure protocols via means such as observing computation time and power consumption, collectively known as *side-channel attacks*.

[6] *Out-of-band* channels are auxiliary channels, such as receiving an email which must be responded to as part of a signup process on a website.

taking into account other values which a human would find indistinguishable from their chosen HPA).

This technique of creating a protocol proof with the human assumptions being included but partitioned in such a way that a human trial will inform how secure the protocol is, is a significant step forward in the quest to prove protocols secure for human use. However, a complete design, informed by iterative cumbersome protocol-specific human studies following each new protocol design and developed proof, would potentially take years with no guarantee of success. Human protocols do need to be verified via human trial post-theoretical proof, however simply writing a security proof in terms of the human is not sufficient and a method of arriving at a more likely to succeed design is required.

Modern cryptography has matured enough, and a necessity for provably secure human-computer protocols has become critical, such that a timely paradigm shift concerning the building blocks of secure protocols is required. Just as Bellare and Rogaway defined POPS in 1993 (Bellare & Rogaway 1993*a,b*), thus shifting the focus to protocols and primitives in use at the time (Bellare 1999), we propose that there is now a requirement for a further paradigm shift, to move to a higher level of abstraction. That is, to treat building blocks for human interaction protocols, such as HTTPS, as primitives, and to create security proofs based on that in the interests of creating protocols better suited to humans. In this way, we propose *POPS+*. The technique remains the same, as does the quality of the proof. That is, if you believe that HTTPS is secure, and a reduction can be made from the security of HTTPS to the security of a protocol, then, as long as there remains no program that can break HTTPS, the protocol will remain secure.

2.2 Ceremonies

In recent years there has been a recognition and concerted effort to include the social sciences in information security in the research community[7]. This multidisciplinary 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" (Shostack & Stewart 2008).

The concept of a security ceremony has been used by Ellison, in network security settings, to capture the human element in the protocol usage (Ellison & Dohrmann 2003, Ellison 2007). In Ellison's main paper on the topic in 2007, the human was modelled as another node on the network, and hence a part of what must be considered from a security point of view. At the time, this innovation did not lead to any formal proofs of security, and only initial work was presented on how the human nodes could be modelled, but simply having the human as a node on the network, distinct from their computer, allowed certain attacks to be clearly presented and demonstrated. Most particularly, the technique demonstrated an attack which exploits an interface design which hides information, that the computer has, from the human who needs the information to give that human any hope of making an informed decision (Ellison 2007).

Since 2007, security ceremonies have been investigated in the fields of formal methods, including the PKI context, which provided some early steps on how a human may be modelled (Martina & Carlos 2008, Martina et al. 2009); in applied cryptography which added extra human elements to allow for humans to vouch for other humans as an extra factor in identification (Brainard et al. 2006); and in the network security community with a focus on a *defence-in-depth* approach, via use of *forcing functions*[8] (Karlof et al. 2009).

Recent research has shown that ceremony analysis is protocol analysis in its context of use (Radke et al. 2011). This raised concerns about conducting a more complete analysis, in particular including humans in a protocol's security proof, as a method for proving the security of the non-human part of the protocol being investigated. For example, a protocol such as HTTPS (HTTP over TLS), can be used in a variety of ways on a variety of devices. If the device, method, and user of the protocol are included in the ceremony, then many ceremonies for HTTPS which will be widely used have not been created yet (and the devices on which they will be used have not been created yet). This viewpoint means that proving the security of HTTPS via use in a ceremony will create a proof of security for HTTPS which is applicable to only that ceremony.

2.3 HCI and Sociotechnical considerations

HCI research on browser-based authentication protocols has revealed much concerning what humans can, what humans will not, and what humans cannot, do, drawing over the years from what Harrison et al. have identified as three broad paradigms of HCI research – a-theoretic, cognitive and situated (Harrison et al. 2007). Lessons can be learned from initial work by Simon (Simon 1969, 1996), which showed us the boundaries of human short term recall, and cognitive load issues, through to *specific controlled studies on decision making in use of security systems.* An example of such research is by Schechter et al. who created a study in which bank websites were progressively changed, to become less and less secure, and the researchers determined whether the participants continued to enter their password into the website (which they did) (Schechter et al. 2007). Recent work has indicated that a recent security improvement, which attempts to provide users with the necessary authentication information via the use of Extended Validation Certificates[9], and the associated inbuilt functionality in current browsers to colour code and present typically real world company name information to the user, is not being used by web-users in their web security decision making (Radke et al. 2010).

