Computationally Sound Mechanized Proof of PKINIT for Kerberos *

Aaron D. Jaggard† Andre Scedrov‡ Joe-Kai Tsay§

Abstract. Here we report initial results on the formalization and analysis, using the CryptoVerif tool [4, 5, 6], of the public-key extension to the Kerberos protocol, PKINIT [10]. This protocol provides a good test case for analysis techniques because it incorporates many different protocol design elements: symmetric and asymmetric encryption, digital signatures, and keyed hash functions. We are able to prove, using CryptoVerif's interactive mode, secrecy and authentication properties for PKINIT at the computational level. Because Kerberos appears to be more complex than the protocols previously analyzed using CryptoVerif, our work provides evidence of the suitability of CryptoVerif for the analysis of real-world industrial protocols.

This work is part of an ongoing project to formalize and analyze the Kerberos protocol suite; earlier work has included symbolic proofs (by hand) of security properties of the basic protocol [7], the discovery of a flaw in a draft version of PKINIT (which led to a Windows Security Bulletin [12]) and the symbolic proof of its fixes [8], and by-hand computational proofs of the security of Kerberos with the fixed version of PKINIT using the BPW model [2]. The current work extends this project to include the use of a mechanized tool, Blanchet's CryptoVerif (v. 1.06).

Kerberos and PKINIT. Kerberos [14] is designed to allow a user to repeatedly authenticate herself to multiple servers based upon a single login. The client's interactions with the servers partition the basic Kerberos protocol into three different rounds. Our focus here is on the first round, called the Authentication Service (AS) Exchange in the protocol specification [14]. The PKINIT extension [10] to Kerberos replaces the basic AS exchange, allowing the use of PKI in place of a long-term key shared between the client C and the KAS K. PKINIT does not change either of the later rounds in the Kerberos protocol. Figure 1 shows the AS exchange in the fixed version of PKINIT: The client C generates two nonces and a timestamp and sends these to K in a message that names the server T (in the second round of Kerberos) for which she wants a *ticket-granting ticket* (TGT); the other data in this message are discussed fully in [8]. (Here $\{\_\}_\_$, $\{\!|\_|\!\}_\_$, and $[\_]_\_$ are symmetric encryption, asymmetric encryption, and a digital signature, respectively.) The KAS K generates the keys k and AK and a timestamp and then sends AK to C encrypted under

*Jaggard was partially supported by NSF Grants DMS-0239996 and CNS-0429689 and by ONR Grant N00014-05-1-0818. Scedrov was partially supported by OSD/ONR CIP/SW URI "Software Quality and Infrastructure Protection for Diffuse Computing" through ONR Grant N00014-01-1-0795 and OSD/ONR CIP/SW URI "Trustworthy Infrastructure, Mechanisms, and Experimentation for Diffuse Computing" through ONR Grant N00014-04-1-0725. Additional support from NSF Grants CNS-0429689 and CNS-0524059. Tsay was partially supported by ONR Grant N00014-01-1-0795 and NSF grant CNS-0429689.

†Tulane University. Current address: Rutgers University, adj@dimacs.rutgers.edu

‡University of Pennsylvania, scedrov@math.upenn.edu

§University of Pennsylvania, jetsay@math.upenn.edu

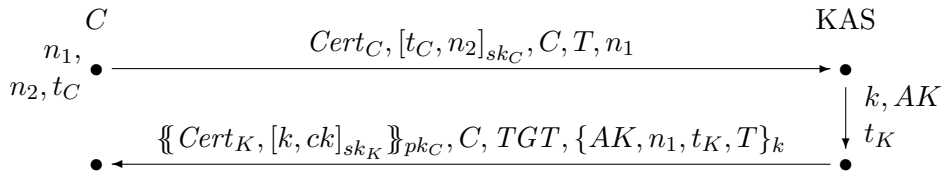

Figure 1: Message flow in the fixed version of PKINIT, where $TGT = \{AK, C, t_K\}_{k_T}$.

k, which in turn is encrypted under C's public key; other parts of this message, again discussed in [8], include the TGT and data to bind this reply to C's request.

