Refining Computationally Sound Mechanized Proofs for Kerberos

B. Blanchet* A. D. Jaggard† J. Rao‡ A. Scedrov§ J.-K. Tsay¶

Kerberos is designed to allow a user to repeatedly authenticate herself to multiple servers based on a single login. The PKINIT extension to Kerberos modifies the initial round of the protocol to use a PKI instead of long-term shared keys (*e.g.*, password-derived keys). Especially with PKINIT, Kerberos uses a rich collection of cryptographic operations and constructs, and Kerberos, both with and without the PKINIT extension, is used in real world settings (including Microsoft Windows). Kerberos is thus a great test case for protocol-analysis tools. The CryptoVerif prover works directly in the computational model to prove properties of protocols that are formalized as games.

This talk will both survey some of our earlier work using CryptoVerif to analyze Kerberos, with and without PKINIT, and describe two recent extensions of this work. First, we briefly survey our work [1] to formalize all three rounds of Kerberos (with and without PKINIT) as games that CryptoVerif could analyze. This allowed us to prove, using CryptoVerif, authentication and secrecy properties under certain cryptographic assumptions (*e.g.*, that the public-key encryption scheme satisfies IND-CCA2 security). This work included the definition of a version of key usability that was stronger than that originally given by Datta *et al.* [2]; the stronger version is amenable to being proved using CryptoVerif, and we showed that freshly generated keys in Kerberos are usable in this strong sense for IND-CCA2-secure encryption.

Second, we describe more recent results that extend our initial work on key usability. We suggest the following definition of strong key usability for INT-CTXT-secure encryption; like our strong notion of IND-CCA2 usability, this definition can be captured in the language used by CryptoVerif.

Definition 1 (Strong INT-CTXT Key Usability)**.** *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme ($\mathcal{K}$ is the key generation algorithm, $\mathcal{E}$ the encryption algorithm, and $\mathcal{D}$ the decryption algorithm), Σ be a key exchange protocol, and $\mathcal{A}$ be an adversary. We consider an experiment $\mathbf{Exp}^*{}_{\mathcal{A},\Sigma,\Pi}(\eta)$ and define the advantage of an adversary $\mathcal{A}$ by $\mathbf{Adv}^{*ke}_{\mathcal{A},\Sigma,\Pi} = \Pr[\mathbf{Exp}^*{}_{\mathcal{A},\Sigma,\Pi}(\eta) = 1]$. Let S be a set of symmetric encryption schemes. We say that keys exchanged through protocol Σ are* strongly INT-CTXT usable *for schemes in S if, for all $\Pi \in S$ and any probabilistic, polynomial-time adversary $\mathcal{A}$, the advantage $\mathbf{Adv}^{*ke}_{\mathcal{A},\Sigma,\Pi}$ is negligible.*

The experiment $\mathbf{Exp}^*{}_{\mathcal{A},\Sigma,\Pi}(\eta)$ proceeds as follows:

- *First, $\mathcal{A}$ is given the security parameter η and $\mathcal{A}$ can interact, as an active adversary, with polynomially many protocol sessions of Σ.*
- *At some point, at the request of $\mathcal{A}$, a session identifier* sid *is drawn at random and $\mathcal{A}$ is given access to a encryption oracle $\mathcal{E}_k(.)$ and a decryption oracle $\mathcal{D}_k(.)$, both keyed with a key k locally output in session* sid*.*
- *Adversary $\mathcal{A}$ plays a variant of an INT-CTXT game in which:*
  - *$\mathcal{A}$ may submit messages m to $\mathcal{E}_k(.)$, which returns $\mathcal{E}_k(m)$;*
  - *$\mathcal{A}$ never queries $\mathcal{D}_k(.)$ on a cyphertext output by $\mathcal{E}_k(.)$;*
  - *$\mathcal{A}$ may interact with uncompleted protocol sessions; and*

*CNRS, Ecole Normale Supérieure, INRIA, Bruno.Blanchet@ens.fr. Partially supported by the ANR project FormaCrypt and by DGA.

†DIMACS, Rutgers University, adj@dimacs.rutgers.edu. Partially supported by NSF Grants CNS-0751674 and CNS-0753492 and by ONR Grant N00014-07-1-1039.

‡mail.jesse.rao@gmail.com.

§Department of Mathematics, University of Pennsylvania, scedrov@math.upenn.edu. Partially supported by ONR Grant N00014-07-1-1039, by OSD/AFOSR MURI "Collaborative policies and assured information sharing," and by NSF Grants CNS-0429689, CNS-0524059, and CNS-0830949.