Dourish has provided a bridge between social science and HCI design, contributing significantly in areas such as defining and using context (for example, in Dourish (2004)). Of specific concern, when defining context, was the impression (still common seven years later) that context is fixed, explicit and can be adequately captured by explicitly measurable information rather than something that is "...being continually renegotiated and defined in the course of action" (Dourish 2004). One simple application of the concept of context is the case of the *rushing user*[10]. As Dhamija et al. describe, security is typically not the

[7] As can be seen by workshops and conferences such as SHB, WEIS, and SOUPS.

[8] A core property of a *forcing function* is to prevent a user from proceeding, until a critical step is completed.

[9] Extended Validation SSL Certificates – The Certification Authority/Browser Forum. `http://www.cabforum.org/`.

[10] A *rushing user* is used by Kumar et al. to describe a user who, in a rush, takes the shortest path through a protocol, skipping steps which are not required for subsequent steps to work (Kumar et al. 2009).

primary task and hence users may not notice security indicators or read warning messages (Dhamija et al. 2006). There is also a body of work which focuses on achieving security by aligning what a system does with the user's mental models of that system (Smith 2003, Yee 2004). As Smith states, "Repeatedly, I ended up with problems because what computers are doing with cryptography doesn't match the mental model that humans have - end users as well as system programmers (Smith 2003)." More recent work includes Chiasson et al.'s research into constructing a set of design principles for security management systems (Chiasson et al. 2007).

While the concept of aligning the actual system to the user's mental model of the system (or vice versa) is useful at a guiding level along the lines of "the user must understand what the system is doing, and what the response to her actions will be," the concept of the human cognitive model that exists prior to the situation is a contentious one. There is significant evidence that people co-construct meaning using embodied competencies and situational circumstances (Suchman 2007). Suchman argues understanding conversations and interactions, as dynamic co-constructions, could prove more useful for designers of human-machine interactions. The lesson we take from this body of work is the necessity for the user to be in control and to have visibility of (and to understand and actively participate in), ideally, the cryptographic authentication processes. This is in keeping with the central concept of Norman's popular design book, which is "when people have trouble with something, it isn't their fault - it is the fault of the design (Norman 2002)."

3 Human Protocols

We now focus our attention on protocols involving humans, specifically two cases, from which we will draw several critical lessons. The first case is by Gajek et al. and describes mutual authentication via the web, from which we will learn techniques used for proofs (Gajek et al. 2008). The second case is a comparative study, by Kumar et al., for pairing methods for previously unassociated devices over some human-imperceptible communication channel (such as bluetooth). This section will show us several important aspects of a human protocol, such as being resistant to the already discussed *rushing user* (Kumar et al. 2009).

3.1 Provably Secure Browser-Based User-Aware Mutual Authentication over TLS

In the past, it has been typical for papers analysing human protocols to present a range of known attacks, such as naive keylogging attacks, phishing attacks, eavesdropping, shoulder surfing, etc, and then, in an informal way, describe how their protocol addresses these concerns, perhaps via statistical analysis on a small set of users. This process remains widely used today (for example, Oorschot & Wan (2009), Arumugam & Sujatha (2010)). This is the style of protocol creation and analysis that the provable security movement of the past twenty years has sought to supersede. Therefore it was a significant step, by Gajek et al., to create a proof of security for a protocol involving a human (Gajek et al. 2008). A sketch of their protocol follows:

1. The protocol is between a server, a human's computer running a web browser (which has state), and the human.
2. Before the protocol begins, the human has selected a HPA and provided that HPA to the server. The HPAs suggested by Gajek et al. are a personally selected image or voice recording.
3. Both the server and the human's computer have authentication certificates and associated private keys, and a secure TLS connection is established between the browser and the server, when the browser on the human's computer opens the server's webpage. This process authenticates the server to the human's browser and from the human's browser to the server.
4. The server sends the human the HPA that the human has stored with the server (by completing a lookup of the human's browser-specific certificate, to know whose HPA to send), via the web browser which *renders* the HPA for the user, and this authenticates the server to the human.
5. Having *recognised* the HPA, the human sends the server their traditional login and password, thus authenticating the human to the server.

