Cryptography Final Project

Saba Eskandarian

CS 6501

1 Goal

The goal of this project is to create a system that could allow users to communicate anonymously while still being accountable for their statements. This would be achieved by having the identity of a user unmasked if some threshold number of users and another threshold number of moderators all voted to have a user unmasked. The aim is to have a way for the system to be created/users to be added, for users to share messages, and for users and moderators to vote to unmask a user.

Such a system could have applications on internet forums where some users take advantage of the cover of anonymity to make threats of violence or other harm against users towards whom they foster animosity. Ideally, this system would allow for the benefits of anonymous communication to continue, while offering the possibility to unmask those who are engaging in flagrant and unacceptable behavior. The judges of what is considered egregious behavior will be those who participate in the forum, with a distinction made between those in a moderating role and a regular user role, both being required to vote to expose someone who has engaged in unacceptable behavior. Although this system does rely on having a sort of moral consensus among a sufficiently large number of participants, it is hoped that the possibility of accountability could deter hateful individuals from freely attempting to terrorize or intimidate others.

2 Definitions

The primitives used in the solution are Public Key Encryption, Digital Signatures, Secret Sharing, and Verifiable Secret Sharing. Each is defined below, and there are well known schemes that satisfy each definition.

Public Key Encryption: A public-key encryption scheme consists of three polynomial time probabilistic algorithms (Gen, Enc, Dec) with the following properties:

1. Gen takes a security parameter and creates a secret key sk and a public key pk
2. Enc takes a message m and a public key pk and outputs a ciphertext c
3. Dec takes a ciphertext c and a secret key sk and outputs a message m or a failure symbol.

It is necessary that $\Pr[\text{Dec}_{sk}(\text{Enc}_p k(m))=m]= 1 - negl()$. The scheme is CCA secure if no adversary can succeed in an indistinguishability experiment where it is given access to a decryption oracle with probability greater than $\frac{1}{2} + negl(n)$, where n is the security parameter.

Digital Signature: A signature scheme consists of three polynomial time probabilistic algorithms (Gen, Sign, Vrfy) with the following properties:

1. Gen takes a security parameter and creates a signing key sk and a public key pk
2. Sign takes inputs sk and a message m, outputing a signature σ
3. Vrfy takes a message m, a signature σ, and a public key pk and outputs a bit b indicating either valid (1) or invalid (0).

It is necessary that Vrfy(m, Sign$_{sk}(m)$)= 1. A signature scheme is secure if an adversary with access to pk and a signing oracle cannot output a valid message, signature pair with probability more than negligible in the security parameter.

Secret Sharing: Shamir defines (k,n) threshold secret sharing as dividing some data D into n pieces $D_1, ..., D_n$ such that

1. D is easily computable given k or more D_i pieces
2. D is completely undetermined given fewer than k of the D_i pieces

Verifiable Secret Sharing: A verifiable secret sharing scheme accounts for the possibility of a dishonest distributor of D_i pieces in a threshold secret sharing scheme who could potentially give out pieces that are not at all part of the secret that is supposed to be distributed. It guarantees that it is possible to verify the shares of a distributed secret without revealing the secret itself.

Additionally, we define the requirements and security definition for the scheme made for this project as follows: each user of a potential anonymous message board has some private identity string I. Also, each anonymous message board takes two parameters t_m and t_u. There are two different types of users in the system, moderator users and regular users. The system consists of four parts (setup, add user, post, vote) with the following properties:

- setup: A group of participants can come together and create a message board by agreeing on values of t_m, t_u, and user roles of moderators and regular users
- add user: A user arriving after the group wishing to establish the board has been set up may become part of the system
- post: A user can broadcast a message to the rest of the group
- vote: moderators and regular users can vote to unmask other users.

In order for the system to be secure, there are additional constraints on the post and vote operations. Every post should be such that any user other than the sender (or in fact any arbitrary bounded adversary) could only forge the identity of the sender with a negligible probability. Additionally, in the voting, setup, and add user processes, any user's I will be made known to other users if and only if at least t_m moderators and t_u regular users vote to have that user's identity revealed.

3 Solution

The following procedures are followed in order to set up a group of users under this scheme, add additional users to an existing group using the scheme, send a message in the group, and vote to unmask a user in the group. In order for a group of users and moderators to set up the system, they must determine beforehand what the values of t_m and t_u will be in addition to deciding who will be a user, who a moderator, and how many of each there will be, u for number of users and m for number of moderators, $u > t_u, m > t_m$. All participants have (pk_s, sk_s) for signing and a different (pk_e, sk_e) for encryption.

