A Proposed Framework for Analysing Security Ceremonies

Marcelo Carlomagno Carlos[1*], Jean Everson Martina[2†], Geraint Price[1] and Ricardo Felipe Custódio[2]
[1]*Royal Holloway University of London, Information Security Group, Egham, Surrey, TW20 0EX, United Kingdom*
[2]*Universidade Federal de Santa Catarina, Laboratório de Segurança em Computação, Florianópolis - SC - Brazil*
{*marcelo.carlos.2009, geraint.price*}*@rhul.ac.uk,* {*everson, custodio*}*@inf.ufsc.br*

Keywords: Security Ceremonies, Security Protocols, Formal Methods, Cognitive Human Formalisation

Abstract: The concept of a ceremony as an extension of network and security protocols was introduced by Ellison. There are no currently available methods or tools to check correctness of the properties in such ceremonies. The potential application for security ceremonies are vast and fill gaps left by strong assumptions in security protocols. Assumptions include the provision of cryptographic keys and correct human interaction. Moreover, no tools are available to check how knowledge is distributed among human peers nor their interaction with other humans and computers in these scenarios. The key component of this position paper is the formalisation of human knowledge distribution in security ceremonies. By properly enlisting human expectations and interactions in security protocols, we can minimise the ill-described assumptions we usually see failing. Taking such issues into account when designing or verifying protocols can help us to better understand where protocols are more prone to break due to human constraints.

1 INTRODUCTION

Protocols have been analysed since Needham and Schroeder (Needham and Schroeder, 1978) first introduced the idea and methods have been researched to prove protocols' claims. We have seen a lot of research in this field. Particularly in developing formal methods and logics to check and verify such claims. We must cite Burrows et al. (Burrows et al., 1989) for their belief logic, Abadi for spi-calculus (Abadi and Gordon, 1997), Ryan (Ryan and Schneider, 2000), Lowe (Lowe, 1996) and Meadows (Meadows, 1996) for works on state enumeration and model checking, and Paulson and Bella (Paulson, 1998; Bella, 2007) for their inductive method as the principal initiatives. We have also seen the creation of a number of tools to verify and check security protocols automatically. These techniques and tools have evolved in such a way that, nowadays, we can check and analyse complex and extensive protocols.

Meadows (Meadows, 2003) and Bella et al. (Bella et al., 2003) in their area survey gave us a broad coverage of the maturity in this field of protocol verification. They also point to trends followed by methods, pinpointing their strong and weak features. They give propositions for research ranging from open-ended protocols, composability and new threat models; something that has changed very little since Dolev and Yao's proposal (Dolev and Yao, 1983). These problems seem very well covered. Current research is, in general, aimed at optimising the actual methods in speed and coverage. Extending protocol verification and description to include fine-grained assumptions and derivations is an unexplored research path.

Nevertheless, recent research (Dhamija et al., 2006; Gajek, 2005; Jakobsson, 2007) shows that even the most deployed, tested and analysed protocols can have security problems. This usually happens when a user acts in an unexpected, but plausible way. Since protocols operate at computer level, we tend to verify them for computer interaction.However, they are built to accomplish a human task and thus we should design and verify protocols (in this case ceremonies) against human interaction. We should take into account human processes when designing computer security protocols. Corroborating the idea that the verification of security protocols should include environmental assumptions, Bella et al. (Bella et al., 2003) state that "it is unwise to claim that a protocol is verified unless the environmental assumptions are clearly specified. Even then, we can be sure that somebody will publish an attack against this protocol".

Ceremonies and their analysis were introduced by Ellison (Ellison, 2007). He states that "ceremonies

*Supported by CNPq/Brazil
†Supported by CNPq/FINEP Brazil

extend the concept of protocols by also including human beings, user interfaces, key provisioning and all instances of the workflow". This idea can give a broader coverage of the protocols' point of view, extending what can be analysed and verified by protocol techniques. Ellison gives an overview and establishes the basic building blocks for ceremony description. Carlos and Price (Carlos and Price, 2012) further analysed the human-protocol interaction problems, proposing a taxonomy of overlooked components in this interaction and elaborating a set of design recommendations for security ceremonies. Although Ellison proposes the possibility of using formal methods for security protocol analysis, no major work is found today in the ceremony formal-analysis field. This creates a weak spot, and leads to empirical analysis, which can be difficult and error-prone, as the history of protocol analysis shows us.