CryptoVerif. The tool CryptoVerif [4, 5, 6], developed by Blanchet, can directly prove protocol security in the computational model. Security protocols are formalized using a probabilistic polynomial-time process calculus that is motivated by the pi-calculus and the calculi introduced in [11] and [13]. In this calculus, messages are bitstrings and cryptographic primitives are functions operating on bitstrings. Given a security parameter η, CryptoVerif proofs are valid for a number of protocol sessions polynomial in η, in the presence of an active adversary. The process calculus represents games, and proofs are represented as sequences of games, where the initial game formalizes the protocol for which one wants to prove certain security properties. In a proof sequence, two consecutive games Q and Q' are *observationally equivalent*, meaning that they are indistinguishable for the adversary. CryptoVerif transforms one game into another by applying, *e.g.*, the security definition of a cryptographic primitive. In the last game of a proof sequence the desired security properties should be obvious. CryptoVerif operates in two modes: a fully automatic and an interactive mode. The interactive mode, which is best suited for protocols using asymmetric cryptographic primitives, requires a CryptoVerif user to input commands that indicate the main game transformations the tool should perform. CryptoVerif is sound with respect to the security properties it shows in a proof. Of course, properties it cannot prove are not necessarily invalid.

Properties proved. We have used CryptoVerif to prove secrecy and authentication properties for PKINIT considered by itself, without the later rounds of Kerberos. The key AK that K generates and sends to C in the second PKINIT message is used to protect communications between C and T in the next round, so it is important for this key to be secret. CryptoVerif proved, for the fixed version of PKINIT, the secrecy of AK (*i.e.*, it proved the query `query secret keyAK`, where `keyAK` is the name in the CryptoVerif source file of the key labeled AK in Fig. 1). This property, automatically defined by CryptoVerif, says that PKINIT preserves the secrecy of the key `keyAK` with respect to the *real-or-random* definition of security, which is a stronger notion than the standard notion from the literature [1]. CryptoVerif also proved authentication properties for the fixed version of PKINIT; these properties must be specified by the user, although CryptoVerif does have predefined keywords, *etc.*, that are typically used to state authentication properties. CryptoVerif proved authentication of K to C by proving the query: `query x:bitstring, k:key; event inj:fullC(K,k,x) ==> inj:fullK(C,k,x).` This correspondence assertion means that there is an injective relationship between the events `fullC(K,k,x)` (identified with the client process finishing a run of PKINIT with K) and `fullK(C,k,x)` (identified with the KAS process finishing a run of PKINIT with C). In particular, when the client process completes its participation in

PKINIT, it creates an event `fullC(hostZ, AK, (m1, m2))` that contains the name `hostZ` of the server K, the name `AK` of the fresh key AK, the client's first message in `m1`, and the reply from K (but without the TGT, the ticket for the second round, or the associated MAC) in `m2`. When the KAS process completes its participation in PKINIT, it creates an event `fullK(hostY, AK, (m3, m4))` that contains the name `hostY` of the client C, the name `AK` of the fresh key that the KAS has just sent to C, the message `m3` to which the KAS is replying, and the reply `m4` without the TGT or associated MAC. Thus, the injective correspondence property proved by CryptoVerif means that every time a client processes a reply from K, there is a corresponding unique instance of the KAS sending a reply to C.

For the flawed draft version of PKINIT, CryptoVerif was not able to produce a positive proof of either the secrecy of the key AK or the authentication of K to C. In fact, neither property holds for the flawed protocol, due to a known attack [8].

Cryptographic Assumptions. The public-key encryption scheme is assumed to be indistinguishable under adaptive chosen ciphertext attacks (IND-CCA2), and the signature scheme is assumed to be unforgeable under chosen message attacks (UF-CMA). Symmetric encryption is implemented as encrypt-then-MAC, where the symmetric encryption scheme is assumed to be indistinguishable under chosen plaintext attacks (IND-CPA), and the message authentication code is weakly unforgeable under chosen message attacks (UF-CMA). This guarantees indistinguishability under adaptive chosen ciphertext attacks (IND-CCA2) and integrity of plaintexts (INT-PTXT) for the symmetric encryption, as shown in [3]. These cryptographic assumptions are slightly weaker than made in our previous work [2], where we considered all three rounds of Kerberos, both with and without PKINIT. In particular, we assumed in [2] the symmetric encryption scheme to guarantee integrity of ciphertexts (INT-CTXT).

