7-4: RustCrypto practice

Artem Pavlov, TII, Abu Dhabi, 11.05.2024

Create new crate

- Create new branch in the repository p74
- Create new library crate p74
- Check that p74 is listed as a member of the workspace in the root Cargo.toml

File hashing using sha2

- Create two binaries in the bin/ folder:
  - rustcrypto_sha2: accepts algorithm string (e.g. “sha256”, “sha512”, etc.) and input file path. Reads the file content, hashes it using requested algorithm, and prints it as a hex-encoded string.
  - rustcrypto_sha3: accepts algorithm string (e.g. “sha3-512”, “shake128”, etc.), output size (if algorithm is “shake128” or “shake256”), and accepts input file path. Reads the file content, hashes it using requested algorithm, and prints it as a hex-encoded string.

File encryption using RustCrypto

- Create two binaries in the bin/ folder:
  - rustcrypto_siv: accepts “enc” or “dec”, input file path, output file path, and key as a hex-encoded string. Reads the input file, encrypts (if the first argument is “enc”) or decrypts (if the first argument is “dec”) it using the AES-128-SIV algorithm, and saves result to the output file
  - rustcrypto_cmac_ctr: accepts “enc” or “dec”, input file path, output file path, and key as a hex-encoded string. Reads the input file, encrypts (if the first argument is “enc”) or decrypts (if the first argument is “dec”) it using AES-128-CTR and AES-128-CMAC algorithms (use Encrypt-then-MAC), and saves result to the output file
- Use random nonces generated with getrandom
- Append generated nonces to encrypted files