A New Approach to the Inductive Verification of Cryptographic Protocols Based on Message Algebras

Lassaad Cheikhrouhou, Werner Stephan and Markus Ullmann

Introduction. The formal analysis of cryptographic protocols is still an active research area with a variety of different approaches that are sometimes difficult to overlook. Well known tool-supported approaches include

- inductive verification techniques going back to Paulson, [1],
- systems for debugging protocol scenarios, like AVISPA [2], and
- the (dis-)proof of protocol goals by resolution and saturation, like in ProVerif [3].

In this paper we extend the inductive approach to protocols that are based on message algebras. The work was motivated by the conviction that this approach should definitively be pursued further, possibly in combination with the other fully automatic techniques, because of a number of advantages:

- Extensibility: New kinds of features (protocol steps and properties) can be easily added while reusing the basic techniques.
- Traceability: Protocol models (including underlying assumptions) and properties are explicitly given. Verification results in (explicit) proofs based on a small set of (given) rules. This supports a third party assessment like in an official Common Criteria Evaluation.
- There are no technical complications that might lead to failures (to come up with a result).

Moreover the price to be paid for these advantages decreases significantly over the years. Our proof strategy implemented in the Verification Support Environment (VSE) reaches a high degree of automatization so that typically more than 90 % of the proof steps (tactics) are selected automatically by heuristics, [4]. In the re-use of proofs that was applied when the protocol discussed in the paper was changed (several times), automation goes still further. Note that the number of user interactions in a proof depends to a lesser extend on the proof size, but rather on the required lemmata in this proof. So, huge proofs (with more than 1000 proof steps) could be generated automatically when no (new kinds of) lemmata are needed.

The main contribution described in this paper is the extension of this implemented verification method to protocols that are based on message algebras given by a set Op of operations and equations E_{Op}. It allowed us to prove standard and non-standard properties of the Password Authenticated Connection Establishment (PACE) protocol, [5], which is deployed in the new German identity cards.

Message algebras were primarily addressed in the context of search-based analysis methods with a main focus on decidabilty and complexity results, e.g., [7, 8]. Only some of the theoretical approaches were actually implemented in tools. To the best of our knowledge, the work on PACE presented in this paper is the first inductive verification of a cryptographic protocol based on message algebras.

Besides the standard properties of confidentiality and authenticity with some subtle restrictions for the latter we proved a number of properties that the resistance against offline password testing was reduced to (as proof obligations) by an additional proof method. Moreover, we proved that the generator that is established during a protocol run is of a certain form corresponding in case of elliptic

curve cryptography to a scalar product of the initial (base point) generator. This result guarantees that an attacker is not able to introduce cryptographic weaknesses.

As indicated in the paper our results are general enough to provide strong hope that even protocols with much richer algebras (for elliptic curve cryptography) definitively outside the scope of current automatic systems can be treated by the inductive approach.

PACE. PACE is a password-based key agreement protocol with mutual entity authentication. It takes place between the RF-chip (A) of a contactless smart card and a (inspection) terminal (B). Thus, a major objective of PACE is to protect the password of A ($pwd(A)$) from offline dictionary attacks. This heavily influences the protocol design and the choice of cryptographic primitives.

The main idea in the design of PACE is to use the password in the *encryption* (by *enc*) of some random value (nonce) s that is necessary for the computation of the session key. In order to prevent offline dictionary attacks, s is utilized (on B's side after decryption by *dec*) together with a static generator g and Diffie-Hellman (DH) values in the computation of a *fresh* generator with the help of one-way functions *dh* and *gen*. This generator *must* be used together with additional *local* secrets, i.e. the private values in a second DH exchange, for the computation of the session key again by the one-way function *dh*. This way, the obtained session key is bound to the password with the intermediate of s used in the computation of the fresh generator.

