The computational and decisional Diffie-Hellman assumptions in CryptoVerif

Bruno Blanchet and David Pointcheval

CNRS, École Normale Supérieure, INRIA
{blanchet,pointche}@di.ens.fr

We present an extension of CryptoVerif to Diffie-Hellman key agreements. CryptoVerif [1] is a security protocol verifier sound in the computational model, which produces proofs by sequences of games. CryptoVerif provides a generic method for specifying security assumptions on primitives. However, this method did not support the computational and decisional Diffie-Hellman assumptions. We have extended it to support these assumptions, which required the following additions:

– Diffie-Hellman key agreements consider a cyclic group G with generator g. One protocol participant A chooses a random exponent a and publishes g^a, another one B chooses a random b and publishes g^b, then both participants compute g^{ab} by $(g^b)^a$ for A or by $(g^a)^b$ for B. For representing g^{ab}, one needs to access b in A and a in B, and these variables are not in scope. We have extended the language for specifying security assumptions to support that (through the "array accesses" already used elsewhere in CryptoVerif).
– When one uses the computational Diffie-Hellman (CDH) assumption, one typically computes a key by $h(g^{ab})$ where h is a hash function in the random oracle model. This hash function is replaced by CryptoVerif with a lookup that compares the argument of h, g^{ab}, with arguments of previous calls to h, say x, and returns the previous result if there is a previous call to h with the same argument. Using the CDH assumption, CryptoVerif replaces the comparison $x = g^{ab}$ itself with a lookup (which tests whether a or b have been revealed to the adversary), thus creating a lookup in the condition of a lookup; we have extended CryptoVerif to support such nested lookups.
– The decisional Diffie-Hellman (DDH) assumption says that, when the adversary has only g^a and g^b, g^{ab} is indistinguishable from g^c for a random c. Hence we abort the game when we try to give a or b to the adversary after having replaced g^{ab} with g^c, since this replacement would be incorrect when the adversary has a or b. We have extended the language for specifying security assumptions to support such abortions.

We apply these extensions to a simple signed Diffie-Hellman protocol and to a variant of the password-based key exchange EKE. They also open the possibility of verifying many important protocols in CryptoVerif (IPSec, SSH, and some modes of TLS and Kerberos).

References

1. B. Blanchet. A computationally sound mechanized prover for security protocols. *IEEE Transactions on Dependable and Secure Computing*, 5(4):193–207, 2008.