An important advance in the reasoning about ceremonies was introduced by Rukšėnas et al. (Ruksenas et al., 2008). They developed a human error cognitive model, initially applied to interaction on interfaces. They show that, normally, security leaks come from mistakes made when modelling interfaces, not taking into account the cognitive processes and expectations of human beings behind the computer screen. They successfully verify problems on an authentication interface and a cash-point interface. They showed that the normal lack of consideration in the human peers cognitive processes is one of the weakest factors in these systems. Their proposal comes with a powerful implementation using a model-checker.

Our approach is different. We do not focus on a specifically difficult to describe limitation of human beings, but on giving to the protocol and ceremony designers a better way to define human expectation and interaction. Thus, by making the assumptions more explicit, and requiring a description of the ceremony's security, we can enable designers to experiment with different ceremony techniques. By stating fine-grained assumptions and analysing their absence, we can get insights of potential break points for security ceremonies. This is the conceptual extension we are proposing in verifying security ceremonies using established techniques based on formal method's .

To try to achieve this complex task of verifying security ceremonies we need to first understand the major differences and features of ceremonies when compared to security protocols (Section 2). We briefly discuss a real world example on Section 3. Then we describe our proposal for the formalisation of human knowledge distribution in security ceremonies in Section 4. The future direction of our work and our next steps are presented in Section 5. Finally, we conclude with some thoughts on what is achievable and the limitations we are likely to encounter.

2 CEREMONY ANALYSIS VERSUS PROTOCOL ANALYSIS

Security ceremonies are a superset of security protocols. They can be seen as an extension of security protocols, including additional node types, communication channels and operations which were previously considered out-of-bound. These operations are normally assumptions we make when trying to check or analyse claims for protocols. They include a safe key distribution scheme for symmetric key protocols; the confidence we must have that the computer executing the protocol is trusted; and whether users will behave as expected or not, among other things. We usually make these assumptions but we rarely do explicitly describe them.

The inclusion of human interaction and, consequently, behaviour and cognitive processes, is a characteristic of ceremonies as human peers are out of bounds for protocol verification. They are normally the most error prone peer in any process, and their inclusion can enrich the details and coverage of any analysis done so far.

Protocol descriptions tend to be easier to transcribe as mathematical notations due to the intrinsic computational characteristics present in them. Much of this comes from them being targeted to computers. Ceremony modelling is a much more subtle approach, since the possibilities involved in modelling human behaviour are immense. However, by adding new components to the specification, such as new node types (humans) and communication mediums (user interfaces, speech, etc), we will be able to describe assumptions related to these components in a more precise manner. Consequently, a more detailed analysis of the ceremony's security properties will be possible.

3 AN EXAMPLE CEREMONY

SSL/TLS are a set of cryptographic protocols that provide privacy and data integrity for communication over networks. A practical application of these protocols can be seen when we connect to websites and a padlock appears in the browser window. The padlock indicates to the user that the connection between client and server is encrypted and the server is authen-

ticated to the client (the clients can also be authenticated to the server, but this is an optional feature).

These protocols are widely used and are also the object of many studies and analysis (Paulson, 1999; Mitchell et al., 1998). The results of those studies show that the SSL/TLS protocols are well designed and secure. Additionally, these protocols are designed to prevent man-in-the-middle attacks (MITM). However, there are specific situations where we can deploy a MITM attack by exploiting assumptions which can be difficult to achieve. We chose as an example the assumption that users are capable of making an accurate decision on whether to accept a certificate or not.

Current analysis of the protocol assumes there is a trusted Certification Authority (CA) and all parties involved possess the CA's public key. Nevertheless, this assumption does not cover some real world scenarios. When there is no valid certification path between the server's certificate and the client's trusted CAs, most implementations allow a dynamic (real-time) acceptance and addition of new trusted certificates. In this case, the initial assumption is weakened, and the verification is less comprehensive.