1. $A \longrightarrow B \ : \ enc(pwd(A), s) \ \% \ z$
2. $B \longrightarrow A \ : \ dh(g, x_1) \ \% \ X1$
3. $A \longrightarrow B \ : \ dh(g, y_1) \ \% \ Y1$
4. $B \longrightarrow A \ : \ dh(gen(dh(g, dec(pwd(A), z)), dh(Y1, x_1)), x_2) \ \% \ X2$
5. $A \longrightarrow B \ : \ dh(gen(dh(g, s), dh(X1, y_1)), y_2) \ \% \ Y2$
6. $B \longrightarrow A \ : \ mac(dh(Y2, x_2), Y2)$
   $\% \ mac(dh(X2, y_2), dh(gen(dh(g, s), dh(X1, y_1)), y_2))$
7. $A \longrightarrow B \ : \ mac(dh(X2, y_2), X2)$
   $\% \ mac(dh(Y2, x_2), dh(gen(dh(g, dec(pwd(A), z)), dh(Y1, x_1)), x_2))$

We have used two notations for every message (seperated by'%'), to explicitly show the receiver's view by the expression on the right-hand side of '%'. The corresponding expressions in steps 6 and 7 describe how the respective receiver computes the own message authentication codes (MACs by *mac*) to complete the authentication. The receiver's view in steps 1–5 is given by variables to express that messages are accepted without tests.

Verified Properties. We have proved the five main security goals of PACE:

- (standard) confidentiality of session keys (the DH values established in steps 4 and 5),
- mutual authentication, i.e. the respective authenticity guarantees for roles A and B, including the fact that the fresh generator is of the form $gen(dh(g, s), dh(dh(dh(dh(\ldots, dh(g, m_0), \ldots), m_{n-1}), x_1), y_1))$, i.e. the attacker is not able to intersperse a cryptographically weaker fresh generator than the initial generator g,
- forward secrecy of session keys (a successfully guessed password does not reveal the session keys that had been generated in previous runs),
- and the resistance against offline password testing.

Note that we have proved the strongest form of resistance, where the attacker may use (in her test attempts) accidentally disclosed session keys in addition to the exchanged messages. Consequently, PACE provides the highest security level of password protocols, as it was required for deployment in the German identity cards. This also explains why PACE can be optionally used in Machine Readable Travel Documents instead of the Basis Access Control (BAC) Protocol, [6]. BAC does guarantee neither forward secrecy nor resistance against offline password testing.

The regularity lemmata that are sufficient for the resistance property express restrictions on the possible messages in protocol runs. For instance, it is required for *dh* messages that either both arguments are known by the attacker or the second argument is *a local* secret that can not be derived (even with the right password).

Inductive, Algebraic Verification. For message algebras given by Op and E_{Op} we define (as in search-based approaches, [7]) DY reasoning on finite message sets ik by a closure operation $DY(ik)$ that uses a subset $Op_a \subseteq Op$. In case of PACE we have $Op_a = Op$.

$$\begin{aligned} DY(ik) &= \bigcup_{i \in N} DY^i(ik) \\ DY^0(ik) &= ik \\ DY^{i+1}(ik) &= DY^i(ik) \ \cup \{f(m_0, \ldots, m_{n-1}) \mid n \in N, \\ &\qquad f \in Op_a, m_0, \ldots, m_{n-1} \in DY^i(ik)\} \end{aligned}$$

Here, the computations given by the equations in E_{Op} are implicit.

In the inductive approach, ik consists of (immediately) observable messages in event traces.

Trace Models. A model for a given protocol $\mathcal{P}$ consists of a set $Tr[\mathcal{P}]$ of (finite) sequences (traces) $tr = \langle e_0, \ldots, e_{n-1} \rangle$ of (protocol-) *events*. The main events are *send events* $send(a, b, m)$ where a protocol participant a sends a message m to some other participant b.

The set $Tr[\mathcal{P}]$ is defined *inductively* by using predicates (rules) $R(tr, e)$ that describe the conditions for a given trace tr to be extended by a new event e. For *send* events there is a predicate R_i for each line i of the protocol and an additional *fake rule* for the attacker named *spy*. We have $R_f(tr, send(spy, b, m)) \leftrightarrow m \in DY(spies(tr))$. By $spies(tr)$ we denote the set of the immediately observable messages by the attacker.