Investigation of the Gajek et al. protocol, model and proof reveals a number of salient points. These points may be categorised into *HCI issues* and *cryptographic issues*.

3.1.1 HCI Issues

For the points of interest that can be drawn from the Gajek et al. case, we will assume the HPA is an image (though these comments apply equally to voice and several other types of HPA). As stated in section 2.1, one of the reasons protocols proven to be secure fail, when subjected to use by a human, is due to the protocol specification not extending far enough into the HCI implementation. Thus, HCI designers, who are not security professionals, are making decisions that security professionals should have made. Issues that could result from the Gajek et al. protocol include:

1. Perhaps the most significant issue is requiring the designer to ensure that at least the image is fully displayed (ie images have not been turned off in the browser, and the image is fully downloaded) before the login and password box is presented to the user. Otherwise, there is no authentication from the server to the human, not even *potentially* any authentication from the server to the human, and authentication from the server to the human is the aim of the protocol. This goes beyond the rushing user concern, which this protocol does not resist at all, since the human can enter their login and password regardless of what image, or whether an image, is sent.
2. As soon as multiple people send images to a server, design decisions will be made regarding what format to store them in, what size to store them in, and what resolution to store them in. This will be done to ensure only a fixed amount of storage is used, and that similar quality images are used. The end result being that some images (which were too small or too low quality) may be rejected, and other images will lose significant detail.
3. Since the decisions at the client end are also not specified, different designers of website login forms will make different decisions about how to display the images. These decisions include the

shape of the image (at least, portrait or landscape) and the size of the image area on the webpage, which will all impact how many HPAs are human distinguishable from the complete set of HPAs.

The authors have seen a variant of the Gajek et al. protocol implemented by a financial institution. In this real world example, the user does not have a certificate, and instead the user's username is sent from the user to the bank, which the bank uses to identify which HPA to send back to the user. Upon the receipt of the HPA from the bank, the user sends their traditional login and password information to the bank.

Exploring this real world example is worthwhile to determine the sorts of design decisions that can be made by implementers of systems. Design decisions, that the creators of this login ceremony have made, include:

1. The bank's users are presented with a set of images to choose their HPA from. That is, the bank has overcome the issues concerning the range of image sizes, shapes, formats, resolutions etc, by providing the set of images to choose from. Unfortunately, this set of images is quite small, less than 20, so the dictionary space $|W|$ of this part of the HPA is quite small.
2. The implementers have added a *pass-phrase* which the users submit when they select their image in the once-off setup stage. Both the image and the passphrase (two parts to this HPA) are sent from the bank to the human at each login.
3. The bank's login proceeds without the image part of the HPA being downloaded. That is, even if the user turns off image downloads in their browser, the login and password entry fields still appear and the user can still login to the system.
4. This protocol is in no way rushing-user resistant. That is, the user can enter their login and password without looking at the HPA at all, and hence the protocol can be completed without the *recognise* task being executed.

3.1.2 Cryptographic Issues

The main cryptographic issues that surround the Gajek et al. protocol are entwined in human issues. From a cryptographic point of view, both security of the channel and authentication of the two parties is achieved by the use of HTTPS and certificates at both ends (TLS in *client authentication* mode). The reason why HPAs are used is due to the recognition that users do not check, know to check, know how to check, certificates. So there is an interesting combination in the security proof where effectively authentication of the user is provided by the user's certificate (wrapped in various cryptographic primitives such as keyed hashes), and authentication of the server is achieved via the HPA. This is in contrast to the words used in the paper, which clearly and intuitively state that TLS ensures that the browser knows it is communicating with the server, and the server knows that it is communicating with the browser, at which point the respective keys (HPA server to human; password human to server) can be securely exchanged. Once the HPA is recognised by the human, the server is authenticated to the human; and once the password is matched by the server, the human is authenticated to the server (Gajek et al. 2008).