The dynamic acceptance of new certificates (and consequently new servers' public keys) is currently not analysed. This happens because these messages are sent through another medium, the computer-human medium, which is out of the scope of protocol analysis. This leaves such implementations susceptible to failure, weakening the achievability of the protocol's goals due to the weakening of the assumptions. Additionally, implementations such as those we find in web browsers force users to authenticate digital objects (Certificates) which, according to some research findings, is not feasible (Carlos and Price, 2012).

We have seen some attempts to include specific human interaction into protocol specification. Gajek et al. (Gajek et al.,) developed a protocol that includes a human node in the specification. This simplified approach is the first attempt that we are aware of to formally verify protocols including human interactions.

By modelling ceremony analysis using formal methods, we will be able to break broad assumptions such as those we use in SSL/TLS analysis, into smaller and more plausible assumptions. Consequently, we will allow designers to have better insights of the protocols' weak spots, such as those we have discussed. In the future, with further development of the ceremony analysis research field, we will be able to model even more complex aspects, such as composability of security protocols and consequently security ceremonies.

4 A PROPOSED METHOD

In traditional protocol specifications we have one communication medium. In ceremonies we include humans into the specification. Consequently we have to define two new communication mediums, one to represent human-computer interaction (user interfaces) and another to symbolise human-human interaction. Figure 1 gives an overview of the communication mediums involved. The area bounded by the dotted line represents the traditional protocol point of view, while the complete figure represents a ceremony point of view.

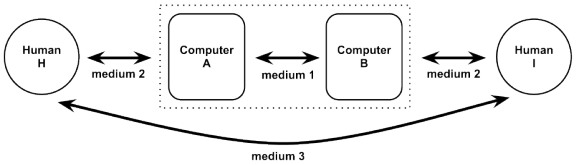

Figure 1: Ceremony communication mediums

As we see, in a protocol specification, the human-protocol and human-human interaction are assumed to happen out-of-band, and become part of the design assumptions. When implemented, the assumptions are replaced by dynamic user-interactions. When these assumptions are too strong, it becomes difficult to implement a protocol providing the expected security properties (Carlos and Price, 2012). By adding new components to the specification, such as users and different communication mediums, we can start to describe these assumptions in the ceremony, and consequently perform a more detailed analysis of them and their impact on the ceremony's security properties. Lack of precise description of assumptions is a weak spot in protocol design.

We decided to implement our description and verification model for security ceremonies based on Paulson's Inductive Method (Paulson, 1998). Paulson introduced the inductive method of protocol verification where protocols are formalised in typed high-order logic as an inductive defined set of all possible execution traces. An execution trace is a list of all possible events in a given protocol. Events can be described as the action of sending or receiving messages, as well as off-protocol gathered knowledge. The attacker is specified following *Dolev-Yao's* propositions. The attacker has his knowledge derived and extended by two operators called *synth* and *analz*. Operator *analz* represents all the individual terms that the attacker is able to learn using his capabilities defined by the threat model within the protocol model, and *synth* represents all the messages he can compose with the set of knowledge he possesses.

Protocols are defined as inductive sets constructed over abstract definitions of the network (computer-to-computer media) and cryptographic primitives. Proofs about a protocol's properties are written as lemmas. Lemmas are constructed taking the properties we desire to achieve within the set of all admissible traces, and are typically proven inductively. This framework is built over induction, which makes the model and all its verifications potentially infinite, giving us a broad coverage and flexibility. This approach has already been used to prove a series of classical protocols (Paulson, 1998) as well as some well-known industry grade protocols, such as the SET online payment protocol, Kerberos and SSL/TLS (Bella et al., 2002; Bella, 2007).

Based on the current model for protocols set by Paulson (Paulson, 1998) and extended by Bella (Bella, 2007), we include a new agent type called *Human*. We also add a set of operators, messages and events to set up our human peer and enable it to work under human capabilities and constraints. This new agent type is capable of storing knowledge and sending messages over the mediums it is capable of operating. This agent is also capable of using knowledge conversion functions to be able to operate its devices. Humans are related to devices they operate or own, and some of the physical constraints existent in the real world are also present in this relation.