Properties of protocols $\mathcal{P}$ are given as subsets of $Tr[\mathcal{P}]$. For *secrecy* we *define* $Sec[\mathcal{P}](a, b)$ as the set of all traces $tr \in Tr[\mathcal{P}]$ where the session key established between a and b in tr (if there is any) is not in $DY(spies(tr))$. Verification by induction on the length of traces then shows $Sec[\mathcal{P}](a, b) = Tr[\mathcal{P}]$ for all a, b. To cope with the used equations, we assume (as in search-based approaches) that they are of certain *restricted* forms.

Message Algebras. We assume E_{Op} to be a partition $E_C \uplus E_F$, where

- E_C consists of *cancellation equations* where the right-hand side is a subterm of the left-hand side or a constant,
- and E_F is a set of equations that induce *finite* equivalence classes (wrt. the term algebra).

We call the head-symbols of the left-hand side of the equations in E_C *cancellation operators*.

Note that E_C can be usually oriented from left to right to obtain a *terminating* term rewrite system for the (terms in the) equivalence classes induced by E_F. We are interested in the cases where normal forms (by this term rewrite system) exist. This allows us to obtain *finite* representations of the equivalence classes induced by E_{Op}.

In case of the PACE algebra, we obtain for the set of equations that are necessary to run the protocol the partition $E_{Op} = E_F \uplus E_C$, where

$$\begin{aligned} E_C = \{&dec(m_0, enc(m_0, m_1)) = m_1,\ enc(m_0, dec(m_0, m_1)) = m_1, \\ &fst(\langle m_0, m_1 \rangle) = m_0,\ snd(\langle m_0, m_1 \rangle) = m_1\} \text{ and} \\ E_F = \{&dh(dh(m_0, m_1), m_2) = dh(dh(m_0, m_2), m_1)\}. \end{aligned}$$

E_F consists of the equation on dh that is essential for a DH exchange.

Since we use pairs (constructed by $\langle ., . \rangle$) to represent initial knowledge, we need (in addition to enc and dec) the cancellation operators fst and snd to select the respective pair components.

Due to the absence of common function symbols in E_F and E_C, it is easy to check that all critical pairs (in rewriting by E_C modulo E_F) are joinable and thus the existence of normal forms.

Inductive Reasoning. Whereas for search-based approaches efficient rewriting is sufficient, for inductive proofs in an algebraic setting we need an axiomatization of the induced initial models.

Initial Models. The primary objectives are axioms that allow us to infer inequalities of message terms with variables and to characterize the (*actual*) substructures of these terms for the recursive definition of auxiliary functions and relations. For this purpose we have used an envisioned measure

$|.|$ on terms, which is consistent with the substructure relation on equivalence classes. Such a measure exists in our setting, e.g., as the maximal term depth in finite representations of equivalence classes.

Using the measure $|.|$, we are now able to specify whether cancellation operators occur as *constructors* and thereby inequalities. These cases are systematically axiomatized according to E_C. For instance, the corresponding case of *enc* is expressed with $|m_2| < |enc(m_1, m_2)|$, since a cancellation (by *enc* instantiating m_2 to $dec(m_1, m_3)$) would lead to a subterm of the second argument. For this constructor occurrence of *enc*, we know additionally that m_1 is a substructure of $enc(m_1, m_2)$, i.e. $|m_1| < |enc(m_1, m_2)|$. Similar axioms provide us with the necessary means for inductive reasoning about messages.

Auxiliary Functions and Relations. Based on the axiomatized structure of messages, we have (recursively) defined several auxiliary functions and relations that are heavily used in our inductive proofs about protocol traces. In particular, we want to mention a local test function *orig* that we have used to prove *confidentiality*.

Given a set *sec* of messages that have to be protected together with an arbitrary message m, $orig(m, sec)$ determines whether an element of *sec* can be analyzed (can *orig*inate) from m assuming that all messages except those in *sec* are available (to the attacker). For instance, suppose we erroneously did not mark a password π as secret (in *sec*). Then $orig(enc(\pi, s), \{s\}) = true$, since s can be obtained by $s = dec(\pi, enc(\pi, s))$.

