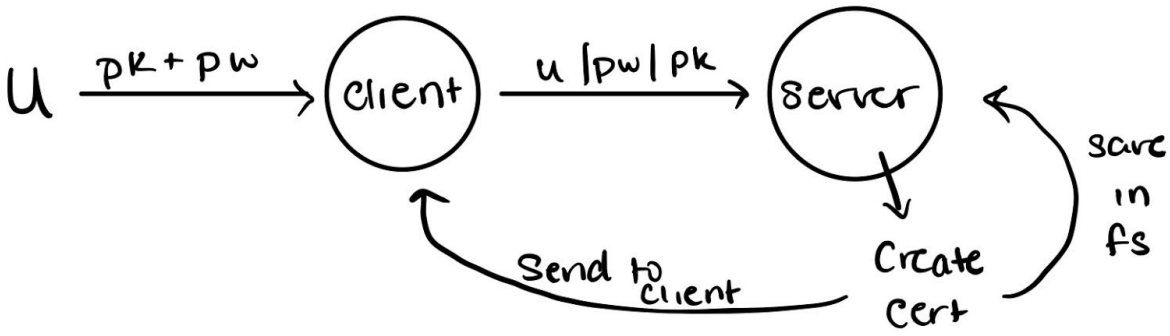
Security Project Design Doc

Mark Ozdemir
Monica Wang
Yasemin Reis
Ava Wood

Design Decisions

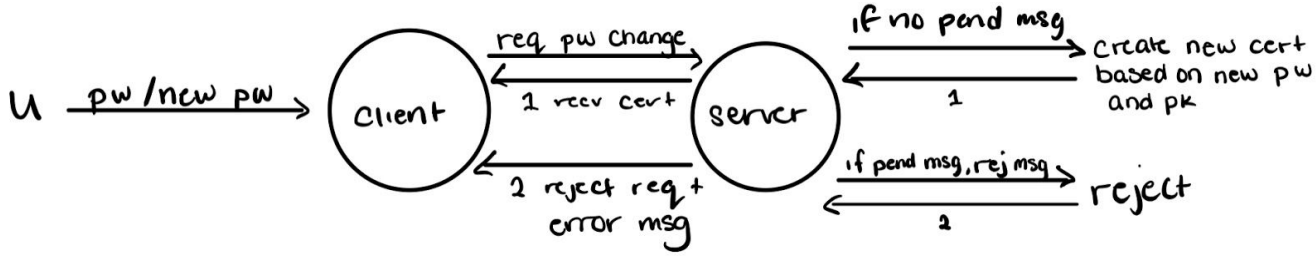
getcert



Save the cert in FS:

- Go to created dir for user (called <username>)
    - Check if the password provided by u matches the hash
    - Create and store new client side cert

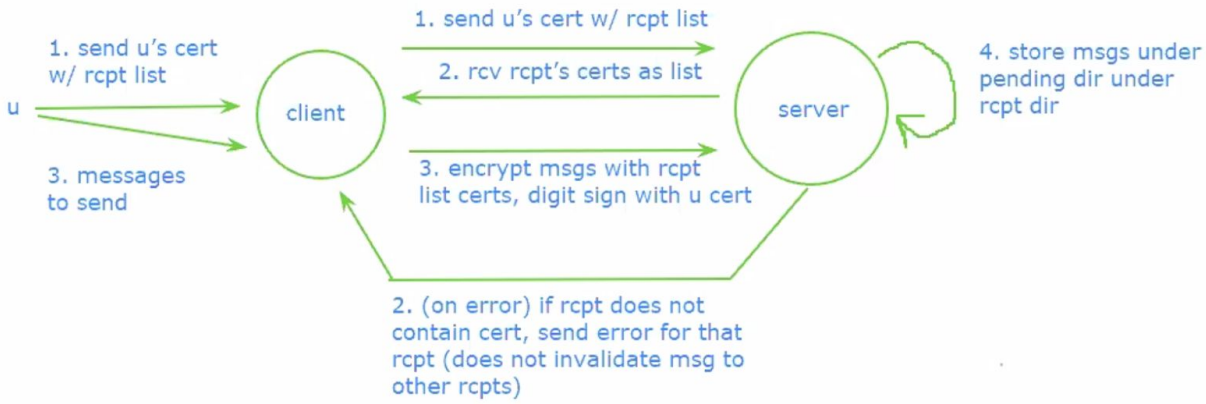
changepw



- If no messages are pending, save in FS

- Go to created dir for u
    - Hash password passed from u via sha256 and check if matches file
    - Hash new password and overwrite the original file
    - Create new cert and store here
- else if pending messages exist
    - Return with err message

