Automated Analysis of Security Protocols

Michael Rusinowitch [a]

[a] *LORIA, INRIA - Lorraine*
Campus Scientifique, 54506 Vandoeuvre-Lès-Nancy Cedex, France
rusi@loria.fr

Cryptographic protocols such as IKE, SET, TLS, Kerberos have been developed to secure electronic transactions. However the design of such protocols often appears to be problematic even assuming that the cryptographic primitives are perfect, i.e. even assuming we cannot decrypt a message without the right key. An intruder may intercept messages, analyse them, modify them with low computing power and then carry out malevolent actions. This may lead to a variety of attacks such as well-known Man-In-The-Middle attacks.

Even in this abstract model, the so-called Dolev-Yao model, protocol analysis is complex since the set of states to consider is huge or infinite. One should consider messages of any size, infinite number of sessions. The interleaving of parallel sessions generates a large search space. Also when we try to slightly relax the perfect encryption hypothesis by taking into account some algebraic properties of operators then the task gets even more difficult.

We have developed in Cassis/LORIA team several translation and constraint solving techniques for automating protocol analysis in Dolev-Yao model and some moderate extensions of it. Protocol specifications as they can be found in white papers are compiled by the CASRUL system [8,2] and then are passed on decision procedures for checking whether they are exposed to flaws [1].

The compiler: The CASRUL compiler performs a static analysis to check the executability of the protocol (i.e. whether each principal has enough knowledge to compose the messages he is supposed to send), and then compiles the protocol and intruder activities into an Intermediate Format based on first-order multiset rewriting. The Intermediate Format unambiguously defines an operational semantics for the protocol. Afterwards, different translators can be employed for translating the Intermediate Format into the input language of different analysis tools. The Intermediate Format can be also generated in a typed mode (the untyped one is the default),

[1] This work was partially funded by the Information Society Technologies programme of the European Commission, Future and Emerging Technologies under the IST-2001-39252 AVISPA project

which leads to smaller search spaces at the cost of abstracting away type-flaws (if any) from the protocol. This compiler is the front-end of several efficient verification tools designed in the context of the European projects AVISS/AVISPA [1]. It is open to connection with other tools. Compared to CAPSL, a related tool, CASRUL is able to handle protocols where a principal receives a message $\{Na\}_S$, denoting a random number Na encrypted by a symmetric key S, and then later receives the key S and then uses Na in some message. In our case, the principal will store $\{Na\}_S$ and will decrypt it when he later receives the key. This might be useful for e-commerce protocols such as non-repudiation protocols.

The intruder model: Our intruder model follows the Dolev-Yao intruder [7]. That is, the intruder has complete control over the network and he can derive new messages from his initial knowledge and the messages received from honest principals during protocol runs. To derive a new message, the intruder can compose and decompose, encrypt and decrypt messages, in case he knows the key. Hence we can represent the intruder by an appropriate deduction system on message terms as follows:

Decomposition			Composition		
$\{a,b\}$	$\vdash$	a,b	a,b	$\vdash$	$\{a,b\}$
$K^{-1},\{a\}_K$	$\vdash$	a	a,K	$\vdash$	$\{a\}_K$
$S,\{a\}_S$	$\vdash$	a	a,S	$\vdash$	$\{a\}_S$

The second decomposition rule for instance specifies that a message encrypted by the public (resp. private) key K can be recovered by the intruder when he knows the associated private (resp. public key). Whether the intruder can compose some message m from a given set of messages M can be checked in polynomial time. This can be explained easily. In a minimal length derivation of m from M the intruder only composes messages that are subterms of the goal m or messages that are (composed) keys encrypting some part of its initial knowledge M. In a similar way the intruder only decomposes subterms of M. Hence in a minimal derivation of m from M only subterms of $M \cup \{m\}$ are derived. Therefore such a derivation has its length bounded by the number of subterms in $M \cup \{m\}$ say n. Hence by iterating the intruder deduction rules on M at most n times we are able to check whether m can be derived, and this can be organized in polynomial time.

Constraint solving for verification. The detection of flaws in a hostile environment can be reduced to solving symbolic constraints in the message space. The intruder should send messages that comply with the protocol specification in order to get undetected by honest agents. After the last

protocol step, the intruder should be able to derive the secret from the replies sent by the honest agents (and possibly using some initial knowledge). Whether there exists an attack on a protocol in a given number of sessions is an NP-complete problem [9]. The interleaving of parallel sessions is a source of combinatorial explosion in the search procedure. However even with a single session the selection of messages to be sent by the intruder introduces enough non-determinism to be a cause for NP-hardness. On the other hand, we have proved that when the protocol is insecure (i.e. subject to an attack), then there always exists an attack such that the sum of sizes of the messages exchanged between agents is linear w.r.t. the protocol size.