There are three central observations:

1. Essentially the server's *password* (user's HPA) is being sent to the human before the human has been authenticated. Most particularly, the separation of the human from the human's browser-computer combination, means that while the browser has been authenticated to the server via the browser's certificate, anyone, especially someone other than the intended user, could be sitting at the terminal. This would allow an adversary, sitting at the user's terminal, to acquire the HPA and later masquerade to the user as the server. If we are to use the HPA as a real indicator of authentication and hence security of the system, then there is no difference between a server *sending a HPA to an unauthenticated human* and a human *sending a password to an unauthenticated server*. This weakness is a result of a limitation of the security model used for the protocol proof, since these sorts of attacks were not modelled.
2. Further, in the real world implementation, since the human's browser has no certificate, then the server is sending the HPA without authentication at the client end, ensuring replay and MITM attacks[11] are possible. This means that this protocol provides no extra security above a standard login and password protocol with no HPA.
3. The human's password, sent in message 5 of the Gajek et al. protocol (see Section 3.1), does not form part of the proof of security of the protocol. Again, this is because the security model used excluded the possibility of the non-intended-user using the computer, the authentication of the user's browser is sufficient to authenticate the user.

As stated earlier in sections 1 and 2, the intention is to create a protocol which is transparent to the human that provides the human with assurance that they are communicating with the party they intend to be communicating with. Users should not be expected to accept that the password that the bank has for them, the HPA, is sent to them before they have consciously provided anything to the bank to authenticate themselves. An interesting observation from the Gajek et al. protocol is that, if the HPA is truly being used as the method to authenticate the server, then the certificate, at least at the server's end, is not required. Indeed, if we presume that users are not checking security certificate information as part of their security decision making process (as evidenced by studies in Radke et al. (2010) and Schechter et al. (2007)) this does suggest a shift in perspective of where the certificate should be used. For example, we could ensure that all banks are made aware of certificates and check for certificates, while we cannot ensure that all users are aware of certificates and check for certificates, therefore the suggestion would be to have the certificates at the user's end (to be issued at the same time as the login and password information is issued to the customer by the bank). This may involve an addition to the TLS protocol, or can be constructed using current traditional server-authentication methods by moving the server's role to the client, such that the client, who has the certificate, becomes the "server", with one extra message

[11] A *man-in-the-middle* (MITM) attack is an attack where a third party intercepts messages between two communicating parties, typically without either intended party detecting this, allowing the MITM attacker to listen in and manipulate messages.

flow in the initialisation of TLS. We shall call this mode which has a certificate at what is traditionally considered the client end, and no certificate at what is traditionally considered the server end, *certificateless-server mode*.

We also seek to create a protocol which is rushing user resistant, that ensures the user is actually checking the authentication provided by the server, which is not enforced in the Gajek et al. protocol. Another lesson learned by the investigation of the Gajek et al. protocol is the ideal that the human, rather than the server, takes the initial steps in the protocol.

A final point on the Gajek et al. protocol is more of a philosophical ideal. The ideal is that the human does not know that *this is the real protocol*. That is, the human is presented with a website and simply follows the instructions - there is no necessity to pre-learn this protocol. This raises an interesting issue, and that is that the adversary can, instead of attacking the Gajek et al. protocol directly, simply create a new website with a new protocol (which does not even have to be related to the real protocol). Therefore, ideally a protocol is created which the human is forced to initiate, and as the initiator, rather than the follower, the human needs to learn what the protocol should look like. This ideal, combined with transparency of the authentication process, should make the protocol *substitution resistant*.

3.2 Usability Testing of Human Protocols

Kumar et al. researched 13 different wireless device pairing methods (Kumar et al. 2009). While the study used now outdated mobile telephones, several lessons learned about how humans take part in protocols can be directly applied. Wireless device pairing typically has unavoidable human involvement, and hence each method which typically combines cryptographic elements with human interaction, is a separate non-trivial security ceremony involving a human. Kumar et al. designed and executed a human study which, over the course of three sessions, compared more than 40 (total) variations of the pairing methods. The study's participants were timed, had their actions logged, completed pre- and post- questionnaires, and an interview. The security ceremonies were assessed for robustness and usability. Robustness results were categorised into *safe errors*[12] and *fatal errors*[13]. Usability was assessed in three categories, being: completion time; successful completion; and user's perception of ease-of-use and personal preference (Kumar et al. 2009).

