Verification of Security Protocols

Yannick Chevalier

February 17, 2002

1 Introduction

Internet is becoming everyday a more widely used medium for electronic commerce. This development is hampered by the natural insecurity of communications, as it is not possible to guarantee that some data exchanged is not listened by someone else, or even that it really originated from the claimed sender. This lack of security lead to the development of *security protocols*, that is small messages sequences, which after the author provides some properties to the user, such as the correct identification of the users (called *principals*) and the privacy of some data pieces. These protocols may be seen as small, loopless parallel programs, upon which some properties have to be verified. Over the years, many protocols were shown to be flawed, that is they fail to provide the properties stated by their authors.

Model-checking is one of the most succesful formal methods used for the verification of security protocols. One gives an operationnal semantic to a protocol in terms of a transition system, and the search for a flaw is stated as the reachability of some flawed states. The transitions model the possible actions of the principals, such as sending and receiving messages, and of the intruder, such as intercepting and decomposing messages, and composing and sending new ones from his knowledge. The main disavantage of this method is that one can only handle finite models. This is often achieved by bounding the size of messages, and by bounding the number of principals.

We represents the messages exchanged, the principals and the knowledge of the intruder as terms (see [9] for more details), and define an operationnal semantic for protocols and intruders behavior by translating a high-level specification of a protocol into a rewrite system. The search for a flaw in a protocol then amounts to the search of an inconsistency in the produced rewrite system. Given an initial state and a goal, one is sure to find a flaw whenever there is one.

Our main contribution was to represent intruder's behavior using a resolution strategy complete for the reachability of an inconsistency with respect to the standard, non-terminating, intruder's behaviour as defined in [8], The advantages of our representation are a finite and small search space, the possibility to consider terms of unbounded size (thus permitting the discovery of type flaws) and the possibility to handle counpound keys (terms of unbounded size) for encryption. This strategy is an extension of the one given in [1], where it is only defined for atomic symmetric keys.

We have also given in [6] an abstraction of principals that permits to handle cases in which a principal may participate in an unbounded number of runs of the protocol at the same time.

2 Intruder's description

The main contribution of our work concerns the description of intruder's behavior.

In Dolev-Yao intruder's model [8], the intruder is allowed to compose new terms from terms he already knows. The immediate drawback of this model is that the intruder may compose terms of arbitrary size.

A restriction to this model was proposed in [2]. A static analysis is performed on messages exchanged by the principals, in order to know what messages shall be composed by the intruder. The intruder can no longer compose terms of arbitrary size, and this limitation is complete with respect to flaws that are not type flaws. The drawback of this method is the state-space explosion, which make it unsuitable for the study of large protocol.

Protocol	Ourresults	Lowe's results	Clark & Jacob's results	Brackin's results
Protocols using symmetric encryption				
ISO Symmetric Key One-Pass Unilateral Authentication Protocol	Attack	Attack	No attack	No attack
ISO Symmetric Key Two-Pass Mutual Authentication Protocol	Attack	Attack	No attack	No attack
Andrew Secure RPC Protocol	Attack	Attack	Attack	Attack
Davis Swick Private Key Certificates, Protocol 1	Attack	Attack	Attack	Attack
Davis Swick Private Key Certificates, Protocol 2	Attack	Attack	Attack	Attack
Davis Swick Private Key Certificates, Protocol 3	Attack	Attack	No attack	No attack
Davis Swick Private Key Certificates, Protocol 4	Attack	Attack	No attack	No attack
Denning Sacco Symmetric Key Protocol	Attack	No attack	No attack	No attack
Wide Mouthed Frog Protocol	Unanalyzed	Attack	Attack	No attack
Needham Schroeder Protocol with Conventionnal Key	Attack	No attack	Attack	Attack
Otway Rees Key exchange Protocol	Attack	Attack	Attack	No attack
Yahalom Protocol	No attack	No attack	Attack	Attack
Woo and Lam Authentication Protocol Π_1	Attack	Attack	Attack	No attack
Woo and Lam Authentication Protocol Π_2	Attack	Attack	Attack	No attack
Woo and Lam Authentication Protocol Π_3	Attack	Attack	Attack	No attack
Woo and Lam Authentication Protocol Π	Attack	Attack	Attack	No attack
Woo and Lam Mutual Authentication	Attack	Attack	Attack	No attack
Needham Schroeder Signature Protocol	Attack	Attack	No attack	Attack
Kao Chow Repeated Authentication Protocol	Attack	No attack	Attack	Attack
Kehne Langendorfer Schoenewalder	Attack	Attack	Attack	No attack
Neumann Stubblebine	Attack	Attack	Attack	No attack
Protocols using hash functions				
ISO One-Pass Unilateral Authentication Protocol with CCFs	Attack	Attack	No attack	No attack
ISO Two-Pass Mutual Authentication Protocol with CCFs	Attack	Attack	No attack	No attack
Protocols using public key encryption				
ISO Public Key One-Pass Unilateral Authentication Protocol	Attack	Attack	No attack	No attack
ISO Public Key Two-Pass Mutual Authentication Protocol	Attack	Attack	No attack	No attack
Diffie-Hellman Exchange	Unanalyzed	Unanalyzed	No attack	No attack
Needham Schroeder Public Key Protocol	Attack	Attack	Attack	No attack
SPLICE/AS Authentication Protocol	Attack	Attack	Attack	No attack
Hwang and Chen's modified SPLICE/AS Authentication Protocol	Attack	Attack	Attack	No attack
Denning Sacco Key Distribution with Public Key	Attack	Attack	Attack	No attack
CCITT X.509	No attack	Attack	Attack	No attack
Shamir Rivest Adelman Three Pass Protocol	Attack	Attack	Attack	Unanalyzed
Encrypted Key Exchange Protocol	Attack	Attack	Attack	No attack
TMN Key Exchange Protocol	Attack	Unanalyzed	Unanalyzed	Unanalyzed

