Security I Supervisions

Example Sheet 1*

Petar Veličković

Easter Term 2016

Cryptography

Most of the Cryptography questions presented here have been relevant when I did the course in 2013/14. If there's anything irrelevant feel free to skip it (but do let me know of such offenders, so I can remove them for the following years).

1. How many *functions* from n-bit strings to n-bit strings there are ($F : \{0,1\}^n \to \{0,1\}^n$)? How about *bijective functions* ($P : \{0,1\}^n \leftrightarrow \{0,1\}^n$)? Note that a block cipher is just a *tiny* subset of P.

2. How could you distinguish a Feistel cipher from a random function if it has only:
   a) one round?
   b) two rounds?

3. What is Kerckhoffs' principle?

4. What happens to the ciphertext block of DES if all the bits of the key and plaintext block are inverted?

5. Explain, using diagrams, how each of the block-cipher-based encryption schemes (ECB, CBC, CFB, OFB, CTR) encrypts and decrypts. Which of these modes can *parallelise* their encryption and/or decryption? Assuming a block size of 128 bits and that there are n ciphertext blocks in total, how many *plaintext bits* are expected to be flipped in each of these modes if one bit in the $\frac{n}{2}$th block is flipped.

6. A programmer wants to use CBC in order to protect both the integrity and confidentiality of network packets. She attaches a block of zero bits P_{n+1} to the end of the plaintext for redundancy, then encrypts with CBC. At the receiving end, she verifies that the added bits after CBC decryption are still all zero. Does this test ensure integrity of the transferred message?

*Special thanks to Markus Kuhn and Rubin Xu (from whose sheets this is heavily based on).

7. Reason about the effective key size of DESX and Triple DES respectively, taking meet-in-the-middle attacks into consideration.

8. In the CBC mode of operation, the IV is chosen uniformly at random, using a secure source of random bits. Show that CBC would not be CPA secure if the initial vector could be anticipated by the adversary (e.g. if it was coming from a counter or a timestamp).

9. Show that CTR mode is not CCA secure.

10. Prove that the basic CBC-MAC construction is not secure if a random IV is used, or if all the blocks are output.

11. Show that the encrypt-and-authenticate method on CBC-MAC is not CPA secure, or even indistinguishable for multiple messages and eavesdropping adversaries.

12. This question discusses certain elements of the Wired Equivalent Protocol (WEP) for wireless traffic encryption, which has earned the unlucky attribution of "how *not* to design a cryptographic protocol". Despite this fact, it is still implemented in millions of wireless routers regardless of its catastrophic security flaws. The basic setup involves a wireless client sharing a 40-bit key with the access point. When a connection is made, encryption is done using RC4-64, with the 64-bit key consisting of the 40-bit shared key and a 24-bit IV which is incremented for each packet. Within the packets, a CRC-32 checksum is appended prior to encryption. The value sent is therefore $IV|E_{K|IV}\{M|\text{CRC}_M\}$.
    a) Why is a new IV used with each packet? In practice, is 24 bits sufficient?
    b) Comment on the choice of 40-bit keys. Also comment on why allowing users to choose their own 40-bit hex values is dangerous.
    c) Why is the CRC included? Is it appropriate? Explain the attacks that are made possible due to the appended CRC information, and what an appropriate fix could be.
    d) The usage of many closely related keys in RC4 is insecure; there are *devastating* algebraic attacks on RC4 given reasonable amounts of known plaintexts encrypted under related keys. Explain where known plaintexts are likely to come from in this case.
    e) Finally, how *critical* is WEP security? Explain what attacks are possible on a laptop browsing the web through an insecure WEP channel.

13. You've been tasked with designing a data transmission protocol over a faulty, insecure and expensive channel (e.g. IP). You want to apply the following transforms to the data packets prior to being sent:
    - Encryption (using AES-CBC) to provide *confidentiality*;
    - Compression (using `gzip`) to increase *bandwidth efficiency*;
    - Calculation of an error-correcting code to ensure *reliability*.

The three operations can be applied to packets in any of the 6 possible orders. Is there a proper ordering? Which will cause problems (for each of those, indicate whether confidentiality, efficiency or reliability end up being jeopardised)?

Practical work

Security, and cryptography in particular, offers us space for some interesting practical exercises. These are, however, **optional** (the second one in particular).

1. You are given 11 ciphertexts of English text encrypted under the same one-time pad (or equivalently under a stream cipher/CTR mode with the same key). Your task is to mount a ciphertext only attack and decrypt the final ciphertext.

   [**Hint** (`rot13`): KBE gur pvcuregrkgf cnvejvfr, naq pbafvqre jung unccraf jura n fcnpr vf KBErq jvgu n punenpgre va [n-mN-M].]

   You may retrieve the ciphertexts at:
   http://www.cl.cam.ac.uk/~pv273/supervisions/SecI/many-time-pad.py

2. Write code, in the language of your choice, that solves the following problem:
   http://www.cl.cam.ac.uk/~pv273/supervisions/SecI/crypto.txt

   [**Hint 1** (`rot13`): Gurer ner ab xrlf zragvbarq, fb vg vf snve gb nffhzr gung gurer ner ab xrlf vaibyirq.]
   [**Hint 2** (`rot13`): Jungrire vf abg jevggra va gur (qrpelcgrq) ceboyrz fgngrzrag, lbh znl nffhzr vf nf trareny nf cbffvoyr.]