Two outcomes of Kumar et al.'s research that are directly applicable to human protocol design in general. Firstly, protocols should be designed to be rushing user resistant, by ensuring that a user's responses depend on prior steps. The user cannot just "accept", or, even worse, ignore. Secondly, the human initiating the protocol is important. There were other useful conclusions, regarding what sorts of activities generate the least number of false positives and false negatives, but they are more protocol specific (Kumar et al. 2009).

[12] A *safe error* is any non-fatal error, typically a false positive, a rejection of a successful pairing.

[13] A *fatal error* is a false negative, or the acceptance of a failed pairing instance (as defined in (Uzun et al. 2007)).

4 Towards A Human-Computer Mutual Authentication Protocol, Provably Secure in POPS+

We target two central improvements and considerations which should be included in authentication protocols involving a human. They include

1. Rushing user resistance
2. A security proof at a level above the cryptographic level

4.1 Rushing user resistance

Mutual authentication, for example where a bank authenticates itself to its account holder, as well as the account holder authenticating themselves to the bank, is important. In most protocols where an entity is authenticated to a human, there will be a step similar to the *recognise* function of the Gajek et al. protocol proof (Gajek et al. 2008). In this step, the entity will show *something* (a HPA) to the human, and the human is meant to examine this HPA and if it is correct they proceed, and if the HPA is incorrect they should abort the protocol run. Unfortunately, as we have shown, both in the research literature and in commercial implementations, quite often there is no assurance that the human has completed the *recognise* assessment - a human who skips such a step is called a *rushing user*.

To increase the chances of humans completing the recognise step, rushing user resistance should be included in the protocol. A construction that could be added to most such protocols is to send the human user not just the real HPA (HPA_1), but also a false HPA (HPA_2) in random order. Now, beyond sending to the server their user name and password, the human must also select which of the two HPAs was their HPA. If the human selects the wrong HPA, then the server must abort the protocol even if the login and password the human provides are correct.

There are a number of intricacies with this solution, especially when trying to combine the cryptographic elements with the human elements:

- This solution does not enhance the cryptographic security of the protocol. Rather, this step is only in place to ensure that the human follows the protocol. This element is not captured in current computational-based security proofs and models.
- Beyond not enhancing the cryptographic security, this action decreases the cryptographic security in that the adversary now has twice as many chances of sending the human a legitimate HPA (if only two HPAs are sent to the human) since two HPAs are now sent to the human.
- Whether the human is completing the *recognise* step is being checked by the server, in that if the wrong HPA is selected then the server should abort the protocol and force the human to start again. If the server is the adversary, then the adversary will accept the username and password regardless of which HPA the human chooses. So this *training of the human to follow the protocol correctly* will only work while legitimate protocol runs occur with the real server.
- The improvement to the human's behaviour in following the protocol will happen over time. This is another concept not captured in current security proofs and models.

4.2 Security proofs at a level above the cryptographic level

This paper has introduced the concept of POPS+, being that a *security ceremony* is, at the level that most security professionals consider security, simply a protocol which includes a human. In the same way that practice oriented provable security (POPS) of block ciphers is not proven by examining a protocol including a block cipher, the POPS+ security of a higher level cryptographic building block such as HTTPS should not be proven by examining a protocol which includes HTTPS. The proof of HTTPS is completed elsewhere, and, once proven secure, the super-protocol which uses HTTPS is proven secure. In this way we have moved beyond ceremonies being protocols in their *context of use* to being protocols which include lower level protocols.

To allow this analysis of suitability of a protocol for human use to happen, *ideal* instances of the cryptographic building blocks can be used. For example, an ideal secure channel providing confidentiality, integrity, and authentication for the participant with the private key where the other participant is known to check the certificate, would be used for a HTTPS secured channel. Cryptographers would argue that if the communication channel is secure then the protocol becomes trivial. However, a secure channel is no guarantee that the correct information is being passed to and from the human, which is the focus of this level of analysis. By assuming that cryptographic building blocks, such as the channel, are secure, greater attention can be focused on the protocol flows that interact with the human allowing for quicker and easier ceremony design and analysis.

5 Conclusion

We have drawn inspiration from a variety of sources, including the provable security cryptographic community, network security community, Human Computer Interface (HCI) design community, wireless communication device pairing, and the sociotechnical community, to create generic enhancements which can be applied to human usable human-computer mutual authentication protocols. Human usable protocols were found to require *rushing user* resistance, achieved by ensuring subsequent protocol steps depended on previous protocol steps, and *spoofing resistance*, achieved by ensuring transparency of the protocol to the human, necessitating the protocol be taught/learned, and ensuring that the human initiates the protocol. These aspects were shown to be missing in research literature and current commercial implementations.

