Brief Announcement: Anonymity and Trust in Distributed Systems

Michael Backes
Saarland University, Germany MPI-SWS
backes@cs.uni-saarland.de

Stefan Lorenz
Saarland University, Germany
stefan.lorenz@stud.uni-saarland.de

Matteo Maffei
Saarland University, Germany
maffei@cs.uni-saarland.de

Kim Pecina
Saarland University, Germany
pecina@cs.uni-saarland.de

ABSTRACT

In this paper, we present a framework for achieving anonymity and trust, two seemingly contradictory properties, in distributed systems. Our approach builds on webs of trust, a well-established and widely deployed decentralized infrastructure for establishing the authenticity of the binding between public keys and users, and more generally, trust relationships among users. We introduce the concept of anonymous webs of trust – an extension of webs of trust where users can authenticate messages and determine each other's trust level without compromising their anonymity. Our framework comprises novel cryptographic protocols based on zero-knowledge proofs for achieving anonymity in webs of trust and a prototype implementation based on GnuPG. We conduct an automated analysis to formally verify the security of our protocol and an experimental evaluation to demonstrate the effectiveness of our approach.

Categories and Subject Descriptors: C.2.0 [General]: Security and Protection

General Terms: Security, Verification

Keywords: Anonymity in decentralized systems, cryptographic protocols, formal verification

1. INTRODUCTION

Over the last years, the Web has evolved into the premium forum for freely disseminating and collecting data, information, and opinions. Not all principals, however, are willing to reveal their true identity to avoid, for instance, associations with their race, ethnic background, or other sensitive characteristics. The ability to anonymously exchange information, and hence the inability of users to identify the information providers and to determine their credibility, raises serious concerns about the reliability of exchanged information.

Webs of trust (WOT) constitute a well-established approach to bind public keys to their owners and, more generally, to establish trust relationships among users in a decentralized manner. This trust is expressed by signing the public keys that are considered trustworthy along with a set of user and key attributes (e.g., user name and key expiration date). These signatures are called *trust signatures*. For ease of the presentation, in this paper we consider only signatures on public keys. In the following, we let $(\mathsf{sk}_P, \mathsf{pk}_P)$ denote P's key pair, composed of the private signing key sk_P and the public verification key pk_P; $\mathsf{sig}(\mathsf{m}, \mathsf{sk}_P)$ denote the signature on message m with key sk_P; and $\mathsf{ver}(\mathsf{m}, S, \mathsf{pk}_P)$ denote the successful verification of signature S on message m with key pk_P. Trust signatures can be chained in order to express longer trust relationships: For instance, the signature chain $\mathsf{sig}(\mathsf{pk}_A, \mathsf{sk}_B), \mathsf{sig}(\mathsf{pk}_B, \mathsf{sk}_C)$ says that Charlie has signed Bob's key (i.e., Charlie trusts Bob) and Bob has signed Alice's key. Intuitively, this signature chain says that Alice is a friend of a friend of Charlie. To authenticate message m, Alice sends the signature $\mathsf{sig}(\mathsf{m}, \mathsf{sk}_A)$ to Charlie, along with the signature chain. Charlie in turn checks whether the statement $\mathsf{ver}(\mathsf{pk}_B, \mathsf{sig}(\mathsf{pk}_B, \mathsf{sk}_C), \mathsf{pk}_C) \wedge \mathsf{ver}(\mathsf{pk}_A, \mathsf{sig}(\mathsf{pk}_A, \mathsf{sk}_B), \mathsf{pk}_B) \wedge \mathsf{ver}(\mathsf{m}, \mathsf{sig}(\mathsf{m}, \mathsf{sk}_A), \mathsf{pk}_A)$ holds true, i.e., there is a signature chain from Charlie to Alice and Alice signed message m.

Our contributions We introduce the concept of *anonymous webs of trust* – an extension of webs of trust that allows users to authenticate messages and determine each other's trust level without compromising their anonymity. More precisely, (*i*) we propose two cryptographic protocols to enforce two of the most widely considered anonymity properties, namely *global anonymity* and *k-anonymity*; (*ii*) we conduct a formal, automated security analysis of our protocols; (*iii*) we develop a prototype implementation and experimentally evaluate its performance. The implementation and the long version of this paper are available at [1].

Our framework is general and can be incorporated as a plug-in into existing distributed applications to enforce fine-grained trust and anonymity properties. The potential application scenarios include distributed social networks, where people may want to share opinions or information anonymously while being able to prove their trust relationships, applications for anonymous message exchange, and services for anonymous yet trustworthy reviews.

2. ANONYMOUS WEBS OF TRUST

The idea behind anonymous webs of trust is to use zero-knowledge proofs[1] to show the knowledge of signature

[1]A *zero-knowledge proof* combines two seemingly contradictory properties. First, it is computationally infeasible to produce a zero-knowledge proof of a wrong statement. Second, a zero-knowledge proof does not reveal any information besides the bare fact that the statement is valid [3]. These proofs were originally very inefficient and of limited use in

PODC'10, July 25–28, 2010, Zurich, Switzerland.
ACM 978-1-60558-888-9/10/07.