Our specification of the Human peer is similar to the one for a computer Agent. It is created as shown in Definition 1 to enable the type constriction provided in the inductive method implemented in Isabelle/HOL.

Definition 1. *Human datatype definition*

```
datatype
  human = Friend nat
```

To enable the representation of the different mediums described above we extended the datatype *Event* as shown in Definition 2. To represent the protocol side, the events were kept unchanged and are composed by *Says*, *Gets* and *Notes*. To represent the new human-computer medium we have the events *Displays* and *Inputs* which takes an agent, a human and a message, and a human, an agent and a message respectively. Representing the human-human medium, we have the events *Tells*, *Hears* and *Keeps*, which are similar to the protocol events, but now we take a human instead of an agent as the parameter. This construction is made to allow us to control the flow of information passed between devices and humans, as well as between human peers.

Definition 2. *Event datatype definition*

```
datatype
  event = Says agent agent msg
    | Gets agent msg
    | Notes agent msg
    | Displays agent human msg
    | Inputs human agent msg
    | Tells human human msg
    | Hears human msg
    | Keeps human msg
```

After modelling the human agent and its messages, we adapted the two functions that deal with knowledge distribution and control the freshness of components appearing in different protocol runs. The function that deals with knowledge distribution is called *knows* and its extended description is shown in Definition 3. One peculiarity of our implementation so far is that the function *knows* deals only with the computer media knowledge flow and the cross-medium flow in the agent direction. We still lack a function to distribute knowledge in the human media flow and in the cross-medium flow in the human direction. The function for freshness is called *used* and is very similar to *knows* in construction. It is not shown here due to space constraints.

Definition 3. *Event datatype definition*

```
primrec knows::"agent=>event list=>msg
set"
 where
  knows_Nil:"knows A []= initState A"
  | knows_Cons:
   "knows A (ev#evs) = (if A = Spy then
   (case ev of
    Says A'B X=>insert X(knows Spy evs)
    |Displays A' H X => if A' ∈ bad
then insert X (knows Spy evs) else
knows Spy evs
    |Inputs H A' X => if A' ∈ bad then
insert X (knows Spy evs) else knows Spy
evs
    |Gets A' X => knows Spy evs
    |Tells C D X => knows Spy evs
    |Hears C X => knows Spy evs
    |Keeps C X => knows Spy evs
    |Notes A' X => if A'∈ bad then
insert X (knows Spy evs) else knows Spy
evs)
   else (case ev of
    Says A'B X => if A'=A then insert X
(knows A evs) else knows A evs
    |Displays A' H X => if A'=A then
insert X (knows A evs) else knows A evs
    |Inputs H A' X => if A'=A then
insert X (knows A evs) else knows A evs
    |Tells C D X => knows A evs
    |Hears C X => knows A evs
    |Keeps C X => knows A evs
    |Gets A' X => if A'=A then insert X
(knows A evs) else knows A evs
    | Notes A' X => if A'=A then insert
X (knows A evs) else knows A evs))"
```

We also define a new set of functions to represent the information that flow,s and is processed through, the two new communication mediums. For the human-computer medium, we created three functions called *Reads*, *Recognises* and *Writes*. *Reads* represents a human reading a message (e.g. a text displayed on the screen) and this adds information to the referred human knowledge. *Recognises* accounts a human recognising something he/she already knows and is being presented to him/her. In other words, something that the human is reading and matches something previously known. Finally, *Writes*, as shown in Definition 4 is equivalent to Paulson's *Synth*, where we enable humans to combine their knowledge in a monotonic way to create new possible inputs. For now we do not consider inherent human constrains.