Table 1: Comparison of our approach with Lowe's, Clark&Jacob's and Brackin's results

The intruder description we use, on the other hand, performs a dynamic analysis in order to derive the terms the intruder must compose. This analysis can be done by using constraints that state messages or messages parts have to be composed from the knowledge the intruder had at the time the message was received (see [5] for a more detailed description).

3 Experimental Results

We have implemented this into rules for the daTac theorem prover, and made a protocol compiler, `casrul`. The compilation and search for flaws in a protocol is automated.

We compare our results with those obtained with two different approaches, and a summary of attacks found by the end of 1997. In table 1, Lowe's results are obtained with a model-checker FDR. Brackin's are obtained with the HOL proof system using the belief logic formalism [4, 3]. We finally added the results given in [7], although they are somewhat outdated.

This table contains only but for the *Denning Sacco symmetric key* protocol, the protocols considered to be flawed in either of the above mentionned articles. The flaw found on this protocol hasn't been reported as far as we know.

One shall note that the flaw on the CCITT X.509 protocol cannot directly be expressed as an authentication nor a secrecy flaw. We were thus unable to express a goal corresponding to one of the property given in the specification of the protocol, which states that the initiator of the protocol must know a specific term.

We also do not find a flaw in the Yahalom protocol, but the one reported in this case is a type flaw on a part of a message. It is dubious, since the intruder will not be able to compose entirely the message.

4 Conclusion

The method presented permitted us to detect all previously known expressible flaws, and moreover permitted to discover a flaw on a protocol that was previously reported as secure. We are able to correctly find attacks in which type flaws are involved, which is not the case of other known tools. Moreover, implementations of this algorithm yields very fast analysis time.

It can still be improved in several ways. First, we were not able to express some of the flaws reported. Second, the handling of principals active in an unbounded number of protocol runs at the same time may be improved, in order to have a richer fresh data model. And last, we were also able to study protocols using XOR encryption, but only by simplifying the nilpotence property of XOR. Moreover, we were not able to study cases in which timestamps properties are used to detect the replay of messages.

We plan to extend the expressivity of the input syntax and of the generated rules by adding timestamps properties, more expressive goals and properties of operators. The issue raised are those of unification in a non-empty theory, and of expressibility of some properties in first-order logic.

References

[1] R.M. Amadio and D. Lugiez. On the reachability problem in cryptographic protocols. In *CONCUR 2000*, pages 380–394, 2000.

[2] Roscoe B. *The theory and practice of concurrency*. Prentice Hall, 2001. ISBN 0-13-674409-5.

[3] S.H. Brackin. Automatic formal analysis of cryptographic protocols. URL: `http://www.arca.com/projects/docs/brackin/nissc97u.pdf`, 1997.

[4] S.H. Brackin. Evaluating and improving protocol analysis by automatic proof. In *Proc. of the 11IEEE Computer Security Foundation Workshop*. IEEE Computer Society Press, 1998.

[5] Y. Chevalier and L. Vigneron. Towards Efficient Automated Verification of Security Protocols. In *Proceedings of the Verification Workshop (VERIFY'01) (in connection with IJCAR'01), Università degli studi di Siena, TR DII 08/01*, pages 19–33. 2001.

[6] Y. Chevalier and L. Vigneron. Automated Unbounded Verification of Security Protocols. Research Report 4369, Institut National de Recherche en Informatique et Automatique, Nancy (France), January 2002. submitted to CAV'02.

[7] John Clark and Jeremy Jacob. A Survey of Authentication Protocol Literature: Version 1.0, 17. Nov. 1997. URL http://www.cs.york.ac.uk/ jac/papers/drareview.ps.gz.

[8] Danny Dolev and Andrew Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 2(29), 1983.

[9] Florent Jacquemard, Michael Rusinowitch, and Laurent Vigneron. Compiling and Verifying Security Protocols. In M. Parigot and A. Voronkov, editors, *Proceedings of LPAR 2000*, LNCS 1955, pages 131–160. Springer-Verlag, Heidelberg, 2000.