Distributed Information-Theoretic Anonymous Authentication with Dynamic Membership

We present a protocol that allows a member of a group, P, to authenticate her membership to a server anonymously. I.e. the server should be convinced that the client is a member of a group but has no information about her identity. Basic requirements for such protocol include the following:

1. Security: Only members of group P can be authenticated.
2. Anonymity: The server learns nothing about the client's identity from the authentication
3. Unlinkability: Different transactions should not be determined to be from the same member.

Practical systems tend to have more demands:

1. Dynamic group membership: the protocol should allow adding and removing members.
2. Identity escrow: in case of emergency, an escrow agent can "open" a transaction and reveal the identity of its owner.

Most existing solutions are centralized. They rely on an "issuer " or a group manager (GM) to distribute the keys and/or perform the ID escrow. Both the GM and the escrow agent have to be trusted and they typically are provided by the server (e.g. the owner of the building installs the card key reader). This is a problem if the users are not familiar with the owner of the server. Other than blindly trusting the server, the GM and the escrow agent, which are under control of a single party, the users have no way to convince themselves that their anonymity is protected.

We propose a protocol that is distributed. It empowers the users to make decisions about the membership, ID escrow, etc. The server only receives information distributed by the group and does not have control over issues such as membership change and revealing ID of a client. Instead the server can appeal to the group to take actions if some inappropriate behavior from some client is seen. Assuming sufficient number of members are honest and play according to the rules, the server should also be confident of its own protection. We believe this model strikes a good power balance between the server and the group and is easier to foster trust among them. Our protocol is based on threshold scheme and is secure as long as there are sufficient number of honest members. Both the security and the anonymity are information-theoretic.

Basic Scheme

Let n be the number of members in the group and t is the upper bound of faulty players among them. Assume the group already has set up a (t+1, n)-threshold public key scheme (x0, y0) with y0 being the public key and x0 being (t+1, n)-shared among members. The group runs another DKG (distributed key generation) protocol such as that of Pedersen

[3] to generate a (t+1, n+t) -threshold key (x, y). Now x is (t+1, n+t)-shared. During the DKG, each player should send t shared to the server S and one share to each of the members (I have sorted out the verification issues). The result is that S has t shared of x. The member-held share xi will be used as authentication key only.

To authenticate a client, S selects a number r at random and encrypts it using y, then decrypts t times using its shares of x (I am trying to "combine" all these t shares so that S only needs to send a single message, but not clear how to do it yet. The complexity now is O(t) which is better than [2], another arguably "distributed" protocol). S then sends the partial decryptions and the ciphertext, together with "proof of correctness" (such as that from [4]) to the client. The client should be able to verify them and decrypt them and obtains r and return r as proof of her membership. Since any member with a valid share of x can do this, S has no way to ID her nor can S link transactions to the same client.

Dynamic Membership

The basic scheme needs some fix to allow for dynamic membership.

Adding or removing members from the group can be performed by a set of t+1 players. Adding a member is basically evaluating another point on the degree t polynomial and is the same as reconstructing the secret x, only that this has to be done without revealing x. Easy to accomplish.

To remove a user i, t+1 players work together to recover her authentication key, xi, and send it to S. S now maintains two sets. T: the t shares of x he received from the group initially and R: the shared of all users whose membership have been revoked (together with their indices). We assume |R| < t. S updates R with info from the group. To authenticate a client, S first decrypts the ciphertext (encryption of r) with all the shared from R, and then decrypts using any t-|R| shares from T. A user from R cannot properly decrypt r since this is equivalent to determining a degree t polynomial using only t points. But other members can still do so.

All the above operations (e.g. the index of the revoked member) should be augmented with signatures using the shared group private key x0, to show the data really comes from at least t+1 good players.

Note that as soon as R becomes nonempty, S knows the secret key x. This is OK. x is used to do authentication only and all S can do with x is to possibly allow someone else use its resource in the name of the group (and may later charge the group). This may not be a problem in some applications. If it is, it can be prevented by requiring S to present proof of the transaction, which can be in the form of a ticket signed with x0 (x0 is never revealed – but if a user is disenrolled, should we consider his share of x0 to be leaked) by the group using blind signature. This has some implication to efficiency since the user needs to obtain a ticket for each transaction. Still working on this. A variant can let the group maintain R collaboratively and let S first query the group about R.

Also note that this scheme only supports disenrolling up to t members. After |R| reaches t, the group needs to do DGK again to refresh the keys if another member is to be removed. This is actually not a problem. If t+1 members need to be disenrolled, the secret key x should be considered leaked and the group needs to generate new key anyway.

ID Escrow

Seems any ID escrow scheme should work, as long as the opening can be performed by t+1members. However, the additional steps will make the protocol less efficient (e.g. may need ZKP). Still working on this.

[1] *D. Boneh, and M. Franklin,* Anonymous authentication with subset queries , In proceedings of the *6th ACM conference on Computer and Communications Security*, pp. 113--119.

[2] Stuart E. Schechter, Todd C. Parnell and Alexander J. Hartemink, Anonymous Authentication of Membership in Dynamic Groups, In Financial Cryptography '99, Anguilla, British West Indies, February 1999.

[3] @inproceedings{pedersen91threshold,
```
   author = "Torben Pedersen",
   title = "A Threshold Cryptosystem without a Trusted Party",
   booktitle = "Advances in Cryptology -- EUROCRYPT'91",
   series = "Lecture Notes in Computer Science",
   volume = "547",
   pages = "522--526",
   year = "1991",
   publisher = "Springer-Verlag"
}
```

[4] Shoup, V.: Practical threshold signatures. Technical Report IBM Research Report RZ 3121, IBM (1999)