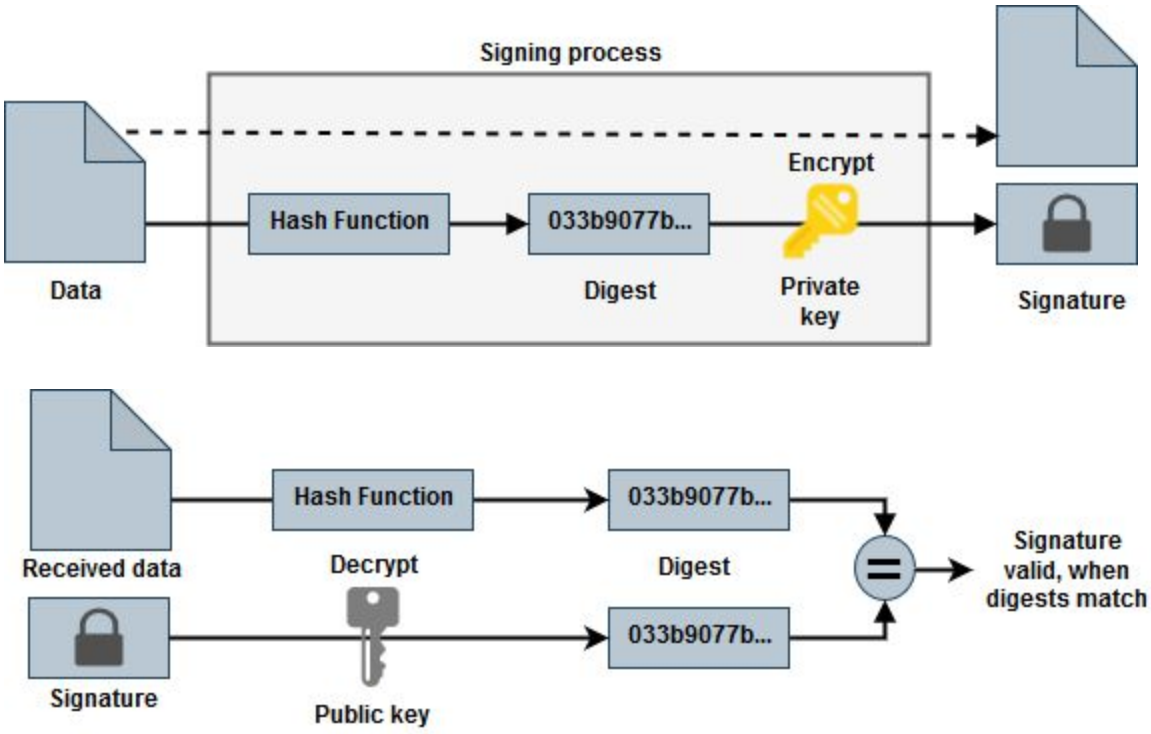
sendmsg



- Search for dir containing username
- Directory for each user (called <username>)
    - File for hashed password
    - File for client side certificate
- If rcpt does not have certificate
    - reject email to rcpt
    - does not invalidate msg to all other rcpts