This paper suggests a shift in thinking regarding ceremony analysis. Previously, ceremony analysis has been regarded by some as a more complete version of protocol analysis which explicitly includes human interaction, setup steps and OOB communication, thus proving a ceremony secure is proving the protocol secure. Recent work has highlighted an issue with this, regarding each ceremony as a protocol in its *context of use*, meaning that proving a protocol secure in one ceremony does not prove the protocol secure in any other ceremony. This paper takes that a step further, treating the underlying protocol as a cryptographic primitive or building block, and considering the ceremony as a protocol which uses that building block (protocol, such as TLS).

We have highlighted that cryptographic building blocks, such as TLS, have become mature to the point where a further level of abstraction is possible from the level that was applied when practice-oriented provable security (POPS) was promoted by Bellare and Rogaway 18 years ago. This allows, for the security proof of security ceremonies that include humans, to abstract away the cryptographic building blocks and extend the security proofs into the human-computer interface. We have called this paradigm shift POPS+. The philosophy remains the same, and that is, a reductionist proof such that the way to break the protocol is to break the cryptographic building block, and as long as the building block remains secure, the protocol remains secure.

Acknowledgments

The authors acknowledge and appreciate the discussions with Mark Manulis, Douglas Stebila, the Information Security Group at Royal Holloway University of London, and Chai Wen Chuah concerning real-world protocols. The anonymous reviewers were also very helpful, not just for this paper but for the future.

References

Arumugam, G. & Sujatha, R. (2010), 'Secured authentication protocol system using images', *IJCSIS* **8**(8).

Bellare, M. (1999), 'Practice-oriented provable-security', *Lectures on Data Security* pp. 1–15.

Bellare, M. & Rogaway, P. (1993*a*), Entity Authentication and Key Distribution, *in* D. R. Stinson, ed., 'CRYPTO', Vol. 773 of *LNCS*, Springer, pp. 232–249.

Bellare, M. & Rogaway, P. (1993*b*), Random oracles are practical: A paradigm for designing efficient protocols, *in* 'CCS', ACM, pp. 62–73.

Blum, M. & Micali, S. (1984), 'How to generate cryptographically strong sequences of pseudo-random bits', *SIAM J. Comput.* **13**(4), 850–864.

Brainard, J. G., Juels, A., Rivest, R. L., Szydlo, M. & Yung, M. (2006), Fourth-factor authentication: somebody you know, *in* 'CCS', ACM, pp. 168–178.

Bringer, J., Chabanne, H. & Dottax, E. (2006), HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks, *in* 'SecPerU', IEEE Computer Society.

Chiasson, S., van Oorschot, P. & Biddle, R. (2007), Even experts deserve usable security: Design guidelines for security management systems, *in* 'SOUPS Workshop on Usable IT Security Management (USM)', Citeseer, pp. 1–4.

Daher, W. & Canetti, R. (2008), POSH: A Generalized CAPTCHA with Security Applications, *in* D. Balfanz & J. Staddon, eds, 'AISec '08', ACM, pp. 1–10.

Dhamija, R., Tygar, J. & Hearst, M. (2006), Why phishing works, *in* 'Proceedings of the SIGCHI conference on Human Factors in computing systems', ACM, p. 590.

Dourish, P. (2004), 'What we talk about when we talk about context', *Personal and ubiquitous computing* **8**(1), 19–30.

Dziembowski, S. (2011), 'How to pair with a human', *Security and Cryptography for Networks* pp. 200–218.

Ellison, C. (2007), 'Ceremony Design and Analysis', Cryptology ePrint Archive, Report 2007/399. `http://eprint.iacr.org/`.

Ellison, C. & Dohrmann, S. (2003), 'Public-key support for group collaboration', *ACM Trans. Inf. Syst. Secur.* **6**(4), 547–565.

Gajek, S., Manulis, M., Sadeghi, A.-R. & Schwenk, J. (2008), Provably Secure Browser-Based User-Aware Mutual Authentication over TLS, *in* M. Abe & V. D. Gligor, eds, 'ASIACCS', ACM, pp. 300–311.

