CSEC 513: Lightweight Cryptography for the Internet of Things - Homework 2

Deadline: 18 January 2024 Thursday 09:40

In this homework, you are required to implement the 80-bit keyed version of PRESENT block cipher using **C programming language**. You can read the original PRESENT paper (should be available at ODTUCLASS) to determine the specifications of this cipher.

1. (50 points) Provide your implementation. You can verify it with the test vectors provided below
2. (30 points) Use you full name (in ASCII) as plaintext and 80-bit key and obtain the ciphertext using CBC mode and standard padding (If you have Turkish characters in your name, convert it to closest character in English). Use a random value as IV. For example:
   Cihangir Tezcan: 436968616e6769722054657a63616e
   Plaintext (blocks with padding): 436968616e676972 2054657a63616e80
   Key: 436968616e6769722054
3. (20 points) Check how many seconds it takes for your single CPU core to encrypt 64 MB of data ($2^{23} \times 64$ bits) (You can fix the plaintext P and the key k. Then perform key schedule algorithm for k (only once) and then encrypt P with the round keys in a for loop of size 2^{23}) Provide the number of seconds of this process with the model number of your CPU (e.g. 457 seconds on a single core of Intel i7-4700k).

```
Plaintext: 0000000000000000
Key: 00000000000000000000 (80-bit)
Round key  1: 0000000000000000 & Round Output  1: ffffffff00000000
Round key  2: c000000000000000 & Round Output  2: 80ff00ffff008000
Round key  3: 5000180000000001 & Round Output  3: 4036c837b7c88c09
Round key  4: 60000a0003000001 & Round Output  4: 73c2cd26b6192359
Round key  5: b0000c0001400062 & Round Output  5: 41d7be58531e4446
Round key  6: 900016000180002a & Round Output  6: 182ef861ad62fd1c
Round key  7: 0001920002c00033 & Round Output  7: 0ea0a5b67effc5a4
Round key  8: a000a0003240005b & Round Output  8: bba0b848a113e080
Round key  9: d000d4001400064c & Round Output  9: fa943423a9142338
Round key 10: 30017a001a800284 & Round Output 10: 69f2e22d63684d54
Round key 11: e01926002f400355 & Round Output 11: 548a4b63c330a59d
Round key 12: f00a1c0324c005ed & Round Output 12: d75f955fa228e4ca
Round key 13: 800d5e014380649e & Round Output 13: 44255864103841f9
Round key 14: 4017b001abc02876 & Round Output 14: e2cc9004363f6c12
Round key 15: 71926802f600357f & Round Output 15: c36682c5cd375421
Round key 16: 10a1ce324d005ec7 & Round Output 16: 597db55cc2a5d9b6
Round key 17: 20d5e21439c649a8 & Round Output 17: e67ce40e71b8b713
Round key 18: c17b041abc428730 & Round Output 18: 751df6d6807b5b59
Round key 19: c926b82f60835781 & Round Output 19: b948414e23332c93
Round key 20: 6a1cd924d705ec19 & Round Output 20: 5b75890dcfb3d563
Round key 21: bd5e0d439b249aea & Round Output 21: 5679203168278f5a
Round key 22: 07b077abc1a8736e & Round Output 22: 17c377c413fa45a3
Round key 23: 426ba0f60ef5783e & Round Output 23: 262a2de73b5f3ecd
Round key 24: 41cda84d741ec1d5 & Round Output 24: d3a053128b4d7bb3
Round key 25: f5e0e839b509ae8f & Round Output 25: 7db29209c28a20fa
Round key 26: 2b075ebc1d0736ad & Round Output 26: 62050c9940f400b9
Round key 27: 86ba2560ebd783ad & Round Output 27: 65d50da21fbcc09f
Round key 28: 8cdab0d744ac1d77 & Round Output 28: 6a50663c540d862f
Round key 29: 1e0eb19b561ae89b & Round Output 29: c79b8ff00a48df35
Round key 30: d075c3c1d6336acd & Round Output 30: 4a38c5e00283fba1
Round key 31: 8ba27a0eb8783ac9 & Round Output 31: 38d2f04c34635345
Round key 32: 6dab31744f41d700 & Ciphertext: 5579c1387b228445
```

WARNING: This homework is not about checking if you can Google and find PRESENT implementations at GitHub (which are generally wrong by the wy). Using implementations of other people as if it is your own is plagiarism and deserves disciplinary action !!!!