POSTER: Deniable Anonymous Group Authentication

Ewa Syta
Ph.D. Student
Yale University
Email: ewa.syta@yale.edu

Bryan Ford
Faculty
Yale University
Email: bryan.ford@yale.edu

I. Introduction

At first glance, anonymous authentication, the ability to authenticate yourself without revealing who you are, seems like a counterintuitive concept. However, it is surprisingly useful in many applications where a regular authentication scheme can be employed, especially, if users can prove certain claims about themselves, such as that they are a unique user or belong to a particular group.

Online subscriptions are a good example of an application where access should be limited to a group of registered users but knowing which users are accessing particular resources is not necessary. For example, a streaming media website does not necessarily need to know that a user is watching a particular movie as long as she is a paying user. Such an approach offers greater privacy protection to users while still allowing the content provider to remain in control of its resources and even in certain cases keep track of users' preferences by linking actions of particular anonymous users.

Anonymous electronic surveys and voting schemes are additional examples of applications where the real identity need not to be verified as long as group membership (i.e., a user belongs to a group of eligible voters) and proportionality (i.e., a user can vote once) is ensured.

Furthermore, anonymous communication systems, such as Dissent [3, 7], can leverage anonymous authentication to set up ephemeral pseudonyms for group members wishing to communicate anonymously. However, since a user's identity is defined by a long-term non-anonymous key pair, a compromise of user's private key may retroactively compromise the user's anonymity in all past exchanges. Therefore, achieving a stronger notion of anonymity (under a full compromise of a user's private key) would offer much greater protection and encourage use of such systems.

In response to these needs, we present a deniable, anonymous group authentication (DAGA) protocol that provides *anonymity*, *proportionality*, *deniability* and *forward anonymity*. The anonymity property allows a user to authenticate as *some* group member without revealing *exactly* which one. Proportionality ensures that a client can authenticate only once as a unique group member per round. Deniability makes it possible to deny ever participating in a protocol. Finally, forward anonymity offers protection in case of a compromise of user's private key after the protocol's completion.

To the best of our knowledge, there is no other scheme that provides all four properties. Group signatures [2] require a trusted third party and a fixed group roster. Ring signatures [6] offer greater flexibility by removing these two requirements but still lack the proportionality property. Linkable ring signatures [4] address this issue but do not offer forward anonymity. Deniable ring authentication [5] offers protection against compromised private keys but lacks proportionality.

II. Overview

A. Security Model

We assume an *anytrust* [7] model, where there is a set of n clients, at least two of which are honest, and a smaller set of m reliable servers which includes at least one honest server; the clients need not to know which server to trust. A client wishing to authenticate sends any of the servers an authentication message and the servers collectively process the authentication request.

We assume that there is a readily available group definition $G = (\vec{X}, \vec{Y})$ listing clients and servers and their long-term public keys, X_i and Y_j respectively. The author of a group definition may conscript arbitrary clients knowing only their public keys. Some of the clients listed in the group definition need not ever participate in the protocol or even be aware that they are included. We further assume that the servers are always online and participate in every step of the protocol.

B. Properties

Anonymity. Informally speaking, we want to ensure that after a complete protocol run, an adversary cannot guess which group member has been authenticated with a probability greater than random guessing.

Forward Anonymity. We extend the anonymity property to situations in which an adversary obtains a client's private key but only after a protocol run has completed.

Deniability. We want to ensure that the protocol does not leave a "paper trail" that could be used to link a client to his actions.

Proportionality. A client can authenticate as a unique member only once given a particular authentication context and each subsequent authentication request within the same context are recognized as coming from that client.

III. Protocol Details

Each client authenticates using a publicly available *authentication context* $C = (G, \vec{R}, \vec{H}, g)$, which consists of a group definition G, a set $\vec{R}$ of server's per-round public randomness, a set $\vec{H}$ of client's per-round generators, and a generator g of a large order group $\mathcal{G}$. To generate $\vec{R} = (R_1, \ldots, R_m)$, each server chooses a random secret r_j and publishes $R_j = g^{r_j}$. $\vec{H} = (h_1, \ldots, h_n)$ consists of n unique per-round generators of $\mathcal{G}$, one for each client i, such that no one knows the logarithmic relationship between any h_i and g or between h_i and $h_{i'}$ for any pair of clients $i \neq i'$.