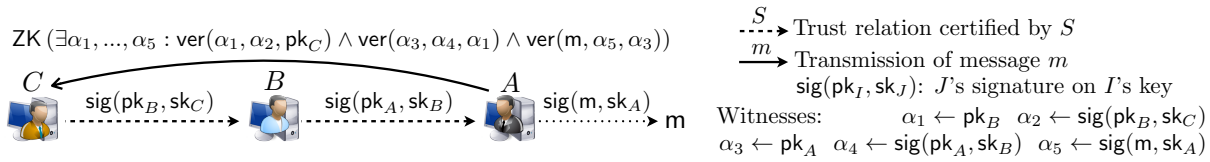

Figure 1: Protocol for anonymous proof of a signature chain of length 2

chains, while hiding any information on the sender's identity. In this way, the sender proves her trust level[2] to the receiver without compromising her anonymity. We propose two cryptographic protocols, for achieving *global anonymity* and *k-anonymity*.

Global anonymity is achieved by proving the knowledge of signature chains, while hiding the witnesses that could reveal some information about the sender's identity, namely, all signatures and public keys except the receiver's public key. In the previous example, Alice would prove the statement $\exists\alpha_1,\ldots,\alpha_5 : \mathsf{ver}(\alpha_1,\alpha_2,\mathsf{pk}_C) \wedge \mathsf{ver}(\alpha_3,\alpha_4,\alpha_1) \wedge \mathsf{ver}(m,\alpha_5,\alpha_3)$. Such a proof does not reveal anything other than the existence of a signature chain of length 2 linking the sender to the receiver. An overview of the overall protocol is given in Figure 1. This scheme can be extended to cover signatures on attributes and to prove the existence of multiple chains, thus accommodating sophisticated trust models (see [1] for more detail).

k-anonymity is achieved by proving the knowledge of a signature on message m which originates from a principal within a group $\mathcal{G}$ of k people. If the recipient of m shares a friendship relation with all principals in $\mathcal{G}$, he is ensured that m comes from a friend but does not know which principal in $\mathcal{G}$ has actually sent m. For instance, if Dave is also a friend of Charlie, Alice could prove that m is signed by Dave or by Alice. Formally, the statement of this zero-knowledge proof looks as follows: $\exists\alpha : \mathsf{ver}(\mathsf{m},\alpha,\mathsf{pk}_A) \vee \mathsf{ver}(\mathsf{m},\alpha,\mathsf{pk}_D)$.

3. AUTOMATED SECURITY ANALYSIS AND EXPERIMENTAL EVALUATION

Formal Analysis We conducted a formal security analysis by modeling our protocol in the applied pi-calculus, formalizing the trust property as a security policy and the anonymity property as an observational equivalence relation, and verifying our model with ProVerif [2], a state-of-the-art automated theorem prover for cryptographic protocols. The trust policy requires that whenever a principal P authenticates a message m as coming from a principal of trust level i, then a friend of P with trust level i intended to authenticate message m. Anonymity is instead formalized as a cryptographic game: A protocol guarantees anonymity if and only if, after every successful protocol run, no external observer can determine which principal generated the proof. In our analysis, we consider a strong active attacker who dictates the signatures released by each party (i.e., the attacker controls the web of trust) and decides which signature chains are to be proven in zero-knowledge.

practical applications. The recent advent of efficient proofs for special classes of statements has paved the way for the deployment of these proofs into modern cryptographic protocols.

[2]Here, the trust level corresponds to the length of the proven certificate chain.

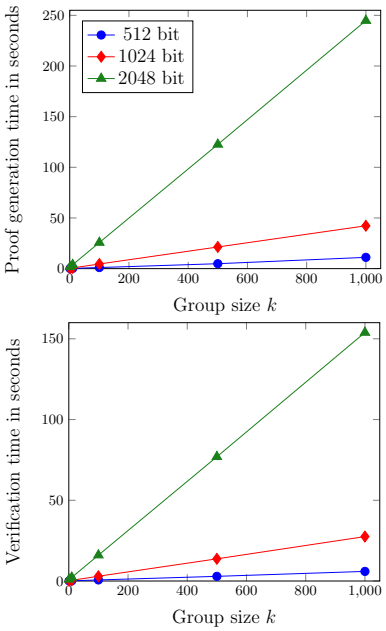

Figure 2: Proof generation and verification time for different key sizes and group sizes k.

Experimental evaluation The communication complexity of the protocol for global anonymity is linear in the key-size, while the communication complexity of the protocol for k-anonymity is independent of the key-size. The protocol for global anonymity turns out to be practical for key sizes up to 1024 bits. We can obtain, however, security guarantees even with keys shorter than 1024 bits, since our protocol proactively refreshes keys after a (user-defined) time interval, in order to deal with dynamic trust relationships. The protocol for k-anonymity is instead very efficient, as shown by the graphs in Figure 2. These results are obtained on a dual core processor with 2.6 GHz and 4 GB ram.

Acknowledgments

This work was partially supported by the initiative for excellence and the Emmy Noether program of the German federal government and by Miur Project SOFT (*Security Oriented Formal Techniques*).

4. REFERENCES

[1] M. Backes, S. Lorenz, M. Maffei, and K. Pecina. Anonymous webs of trust. Available at `www.lbs.cs.uni-sb.de/awot/index.html`.

[2] B. Blanchet. An efficient cryptographic protocol verifier based on Prolog rules. In *CSFW'01*, pages 82–96. IEEE, 2001.

[3] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):690–728, 1991.