Definition 4. *Writes function definition*

```
inductive_set writes::  "msg set=>msg set"
for H ::  "msg set" where
 Inj [intro]:"X ∈ H ==> X  writes H"
 |Agent [intro]:"Agent agt ∈ writes H"
 |Human [intro]:"Human hum ∈ writes H"
 |Secret [intro]:"Secret n ∈ writes H"
 |Number [intro]:"Number n ∈ writes H"
 |MPair [intro]:"[|X ∈ writes H; Y ∈
writes H|] ==> |X,Y| ∈ writes H"
```

In addition to the functions described above two new events, *Displays* and *Inputs*, are available. *Displays* represents a computer displaying a message to a human (e.g., via user interface). *Inputs* describes the event of a human sending a command or data to the computer (e.g., typing text in a text box). This gives us an abstract representation of the complex human-computer medium. In this cross-medium space we believe more details can be plugged to describe inherent factors of human-computer interaction.

For the human-human medium, we created three functions called *Listens*, *Understands* and *Voices*. *Listens* represents a human listening to a message sent from another human (e.g., one human listening to another). *Understands* accounts for a human understanding something that has been said and matches with previously known information. This information can be something gathered beforehand, creating a paradox in the ceremony concept, or acquired during a previous or current run of the ceremony. To conclude, *Voices* is the equivalent to a human saying something to another human passing its knowledge via the human-human medium. These constructions enable us to explore a different threat model for the human media as we will briefly discuss in Section 5.

Finally, we define three new events for human-human interaction as said above: *Tells*, *Hears* and *Keeps*. A human sending a message to another human is represented by the event *Tells*. *Hears* is the complementary event, describing a human receiving a message from another. And *Keeps* is equivalent to the *Notes* event, already existent for protocols. From *Notes* we may receive something out-of-band of the protocol, and consequently, from *Keeps* something out-of-band to the ceremony may be received. This construction is present because we believe there is always a limit on what can be described.

With the infrastructure described above implemented, we already have a partially working framework. Together with the definitions we have proven a series of lemmas required by Isabelle/HOL. These lemmas are required for reasoning about the definitions. We have already proven more than 80 technical lemmas regarding monotonicity, idempotence, transitivity, and set operations for the inductive sets and recursive definitions we specified. We have also proven lemmas regarding the relation of the functions we introduced with those already existent in the method.

5 FUTURE WORK

Protocols are, by design, implemented to attend to human demands. The method we propose approaches real world concerns on the design level. It is impossible to represent all possible human characteristics in a limited set of operations, but by including the human node in the specification, we can thoroughly study interactions and factors which were previously included in the set of assumptions for each protocol.

Our next step is to verify simple ceremonies using our model and then further develop and refine the model. In addition to that, we will apply the proposed model to check whether a specified ceremony overlooks human-protocol interaction components, such as those described in (Carlos and Price, 2012). We also plan to use the model to verify whether some of the design recommendations proposed by Carlos and Price (Carlos and Price, 2012) (e.g. the use of forcing functions to prevent inappropriate user interaction) are correctly implemented or not.

The contextual coverage that ceremonies bring to security protocols is another property is worth verifying. This can give us better insights into the problem of protocol composability. The composability problem normally happens because of clashes among environmental assumptions that are embedded into protocols. By not being able to model the environment, we also cannot predict what will happen when two protocols, that are designed focusing on their own respective environments are put to work together.

Another point worth mentioning in the ceremony verification area is the lack of a tailored threat model.

We need a model that encompasses active threats, as we have in protocols, as well as passive threats such as unreliable behaviour and memory. We can use some work from Roscoe (Roscoe et al., 2003) that talks about human centric security as a basis. However our initial experiments already show that the threat model described by Dolev and Yao is not realistic for our human-to-human interaction media. The presence of an omnipotent and omnipresent being in human interactions is highly debatable.

6 FINAL CONSIDERATIONS

The idea of modelling ceremonies and applying formal methods to them seems promising. The knowledge acquired by the protocol analysis community can be used to boost the ceremony analysis area. Such analysis can help us detect scenarios where protocols are more prone to failure. By better understanding these issues we will be able to design more user centric protocols which are less-likely to fail.

We don't want to change the way we analyse protocols today, since the formal methods available are mature and powerful for their intended purposes. We want to approach the problem from an extended point of view. Our focus on using a mature and powerful method, such as Paulson's inductive method, is reasonable. Our objective with this model is to extend the coverage from the verification of security protocols to ceremonies. Human behaviour is indeed unpredictable, but by including humans in the formal models we can, at least, begin to detect some previously undetectable flaws due to human interaction.