A. Steps performed by the client

To authenticate, a client i generates a *linkage tag* T using a context C, and he proves in zero knowledge that he correctly computed T with respect to the assigned generator h_i *and* that he is a member of a group G because he knows a private key x_i that corresponds to one of the keys included in $\vec{Y}$. We use a generalized version of an "OR" proof of knowledge of one out of two discrete logarithms and other standard proofs of knowledge about discrete logarithms [1].

To generate the tag, a client i first computes a shared secret s_j for every server j such that each server is able to reconstruct only its own secret shared with i. Then, using his per-round generator h_i and all m shared secrets s_j, i computes the tag as $T = h_i^{s_1 s_2 \ldots s_m}$. Finally, i executes an "OR" proof that he knows one of n private keys and that the tag T is correctly computed using h_i:

$$\text{PK} = \{\vee_{i=1}^{n}(\text{I know private key } x_i \wedge \text{T is correctly based on } h_i)\}$$

Client i securely erases every s_j and creates a message M, which consists of the context C, the linkage tag T, the proof P and all other information needed by the servers to reconstruct their shared secrets and confirm their validity.

B. Steps performed by the servers

Upon receiving a message M, the first server verifies the client's proof P and if it is valid, the server processes his tag T. The server reconstructs and scrubs the randomness s_1 it shares with i, adds his own per-round random secret r_1, and then proves that he correctly performed these steps and that he indeed knows r_1. The remaining servers repeat this process, however, also verifying that the proof coming from the previous server is valid. Each message M_j contains all prior messages, including M, so that each server j can verify the client's proof as well as the behavior of every other server that processed the tag thus far.

Therefore, a linkage tag produced by server j contains the randomness of servers that come before j and the secrets client i shares with the servers that come after j: $T_j = T_{j-1}^{r_j/s_j} = h_i^{r_1 \ldots r_j s_{j+1} \ldots s_m}$. The proof server j generates is as follows:

$$\text{PK} = \{(\text{Tag } T_j \text{ is correct} \wedge \text{I know my secret } r_j)\}$$

Provided that the client i and the servers correctly follow the protocol, it yields a final linkage tag $T_m = h_i^{r_1 r_2 \ldots r_m}$.

C. Achieving the properties

Anonymity and Forward Anonymity. The authentication is anonymous if authentication transcripts of any two members $i, i' \in G$ are indistinguishable, and it is forward anonymous if the transcripts remain indistinguishable given the knowledge of private keys. Under the DDH assumption, an adversary cannot decide if any of the tags $T = h_i^{s_1 s_2 \ldots s_m}$, $T_{j-1} = h_i^{r_1 r_2 \ldots r_m}$, and $T_m = h_i^{r_1 r_2 \ldots r_m}$ is created with respect to h_i or $h_{i'}$ without the honest server's secrets. Further, the knowledge of x_i does not aid the adversary.

Deniability. The authentication is deniable if an authentication transcript of any member $i \in G$ can be simulated by a verifier. The tag T does not depend on i's private input and the proof P can be simulated because it is zero knowledge.

Proportionality. A linkage tag T depends only on a client's unique generator h_i and every r_j, thus, a client i can obtain one and only one linkage tag within the same context C, even if he authenticates several times.

IV. Conclusion and Future Work

We have outlined a novel authentication scheme that achieves four important properties: anonymity, forward anonymity, deniability, and proportionality. We have begun formally analyzing our protocol and expect to be able to prove these properties under standard security assumptions. We plan to implement a fully functional prototype to determine DAGA's applicability in real world's applications. Possible extensions of the protocol include using diverse client keys (i.e., a mix of DL and RSA keys), authenticating a subset of the group members (i.e., a unique subset k out of n group members requests access), and optional counting of members (i.e., any member can "authenticate" and participate but only specific group members are given privileges).

References

[1] J. Camenisch and M. Stadler. Proof systems for general statements about discrete logarithms. Technical Report 260, Dept. of Computer Science, ETH Zurich, March 1997.

[2] D. Chaum and E. Van Heyst. Group signatures. In *EUROCRYPT*, 1991.

[3] H. Corrigan-Gibbs and B. Ford. Dissent: Accountable anonymous group messaging. In *CCS*, 2010.

[4] J. K. Liu, V. K. Wei, and D. S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups. In *ACISP*, 2004.

[5] M. Naor. Deniable ring authentication. In *In Proceedings of Crypto 2002, volume 2442 of LNCS*, 2002.

[6] R. L. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. In *ASIACRYPT*, 2001.

[7] D. I. Wolinsky, H. Corrigan-Gibbs, A. Johnson, and B. Ford. Dissent in numbers: Making strong anonymity scale. In *OSDI*, 2012.