Goldwasser, S. & Micali, S. (1984), 'Probabilistic encryption', *J. Comput. Syst. Sci.* **28**(2), 270–299.

Goldwasser, S., Micali, S. & Rivest, R. L. (1988), 'A digital signature scheme secure against adaptive chosen-message attacks', *SIAM J. Comput.* **17**(2), 281–308.

Hammouri, G. & Sunar, B. (2008), PUF-HB: A Tamper-Resilient HB Based Authentication Protocol, *in* 'ACNS', Vol. 5037 of *Lecture Notes in Computer Science*, pp. 346–365.

Harrison, S., Tatar, D. & Sengers, P. (2007), The three paradigms of HCI, *in* 'Alt. Chi. Session at the SIGCHI Conference on Human Factors in Computing Systems San Jose, California, USA', Citeseer.

Hopper, N. J. & Blum, M. (2001), Secure Human Identification Protocols, *in* C. Boyd, ed., 'ASIACRYPT', Vol. 2248, Springer, pp. 52–66.

ISO/IEC 27001 Information technology - Security techniques - Information security management systems - Requirements (2005), ISO.

Juels, A. & Weis, S. A. (2005), Authenticating pervasive devices with human protocols, *in* 'CRYPTO', Vol. 3621 of *Lecture Notes in Computer Science*, Springer, pp. 293–308.

Karlof, C., Tygar, J. D. & Wagner, D. (2009), 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.

Kumar, A., Saxena, N., Tsudik, G. & Uzun, E. (2009), 'A comparative study of secure device pairing methods', *Pervasive and Mobile Computing* **5**(6), 734–749.

Martina, J. & Carlos, M. (2008), 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'.

Martina, J. E., de Souza, T. C. S. & Custodio, R. F. (2009), Ceremonies Formal Analysis in PKI's Context, *in* 'CSE '09: Proceedings of the 2009 International Conference on Computational Science and Engineering', IEEE Computer Society, Washington, DC, USA, pp. 392–398.

Menezes, A., van Oorschot, P. C. & Vanstone, S. A. (1996), *Handbook of Applied Cryptography*, CRC Press.

Mollin, R. (2005), *Codes: The Guide to Secrecy from Ancient to Modern Times*, Chapman & Hall/CRC Press.

Norman, D. (2002), *The design of everyday things*, Basic Books New York.

Oorschot, P. & Wan, T. (2009), 'Twostep: An authentication method combining text and graphical passwords', *E-Technologies: Innovation in an Open World* pp. 233–239.

Radke, K., Boyd, C., Brereton, M. & Nieto, J. G. (2010), How HCI Design Influences Web Security Decisions, *in* 'OzCHI', ACM.

Radke, K., Boyd, C., Nieto, J. G. & Brereton, M. (2011), Ceremony analysis: Strengths and weaknesses, *in* 'IFIP SEC'11', LNCS (to appear), Springer.

Schechter, S., Dhamija, R., Ozment, A. & Fischer, I. (2007), Emperor's new security indicators: An evaluation of website authentication and the effect of role playing on usability studies, *in* 'In Proceedings of the 2007 IEEE Symposium on Security and Privacy', Citeseer.

Shostack, A. & Stewart, A. (2008), *The New School of Information Security*, Addison-Wesley Professional, Upper Saddle River, N.J.

Simon, H. (1969), *The science of the artificial*, MIT press.

Simon, H. (1996), *The Sciences of the Artificial. 3rd ed.*, The MIT Press, Cambridge, Massachusetts.

Smith, S. (2003), 'Humans in the loop: Human-computer interaction and security', *Security & Privacy, IEEE* **1**(3), 75–79.

Suchman, L. (2007), *Human-machine reconfigurations: Plans and situated actions*, Cambridge Univ Pr.

Uzun, E., Karvonen, K. & Asokan, N. (2007), 'Usability analysis of secure pairing methods', *Financial Cryptography and Data Security* pp. 307–324.

Yao, A. C.-C. (1982), Theory and applications of trapdoor functions (extended abstract), *in* 'FOCS', IEEE, pp. 80–91.

Yee, K. (2004), 'Aligning security and usability', *Security & Privacy, IEEE* **2**(5), 48–55.