Set up: All participants publish both their pk_s and pk_e. Each user begins by encrypting her I with her pk_s and publishing it (note that although this is the key used for signing messages later, it is used instead of the encryption key to encrypt I). Then each user takes the XOR of her sk_s and a randomly chosen string. The result of this XOR operation and the random string are two random strings that produce sk_s when XORed together. We will call these strings k_m and k_u respectively. A (t_u, u) verifiable secret sharing scheme is then used to distribute shares of k_u among the regular users, and a (t_m, m) verifiable secret sharing scheme is used to distribute shares of k_m among the moderators. The shares must be distributed with the recipients' pk_e in order to prevent others from seeing them.

Add a user: The process for adding a user is at first identical to that which is carried out by each user when the board is first set up. The difference is that the initial setup requires more than t_u users and t_m moderators to participate in order to guarantee that there are enough participants for all the secrets to be distributed. Next, all other users need to send the new user an additional share of either their k_m or their k_u. Adding a user increases the value of u or m, but does not change the thresholds for unmasking any user. A drawback of this scheme is that the proportion of participants needed to unmask a user shrinks as the number of users grows. This can hopefully be compensated for to some degree by the separation of regular users from moderators, as the moderator group size may be less likely to grow.

Send a message: Sending a message makes direct use of a digital signature. The user sending the message publishes a message m signed with $\text{Sign}_{sk_s}(m)$. Other users can verify the authenticity of the message using the posting sending user's pk. Private messages can be sent by encrypting the message with the intended recipient's pk_e.

Vote to unmask: In order for a user or moderator to vote to unmask someone, they only need to publish the share they've been given of the user's sk_s. Once enough shares from both the regular users and moderators have been published, anyone can reconstruct the sk_s and decrypt the user's identity.

4 Sketch of Security Proof

We now endeavor to show that the solution described above satisfies the definition and security requirements set forth before. We will begin with ensuring that I for any user is revealed if and only if at least t_m moderators and t_u regular users have voted to have I revealed by publishing

their shares of I. After that we need to ensure the security of posting messages.

Our setting is as follows: for a given user, each other user has pk_s, pk_e, and either one k_m share or one k_u share. There may also be other k_m and k_u shares published by other users. If t_m shares of k_m are published and t_u shares of k_u are published, then anyone can recover both k_u and k_m, XOR them together to get sk_s and decrypt I, thus revealing the user's identity. Next we show that without the publication of t_m shares of k_m and t_u shares of t_u, no user or outside adversary can decrypt I with more than negligible advantage. Any adversary who could decrypt I without sk_s would be breaking the underlying public-key encryption scheme. In order to get I from the available information, k_m and k_u need to be XORed together, since they have the same distribution as random strings and give no information about sk_s until they are XORed together. This means that an adversary would need to recover both k_m and k_u before being able to decrypt I. But if an adversary could recover k_m or k_u with fewer than t_m or t_u shares having been published, then that adversary would be breaking the security of the secret-sharing scheme used to distribute k_m and k_u pieces. Thus I cannot be decrypted unless at least t_m and t_u shares of k_m and k_u have been published, respectively.

The security of posting is fairly straightforward and follows almost directly from the definitions of the primitives. In the proof that I cannot be revealed unless enough shares of k_m and k_u had been revealed by the moderators and regular users, respectively, we showed that sk_s also cannot be revealed unless that condition is met. This means that sk_s is indeed a secret to the posting user unless the user has already been unmasked, at which point they are presumably no longer welcome to post in the group anyway. With a secret sk_s, the authenticity required of the message is exactly that provided by the definition of a secure digital signature scheme.

5 Sources

A. Shamir, How to Share A Secret, CACM Vol. 22 No. 11, 1979

Chor, Benny; Goldwasser, S.; Micali, S.; Awerbuch, Baruch, "Verifiable secret sharing and achieving simultaneity in the presence of faults," Foundations of Computer Science, 1985., 26th Annual Symposium on , vol., no., pp.383,395, 21-23 Oct. 1985

Jonathan Katz , Yehuda Lindell, Introduction to Modern Cryptography, Chapman & Hall/CRC, 2007