Secure Channels

Christian Grothoff

Secure Channels

1. Install libgcrypt

GNU libgcrypt is a C library offering a wide range of cryptographic primitives.

1. `# apt install libgcrypt20-dev`
2. `# apt install gcc gdb valgrind emacs`
3. Download source templates (`exercise.tgz`) from course Git

2. Encryption

This exercise is about using symmetric encryption primitives.

- Use the provided `encrypt` and `decrypt` programs to encrypt "Hello world" text using AES256+GCM and then decrypt it.
- Study the `libgcrypt` documentation. Use it to switch the program to use AES256+CBC instead.
- Switch back to AES256+GCM. Extend the program to obtain, transmit and verify the authentication tag.
- Extend the program to authenticate additional plaintext data that is not at all encrypted.

3. Hashing

This exercise gives you the opportunity to actually use a hash function from code.

- Write a new program `hash.c` to compute the SHA-256 hash of the data read from `stdin`. Output the result in HEX and compare to `sha256sum`.
- Modify your program to use `SHA-512` instead.
- Write a new program `kdf.c` to compute the SCRYPT key derivation function. Output the result in HEX.

4. Performance

This exercise is supposed to give you some intuition about how fast various cryptographic constructions are.

- Modify your programs to perform 10000 iterations each time before generating any output.
- Measure the time the various operations take.
- Modify your programs to process 1 MB of input instead of the 11 bytes of "Hello world".
- Again, measure the time the various operations take.
- Change the IV length from 96 bits to 128 bits for AES256+GCM and measure again.