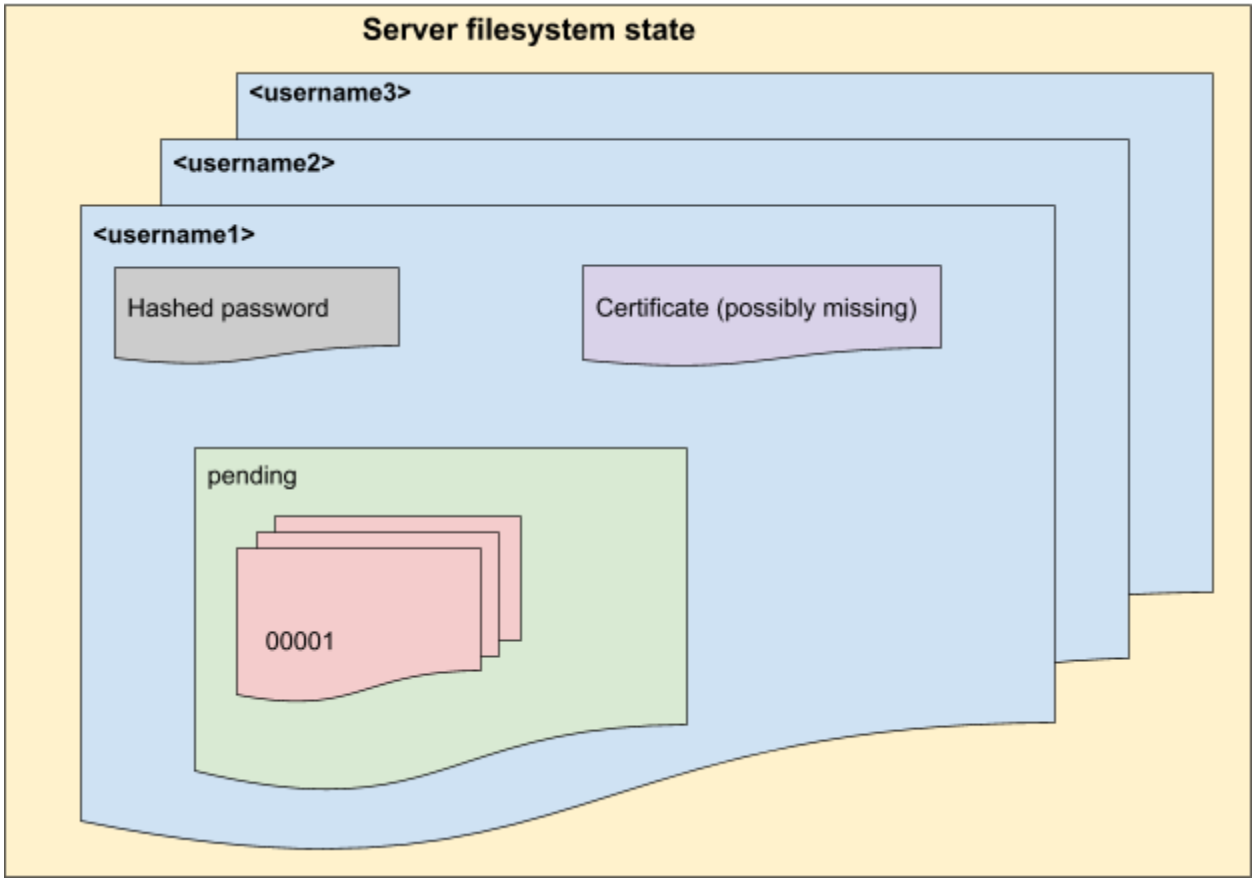
recvmsg

- User sends client side certificate to server
    - Server deletes single encrypted message from pending dir of user
    - Server sends back single encrypted message to client (the oldest message)
        - Client verifies signature via public key of sender
            - If verified, display message (decrypting using private key)
            - If not verified, do not display message (output error msg)



(source https://pagefault.blog/2019/04/22/how-to-sign-and-verify-using-openssl/)

Server Filesystem



Sandboxing

The server will be run inside of a docker sandbox, so that it does not unintentionally interfere with any other programs that may also be running on it's machine, possibly including the client.

Each user should only be able to receive and read their own messages, so each client will have a separate docker sandbox.

File Permissions

Within the server's docker file, we set permissions by creating one privileged user, **server,** who has access to the filesystem directory. The container will be executed as some unprivileged user e.g. polypose. The https_server program is setuid to server so that only https_client will be able to read and write to the user mailboxes in the filesystem.

