Dear All,

For this week, read on symmetric and asymmetric cryptography. Read,

a) pages 35-107 from Opplinger's open slides, 2018 edition (see books in CLASS1), and
b) for a more theoretical discussion, see Section 2.3 from Pfleeger's book.

For this week, we have already used (1) to

In addition, for the lab section,

A) install the lightweight ARM mbedtls package (see https://github.com/Mbed-TLS/mbedtls). Examine the code in the programs directory which provides the low-level primitives that help integrate security into applications (see README.md file for more). For example consider a) AES (e.g. standard encrypt/decrypt) and b) hash (e.g. for standard SHA3). This software is in the directories aes and hash in the programs subdirectory. Run the executables aescrypt2, hello and generic_sum to see how they work. For the API, you can read the code, or see the Doxygen manual for ARM mbedtls online (listed below).
   Also examine the following code snippets:
   a. symmetric cryptography implementations for confidentiality/integrity that use input strings, rather than files, by running the code in programs/benchmark.c. You can view or grep the code for the primitives. The preliminary Doxygen manual for ARM mbedtls is available from: https://tls.mbed.org/api/files.html
   b. simple operations on large numbers (encrypt/decrypt) by examining/running the code in mpi_demo.c
   c. rsa cryptography using the RSA PKCS#1 v1.5 algorithm, by examining/running the code in programs/pkey, for rsa_gen_key, rsa_encrypt/decrypt, rsa_sign/verify,
   d. Diffie-Hellman key exchange by examining/running the code in programs/pkey, for dh_server/client in two terminals. The video https://www.youtube.com/watch?v=YEBfamv-_do&t=138s explains the details very well.
   (Notice that to recompile your code (if you make changes) you need to type make in the main directory, i.e., above programs.)

B) Consider how keys and certificates are issued/revoked and used for application security (e.g., in SSH, TLS, Email). For certificates, try to understand the main functions from GPG (open-source PGP) to do key/certificate management.
   a. For instructions on how to install GPG (if there is no package available in your distro's package manager), see below:
      gnupg - https://gnupg.org (see https://gnupg.org/software/libraries.html for libraries to install first)
   b. Manuals available from: https://gnupg.org/documentation/manuals.html

If you have time, you can also read on SSH cryptography (e.g., how to login without password), Email (how certificates are imported and propagated from GPG to implement digital signatures for email authentications), and how Cryptography is used for Application Security. It is also

interesting to look into gnutls - https://www.gnutls.org Manuals are available from: https://gnutls.org/manual/gnutls.pdf

During the following week, we will discuss more on security protocols, embedded/IoT security and Cyber-Physical systems security where we can share a couple of demos. Also, you need to select a project related to these topics.

Miltos