Using $Orig(sec) = \{m \mid orig(m, sec) = true\}$ the main theorem is:

$$(Orig(sec) \cap ik = \emptyset) \Rightarrow (Orig(sec) \cap DY(ik) = \emptyset)$$

Having fixed a suitable set *sec* of secrets this theorem allows us to reduce the confidentiality of all elements $s \in sec$ expressed by $s \notin DY(spies(tr))$ to $Orig(sec) \cap spies(tr) = \emptyset$. That is, the main proof work shifts to the instantiation of *sec* and this heavily depends on the protocol messages. Intuitively, the set *sec* needs to contain with every message part that will be proven confidential those message parts used for its protection.

The local test function *orig* is a pessimistic approximation as $orig(m, sec) = true$ for some $m \in spies(tr)$ does not imply that for some secret $s \in sec$ we have $s \in DY(spies(tr))$.

Conclusion/Future Works. We have applied our techniques to PACE using the VSE tool. Parts of the theories in the axiomatization were used in prior case studies [4, 9] with a freely generated data type for messages. It thus sufficed to add new VSE theories for the new notions. Altogether, the complete axiomatic system has about 1700 lines of code.

Besides the main PACE properties (5) we have proved 43 help lemmata. Most of these proofs are very large (> 2000 proof nodes). By using (theory-specific) heuristics for selecting the appropriate tactics, the average degree of automation was higher than 70 %. Since parts of the theories (in the specification) were reused from prior case studies, the corresponding heuristics were already available in the VSE prover. Nevertheless, it was necessary to implement new proof heuritics to deal with the new notions.

Our results apply not only to (the) PACE (algebra). They can be straightforwardly transferred to any message algebra where the sets E_F and E_C do not share common function symbols. The axiomatization and the theory-specific proof heuristics used in the PACE proof can be re-used after some simple adaptations.

Currently we are extending our approach to deal with another (more challenging) class of message algebras, where E_F contains equations about cancellation operators. We also plan to enhance the automatization degree by implementing more proof heuristics.

References

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

[2] Viganò, L.: Automated Security Protocol Analysis with the AVISPA Tool. Proceedings of the XXI Mathematical Foundations of Programming Semantics (MFPS'05), ENTCS 155:61–86, Elsevier (2005)

[3] ProVerif: Cryptographic protocol verifier in the formal model. http://prosecco.gforge.inria.fr/personal/bblanche/proverif/.

[4] Cheikhrouhou, L., Nonnengart, A., Stephan, W., Koob, F., Rock, G.: Automating Interactive Protocol Verification. KI '08: Proceedings of the 31st annual German conference on Advances in Artificial Intelligence, pp. 30–37, Springer, Heidelberg (2008)

[5] Ullmann, M., Kügler, D., Neumann, H., Stappert, S., Vögeler, M.: Password Authenticated Key Agreement for Contactless Smart Cards. 4th Workshop on RFID Security (RFIDSec 2008),pp. 140–161.

[6] Federal Office for Information Security (BSI), TR-03110, Version 2.05, Technical Guideline: Advanced Security Mechanisms for Machine Readable Travel Documents – Extended Access Control (EAC), Password Authenticated Connection Establishment (PACE), and Restricted Identification (RI)

[7] Basin, D., Mödersheim, S., Viganò, L.: Algebraic Intruder Deductions. In Proceedings of LPAR05, LNAI 3835, pp 549–564, Springer-Verlag (2005)

[8] Abadi, M., Cortier, V.: Deciding knowledge in security protocols under equational theories. In Theoretical Computer Science, Volume 367, Issues 1-2, 24 November 2006, pages 2-32.

[9] Cheikhrouhou, L., Rock, G., Stephan, W., Schwan, M., Lassmann, G.: Verifying a chip-card-based biometric identification protocol in VSE. In J. Górski (ed.) SAFECOMP 2006. LNCS, vol. 4166, pp. 42–56. Springer, Heidelberg (2006)

Lassaad Cheikhrouhou
German Research Center for Artificial Intelligence,
Saarbrücken,
Germany
e-mail: lassaad@dfki.de

Werner Stephan
German Research Center for Artificial Intelligence,
Saarbrücken,
Germany
e-mail: stephan@dfki.de

Markus Ullmann
Federal Office for Information Security,
Bonn,
Germany
e-mail: markus.ullmann@bsi.bund.de