Within the client's docker file, we set permissions by creating one privileged user, **client,** who has access to the tmp directory and write access to the certs directory. The container will be executed as some unprivileged user e.g. polypose. The https_client executable is setuid to client. The private key needs to be accessed in order to create a csr, which is why it is readable by the "unprivileged" user.

Test Plan

Each test script sets up the client and server from scratch and then cleans everything up at the end, returning the directory to the state it was in before the test case was run.

Test	**Functionality being tested**	**Steps to test**	**Expected Output (Pass)**
GETCERT TEST 0	correct password, cert for user doesn't exist	- call getcert with correct password	a new cert gets created for user
GETCERT TEST 1	correct password, cert for user already exists	-call getcert with correct password twice	a new certs gets created for the user that is different from the first cert
GETCERT TEST 2	incorrect password	-call getcert with incorrect password	client cert doesn't get created
GETCERT TEST 3	username not in filesystem (invalid user)	-call getcert with invalid user	client cert doesn't get created
GETCERT TEST 4	long username/password	-call getcert with a long username -call getcert with a long password	client cert doesn't get created
GETCERT TEST 5	username/password contains invalid characters	-call getcert with a username with invalid characters -call getcert with a password with invalid characters	client cert doesn't get created
CHANGEPW TEST 0	correct old password	-call getcert with correct password -call changepw with correct old password	a new certs gets created for the user that is different from the first cert created

			by getcert
CHANGEPW TEST 1	user has pending messages	-call getcert -call sendmsg to send message to user -call changepw with correct old password	password doesn't get updated, an error message stating that there are pending messages gets returned
CHANGEPW TEST 2	incorrect old password	-call changepw with incorrect old password	password doesn't get updated, an error message stating that the username and password don't match gets returned
SENDMSG TEST 0	valid client cert and valid recipient	-call getcert -call sendmsg to send a message with a valid client cert and a valid recipient	a message is written into the mailbox of the recipient (in filesystem/recipient/pending), the contents of the message match the message that was sent
SENDMSG TEST 1	invalid client cert	-call getcert -call sendmsg with an invalid client cert	a message isn't written to the mailbox of the recipient
SENDMSG TEST 2	the client cert file doesn't exist	-call getcert -call sendmsg with a client cert file that doesn't exist	a message isn't written to the mailbox of the recipient
SENDMSG TEST 3	all recipients are inavlid	-call getcert -call sendmsg with al invalid recipients	no message gets sent, an error stating that all recipients were invalid gets returned
SENDMSG TEST 4	valid recipient but the recipient doesn't have a certificate	-call getcert -call sendmsg with a recipient who doesn't have a certificate (the recipient hasn't called getcert yet)	a message isn't written to the mailbox of the recipient
RECVMSG TEST 0	valid client certificate	-call getcert -call sendmsg -call recvmsg to get the message that	a message gets received, and the message received matches the

		was just sent to the user	message sent
RECVMSG TEST 1	client certificate belongs to another client	-call getcert -call sendmsg -call recvmsg with a client cert that belongs to another user	a message doesn't get received (it's still in the mailbox of the user)
RECVMSG TEST 2	the file passed in as client cert isn't a certificate	-call getcert -call sendmsg -cal recvmsg with a file that's not a certificate	a message doesn't get received (it's still in the mailbox of the user)
SERVER TEST 0	invalid server cert	-start server with invalid server cert -call getcert	a certificate doesn't get created
SERVER TEST 1	invalid server private key	-start server with invalid private key -call getcert	a certificate doesn't get created
SERVER TEST 2	invalid server certificate chain file	-start server with invalid cert chain file -call getcert	a certificate doesn't get created
SANDBOXING & PERMISSIONS	accessing the client directories of other users (getting out of the sandbox), writing to the cert and keys directory/files in client, reading/writing to the filesystem mailbox directories or the messages in the pending directories in server	-try to cd .. inside the sandbox -check permissions by running ls -l -try to access directories/files you shouldn't have access to as outlined in the permissions above using cd and cat -try to write to the filesystem mailbox directories in the server by running 'echo "hi" > filesystem/username/pending'	can't get out of the sandbox, read read-protected directories/files or write to write-protected directories/files