Crypto 1 Project

William Burton
burtow@rpi.edu

Greg Cowan
cowang@rpi.edu

December 7th, 2017

1 Requirements

Alice and Bob each have m files, each file is $>$ 10MB. They want to find out how many files they have in common and which ones are shared.
They start without a shared key, and they communicate over an insecure channel (e.g. LAN, WAN, or Internet).
And, Alice and Bob are semi-honest, so our protocol should handle that.
An eavesdropper should not be able to alter their communication or learn something about the transmitted messages. Alice and Bob should not learn anything else than the number and identities of the common files.
We can use: collision resistant hash functions, existentially unforgeable MACs and digital signatures, and CPA-secure public-key and symmetric-key encryption.

2 Protocol

1. Ahead of time, a private-key, public-key pair is generated for both Alice and Bob.
2. Alice is initialized with Bob's public key, and her private key. Bob is initialized with Alice's public key and his private key. In this case, these are shared by a trusted-courier, the user of the simulation. But they could be shared another secure way in a real application.
3. Alice and Bob use a Hash-and-sign signature sheme, like RSA-FDH, to transform the insecure channel to an authenticated one.
4. Alice and Bob transform the authenticated channel into an encrypted one using the El Gamal scheme.
5. Now, they can start the application-specific protocol.
   (a) Alice and Bob hash one copy of all their files.
   (b) Alice and Bob encrypt their hashes with *their public* keys.
   (c) Alice and Bob send those encrypted hashes to the other one.
   (d) Bob and Alice encrypt their file's hashes with the other's public-key.
   (e) If an encryption matches, then they share that file.