Conclusions. We have formalized and mechanically analyzed the PKINIT extension to the Kerberos authentication protocol using version 1.06 of Blanchet's CryptoVerif tool. The success of CryptoVerif in proving security properties for PKINIT—a particularly complex part of the Kerberos suite—gives evidence of its utility for analyzing industrial protocols. This also extends our ongoing Kerberos analysis project to include mechanized tools.

We are currently investigating using weaker assumptions and how CryptoVerif is responsive to the degradation of the strengths of the assumed cryptographic primitives. From our experience with CryptoVerif so far, it seems evident that one needs to know the underlying cryptography well in order to use the tool.

PKINIT is only a small fragment of the Kerberos suite, so we are working to use CryptoVerif to prove results for larger fragments. We would like to use CryptoVerif to analyze all three rounds of Kerberos (both with and without PKINIT), as we did in [2]. It would then be interesting to investigate how the notion of *key usability*, as introduced in [9], will be relevant. Of particular interest is also the Diffie-Hellman mode of PKINIT, which we did not study here. As noted by Blanchet [4], the language of equivalences used by CryptoVerif will need to be extended in order to handle Diffie-Hellman key exchange, so this provides an interesting open problem extending our work here.

Acknowledgements. We are grateful to Bruno Blanchet, Michael Backes, John Mitchell, and Arnab Roy for helpful discussions.

References

[1] Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-Based Authenticated Key Exchange in the Three-Party Setting. *IEE Proceedings Information Security*, 153(1):27–39, 2006.

[2] Michael Backes, Iliano Cervesato, Aaron D. Jaggard, Andre Scedrov, and Joe-Kai Tsay. Cryptographically Sound Security Proofs for Basic and Public-key Kerberos. In *ESORICS 2006*, volume 4189 of *LNCS*, pages 362–383. Springer Verlag, September 2006.

[3] Mihir Bellare and Chanathip Namprempre. Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm. In *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer Verlag, December 2000.

[4] Bruno Blanchet. A Computationally Sound Mechanized Prover for Security Protocols. In *IEEE Symposium on Security and Privacy*, pages 140–154, May 2006.

[5] Bruno Blanchet. Computationally Sound Mechanized Proofs of Correspondence Assertions. In *CSF 2007*, July 2007. To appear.

[6] Bruno Blanchet and David Pointcheval. Automated Security Proofs with Sequences of Games. In *CRYPTO 2006*, volume 4117 of *LNCS*, pages 537–554. Springer Verlag, August 2006.

[7] Frederick Butler, Iliano Cervesato, Aaron D. Jaggard, Andre Scedrov, and Christopher Walstad. Formal Analysis of Kerberos 5. *Theoretical Computer Science*, 367(1–2):57–87, 2006.

[8] Iliano Cervesato, Aaron D. Jaggard, Andre Scedrov, Joe-Kai Tsay, and Christopher Walstad. Breaking and Fixing Public-key Kerberos. In *ASIAN 2006*, LNCS. Springer Verlag. To appear. Preliminary version available at `http://eprint.iacr.org/2006/009`.

[9] Anupam Datta, John Mitchell, and Bogdan Warinschi. Computationally Sound Compositional Logic for Key Exchange Protocols. In *CSFW 2006*, pages 321–334, July 2006.

[10] IETF. Public Key Cryptography for Initial Authentication in Kerberos, 1996–2006. RFC 4556. Preliminary versions available as a sequence of Internet Drafts at `http://tools.ietf.org/wg/krb-wg/draft-ietf-cat-kerberos-pk-init/`.

[11] Peeter Laud. Secrecy Types for a Simulatable Cryptographic Library. In *CCS 2005*, pages 71–85, May 2005.

[12] Microsoft. Security Bulletin MS05-042. `http://www.microsoft.com/technet/security/bulletin/MS05-042.mspx`, August 2005.

[13] John Mitchell, Ajith Ramathan, Andre Scedrov, and Vanessa Teague. A Probabilistic Polynomial-Time Process Calculus for the Analysis of Cryptographic Protocols. *Theoretical Computer Science*, 353(1–3):118–164, 2006.

[14] Clifford Neuman, Tom Yu, Sam Hartman, and Kenneth Raeburn. The Kerberos Network Authentication Service (V5), July 2005. `http://www.ietf.org/rfc/rfc4120`.