¶Department of Electrical Engineering and Information Sciences, Ruhr-University Bochum, joe-kai.tsay@trust.rub.de.

– *if k is a key that is used at least once for encryption, then $\mathcal{A}$ never queries $\mathcal{D}_k(.)$ on a cyphertext encrypted by any key playing the role of k in any one of the protocol sessions. (*I.e.*, in Kerberos, if k is an authentication, resp. service, key that is used at least once for encryption, then $\mathcal{A}$ never queries $\mathcal{D}_k(.)$ on a cyphertext encrypted by* any *authentication, resp. service, key.)*

- *the experiment outputs* 1 *if the decryption oracle properly decrypts a query by the adversary,* i.e.*, outputs $m \neq \perp$, otherwise the experiment outputs* 0.

We then use CryptoVerif to show that the various fresh keys in Kerberos (with and without PKINIT) are usable in this sense. (As in [1], our cryptographic assumptions here include that the symmetric encryption scheme used in the protocol is INT-CTXT secure; however, INT-CTXT usability was not defined at that time and we only proved usability for IND-CCA2-secure encryption.)

Third, we update the formalization in a number of important technical ways and study the cryptographic assumptions needed by Kerberos. Prompted by and extending comments by Chao Feng [3], we add additional oracles to provide certificates to parties other than just the client. More precisely, we add two oracle processes called CCERT and KCERT, to our model of Public-key Kerberos in CryptoVerif's process calculus: The first oracle process (CCERT) allows the attacker to obtain valid certificates for dishonest clients, whereas the second oracle process (KCERT) allows the attacker to obtain valid certificates for dishonest Kerberos Authentication Servers (KASes). Both oracles are relevant for public-key Kerberos, as in this setting neither the client nor the KAS need to be pre-registered to each other before starting a protocol session (in contrast to basic Kerberos). One may consider KCERT redundant under the assumption that any KAS is a trusted third party; however, the lack of CCERT, which was brought to our attention by Chao Feng [3], prevented some insider attacks, in particular the man-in-the-middle attack on the flawed version PKINIT-25 [4].

Because of these additional oracles, we also need to study some cryptographic requirements on the MAC in a part of the protocol. CryptoVerif is not able to prove some security properties for public-key Kerberos if, as in [1], we assume only UF-CMA security for the MAC used for the `asChecksum` field in PKINIT. CryptoVerif succeeds if we instead make the assumption that the MAC is an HMAC based on a family of collision-resistant hash functions. That is, we are assuming $HMAC_i(k, m) = h_i(k \oplus \mathsf{opad}||h_i(k \oplus \mathsf{ipad}||m))$, where m is a message to be hashed, k is the MAC-key, and i is a non-secret index of the collision-resistant hash function family $\{h_i\}_{i \in I}$. We stress that although CryptoVerif is not able to complete the security proofs for public-key Kerberos if one assumes general UF-CMA security for the MAC for `asChecksum`, we did not discover an attack on PKINIT under this assumption. We believe that CryptoVerif fails to complete the proofs both for model-dependent reasons and because it is presently not capable of temporal reasoning.

In comparison, in the computational analysis of the symmetric encryption scheme of Kerberos in [5], the assumptions on the MAC algorithm are stronger than UF-CMA. There, the MAC is assumed to be a PRF; this holds if the MAC is an HMAC and the underlying compression function is a PRF. This can be seen as another indication that one needs to make strong assumptions on the MAC to prove the correctness of Kerberos.

With the added oracles and the changes we made to the cryptographic assumptions, we are still able to prove authentication, secrecy, and usability results as before. As with our new work on INT-CTXT usability, this illustrates how CryptoVerif is useful for exploring different cryptographic definitions and assumptions.

References

[1] B. Blanchet, A. D. Jaggard, A. Scedrov, and J.-K. Tsay, "Computationally Sound Mechanized Proofs for Basic and Public-Key Kerberos," In *ASIACCS*, pp. 87–99 (2008).

[2] A. Datta, A. Derek, J. C. Mitchell, and B. Warinschi, "Computationally Sound Compositional Logic for Key Exchange Protocols," In *CSFW*, pp. 321–334 (2006).

[3] C. Feng, Personal communication (2009).

[4] I. Cervesato, A. D. Jaggard, A. Scedrov, J.-K. Tsay, and C. Walstad, "Breaking and Fixing Public-Key Kerberos", *Information and Computation* **206**, pp. 402–424 (2008).

[5] A. Boldyreva and V. Kumar, "Provable-Security Analysis of Authenticated Encryption in Kerberos", In *IEEE Symposium on Security and Privacy*, pp. 92–100 (2007).