REFERENCES

Abadi, M. and Gordon, A. D. (1997). Reasoning about cryptographic protocols in the spi calculus. In *Proc. of the 8th Int. Conf. on Concurrency Theory*, pages 59–73. Springer-Verlag.

Bella, G. (2007). *Formal Correctness of Security Protocols*, volume XX of *Information Security and Cryptography*. Springer Verlag.

Bella, G., Longo, C., and Paulson, L. C. (2003). Is the verification problem for cryptographic protocols solved? In *Security Protocols Works.*, volume 3364 of *LNCS*, pages 183–189. Springer.

Bella, G., Massacci, F., and Paulson, L. C. (2002). The verification of an industrial payment protocol: the SET purchase phase. In *Proc. of the 9th ACM CCS*, pages 12–20, Washington, DC, USA. ACM Press.

Burrows, M., Abadi, M., and Needham, R. (1989). A logic of authentication. In *Proc. 12th ACM Symposium on Operating Systems Principles*, Litchfield Park, AZ.

Carlos, M. C. and Price, G. (2012). Understanding the weaknesses of human-protocol interaction. In *Works. on Usable Security at 16th Int. Conference on Financial Cryptography and Data Security*.

Dhamija, R., Tygar, J. D., and Hearst, M. (2006). Why phishing works. In *Proc. of the SIGCHI conference on Human Factors in computing systems*, CHI '06, pages 581–590, New York, NY, USA. ACM.

Dolev, D. and Yao, A. (1983). On the security of public key protocols. *Information Theory, IEEE Transactions on*, 29(2):198–208.

Ellison, C. (2007). Ceremony design and analysis. Cryptology ePrint Archive, Report 2007/399. http://eprint.iacr.org/.

Gajek, S. (2005). Effective protection against phishing and web spoofing. In *Proc. of the 9th IFIP Conf. on Comm. and Multimedia Sec., LNCS 3677*, pages 32–41.

Gajek, S., Manulis, M., Sadeghi, A.-R., and Schwenk, J. Provably secure browser-based user-aware mutual authentication over tls. In *Proc. of the 2008 ACM symposium on Information, computer and communications security*.

Jakobsson, M. (2007). The human factor in phishing. In *In Privacy & Security of Consumer Information '07*.

Lowe, G. (1996). Breaking and fixing the needham-schroeder public-key protocol using fdr. In *Proc. of the 2nd Int. Works. on Tools and Algorithms for Construction and Analysis of Systems*, pages 147–166.

Meadows, C. (1996). Language generation and verification in the nrl protocol analyzer. In *Proc. of the 9th IEEE CSF*, page 48, Washington, DC. IEEE Comp. Soc.

Meadows, C. (2003). Formal methods for cryptographic protocol analysis: Emerging issues and trends. *IEEE Journal on Selected Areas in Communications*, 21.

Mitchell, J. C., Shmatikov, V., and Stern, U. (1998). Finite-state analysis of SSL 3.0. In *Proc. of the 7th conference on USENIX Security Symposium*, volume 7, page 16, San Antonio, Texas. USENIX.

Needham, R. M. and Schroeder, M. D. (1978). Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12):993–999.

Paulson, L. C. (1998). The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6(1-2):85–128.

Paulson, L. C. (1999). Inductive analysis of the Internet protocol TLS. *ACM Transactions on Information and System Security*, 2(3):332–351.

Roscoe, A. W., Goldsmith, M., Creese, S. J., and Zakiuddin, I. (2003). The Attacker in Ubiquitous Computing Environments: Formalising the Threat Model. In *Proc. of 1st Int. Works. on Form. Asp. in Security and Trust*.

Ruksenas, R., Curzon, P., and Blandford, A. (2008). Modelling and analysing cognitive causes of security breaches. *Innovations in Systems and Software Engineering*, 4(2):143–160.

Ryan, P. and Schneider, S. (2000). *The modelling and analysis of security protocols: the csp approach*. Addison-Wesley Professional.