Experiments. The CASRUL implementation of symbolic constraints solving is relatively efficient and has been successful with many security problems. For a report on experiments see [2]. The CASRUL tool has a web-based graphical user-interface (accessible at: `http://www.loria.fr/equipes/cassis/softwares/casrul/`). Among the 51 protocols in the Clark/Jacob library [6], 46 can be expressed in the tool specification language. Among the 51 protocols 35 are flawed, and CASRUL can find a flaw in 32 of them, including a previously unknown type flaw in Denning Sacco Protocol. The tool finds the 32 attacks in less than one minute (1.4 GHz processor).

Algebraic operators: relaxing perfect cryptography hypothesis. Our approach allows for a uniform treatment of many extensions of Dolev Yao intruder. These extensions have been designed to get more realistic models of cryptographic protocols.

If we want to express that in the protocol model we consider plaintexts and ciphered texts cannot be distinguished we can introduce the following intruder deduction rule:

$$a \quad \vdash \quad \{\{a\}_S\}_S$$

We can also consider an intruder equipped with the ability to exploit prefix properties of encryption algorithms based on cipher-block-chaining (CBC). This again can be expressed by introducing a new deduction rule:

$$\{a, b\}_S \quad \vdash \quad \{a\}_S$$

We have recently extended the framework and the complexity results to the case where messages may be constructed using the bitwise XOR operator (denoted $\oplus$) and where the Dolev-Yao intruder is extended by the ability to compose messages with this $\oplus$ operator (see [5]). This extension is non-trivial due to the complex interaction of the $\oplus$ algebraic properties such as associativity and commutativity and the standard Dolev-Yao intruder rules. The technical problems raised by the equational laws are somewhat related to those encountered in semantic unification.

We finally present in [4] the first decision procedure for the formal analysis of protocols in presence of modular exponentiation with products allowed in exponents. This is particularly interesting for finding flaws in protocols based on Diffie-Hellman key construction technique. We illustrate in [4] that the procedure is powerful enough to uncover known attacks on the A-GDH.2 protocol suite.

References

[1] A. Armando, D. Basin, M. Bouallagui, Y. Chevalier, L. Compagna, S. Moedersheim, M. Rusinowitch, M. Turuani, L. Vigano, L. Vigneron. The AVISS Security Protocols Analysis Tool - system description In *Int. Conference on Automated Verification '02*, LNCS, Copenhagen, Denmark, July 27-31, 2002.

[2] Y. Chevalier and L. Vigneron. A Tool for Lazy Verification of Security Protocols. In Proceedings of ASE'01. IEEE CS Press, 2001.

[3] Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols In Int. Conference on Automated Verification '02, LNCS, Copenhagen, Denmark, July 27-31, 2002.

[4] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani. Deciding the Security of Protocols with Diffie-Hellman Exponentiation and Products in Exponents. FST-TCS 03. December 15–17, 2003. Indian Institute of Technology, Bombay. Mumbai, INDIA Lecture Notes in Computer Science.

[5] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani. An NP Decision Procedure for Protocol Insecurity with XOR, 18th Annual IEEE Symposium on Logic in Computer Science June 22 - 25, 2003, Ottawa, Canada.

[6] J. Clark and J. Jacob. A Survey of Authentication Protocol Literature. Version 1.0, November 1997. `http://www.cs.york.ac.uk/ jac/papers/drareview.ps.gz`.

[7] D. Dolev and A. Yao. On the security of public key protocols. In *Proc. IEEE Symp. on Foundations of Computer Science*, pages 350–357, 1981.

[8] F. Jacquemard, M. Rusinowitch, and L. Vigneron. Compiling and Verifying Security Protocols. In *Proc. of LPAR'00*, LNCS 1955, pp. 131–160. Springer, 2000. In M. Parigot and A. Voronkov, editors, *Proceedings of LPAR 2000*, LNCS 1955, pages 131–160. Springer, 2000.

[9] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proceedings of 14th IEEE Computer Security Foundations Workshop*, Cape Breton, Nova Scotia, Canada, June 2001. long version in: Theoretical Computer Science A, 299 (2003), pp 451-475