COLLEGE OF ENGINEERING

Electrical Engineering and Computer Science

SHARING AN EMAIL ACCOUNT USING SECURE MULTI-PARTY COMPUTATION

PROBLEM

Two people wish to have an email account from which they can send emails only when both people agree on the text of the email.

This may be useful if, for example, two people are in charge of making decisions over email and wish to communicate that both parties have agreed on the decision in a single email.

REQUIREMENTS

- A single party should not be able to send an email from the shared account on their own.
- It should only be possible to send an email from the account when both parties agree on the exact contents of the email.
- To be able to work with popular email servers, our protocol cannot require the server to run any additional code. Thus, we must make our email protocol appear identical to a regular, single party email protocol to an email server.

MULTI-PARTY COMPUTATION

To meet these requirements, we use a technique called multi-party computation.

Multi-party computation (MPC) is a subfield of cryptography that studies methods for multiple parties to evaluate the output of a function given secret inputs from each party, which they do not wish to reveal.

MPC has been studied since the 1980s, but was impractical for non-trivial applications up until the past decade or so, due to the high computational and bandwidth costs of MPC techniques.

Oregon State University

PROTOCOL OVERVIEW

Emails are sent using a protocol called SMTP, which itself is performed over another protocol called TLS, a protocol for sending and receiving encrypted data over the internet.

TLS begins with a "handshake" process that, among other things, establishes a set of cryptographic keys to be used to encrypt and decrypt data. To meet our requirements, we must ensure that neither user has access to this key, but that they can together compute the correct encryption of some data under the key.

To achieve this, we seek to implement the key agreement process of the TLS handshake as well as the encryption of TLS packets inside of MPC.

The TLS handshake protocol, along with the MPC processes this project seeks to implement, are illustrated on the right.

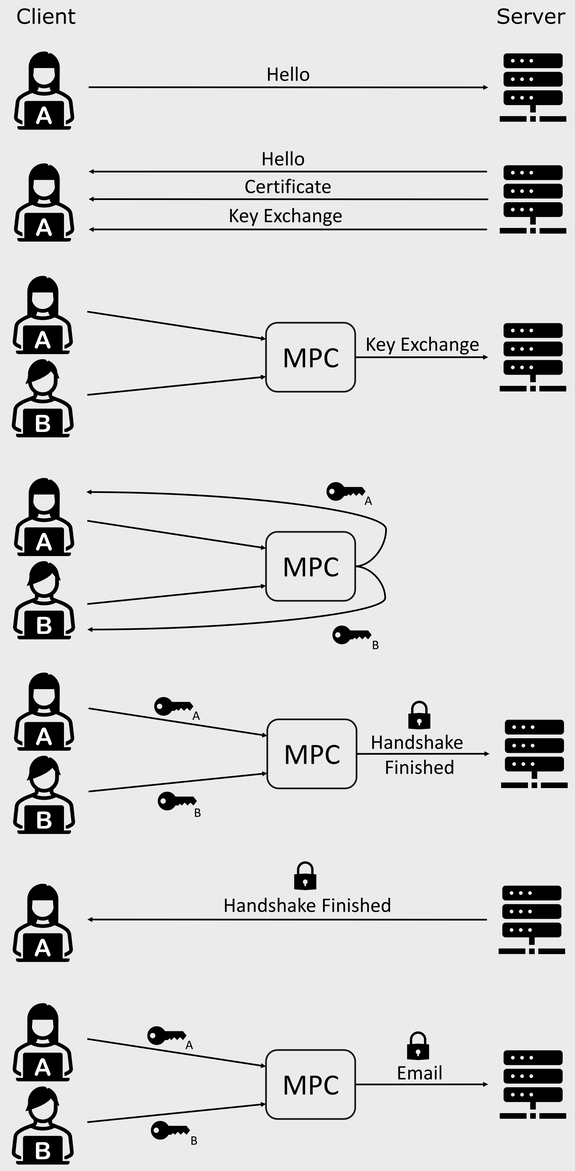
TOOLS

To implement the multi-party computation, we make use of EMP toolkit, which supports authenticated garbling, the MPC technique we use.

This is wrapped in a command-line interface written in Python.

STATUS

This project is still a work in progress. More work needs to be done on the key exchange MPC protocols.



A depiction of the TLS handshake process, with MPC processes allowing for two users to act as the client. Boxes labeled "MPC" represent multi-party computation processes between users A and B.

RYAN LITTLE
littler@oregonstate.edu

ACKNOWLEDGMENTS

This work was a research project done under the supervision of Dr. Mike Rosulek. Thanks to Mike for his guidance and for introducing me to the field of multi-party computation.

Thanks as well to Dr. Kirsten Winters for her advice and patience throughout the capstone process.