Introduction to Modern Cryptography
Flipped-Classroom Exercises about DES and AES

University of Amsterdam, Master of Logic, 2014
Lecturer: Christian Schaffner
TA: Malvin Gattinger

Thursday, 25 September 2014

Abstract

Thanks a lot for your (mostly) great and thoughtful questions! I've compiled two lists out of them, mostly eliminating duplicates and sometimes slightly changing the wording.

Content Questions

1. What is a side-channel attack?
2. Since DES is insecure against exhaustive search attacks, people are using triple-DES scheme even though it is three times slower than the DES. The key-length of double-DES would be enough to ensure security against exhaustive-search attacks. So why don't we use it?
3. What are the two essential features that you need to specify in an iterated block cipher?
4. What is an S box in DES? Why is S-box design crucial for the security?
5. Why should you never design nor implement block ciphers yourself (despite having followed this crypto course)?
6. Why does the AES instruction set provide two distinct instructions to perform one round of encryption given state and the round's key?
7. How long (and how much money) did it take for DES challenge to be solved?
8. Draw a Feistel network and describe the operations in one round with formulas.
9. Is a block cipher a PRG, PRF or PRP?
10. What is the construction (encode function) of DESX?
11. Prove that DES is not a random function.

Exercises

1. Show that for any cryptosystem that purely consists of matrix multiplication of some matrix A and the vector consisting of the message and the key, it is possible to recover the key from enough input-output pairs using the same key.
2. What is the time needed to find the key in DES (with probability 1) using the following two methods:
   (a) a straightforward exhaustive-search attack that does not take into account properties of the DES.
   (b) the complementary property of DES and a CPA attack. **Hint:** One can show that for every key k and input x, it holds that $DES_k(x) = \overline{DES_{\overline{k}}(\overline{x})}$, where $\overline{z}$ denotes the bitwise complement of z. Use a chosen-plaintext attack with two messages x and $\overline{x}$ to argue that it is possible to find the secret key in DES (with probability 1) using 2^{55} local computations of DES.
3. Let m be a message consisting of ℓ AES blocks, say $\ell = 100$. Alice encrypts m using CBC mode of AES and transmits the resulting ciphertext to Bob. Due to a network error, ciphertext block number $\ell/2$ is corrupted during transmission. All other ciphertext blocks are transmitted and

received correctly. Once Bob decrypts the received ciphertext, how many plaintext blocks will be corrupted? Would your answer change if Alice was using CTR mode?

Further, say that our encryption algorithm malfunctions at some stage (any of the operations AddRoundKey, SubBytes, ShiftRows, MixCoIumns, CBC, XOR). What is the latest stage of a malfunction for Bob to receive at least $\ell/2$ of the plaintext blocks?

4. In the actual construction of DES, the two halves of the output of the final round of the Feistel network are swapped. That is, if the output of the final round is (L_{16}, R_{16}) then the output of the cipher is in fact (R_{16}, L_{16}). Show that the only difference between the computation of DES^{-1} and DES (given the swapping of halves) is the order of subkeys.
5. What is the output of a 4-round Feistel Network when the input is (L_0, R_0) and each round function outputs all 1s, regardless of the input?
6. Why does DESX require outside *and* inside xor? More specifically, show that "outside xor": $E_1((k_1, k_2), m) := k_1 \oplus E(k_2, m)$ and "inside xor": $E_2((k_1, k_2), m) := E(k_2, m \oplus k_1)$ are as vulnerable as DES to exhaustive-search attacks.
7. In 2DES there is the problem of a 'meet-in-the-middle' attack. However, this attack requires to store a large (sorted) table that maps ciphertexts to keys. It is possible to only store a partial table, hoping that the key is in there and then run the attack. If it is not, construct a non-covered part of the table and try again. What is the space-runtime complexity tradeoff? Provide a function that takes N, the number of parts the table is split into and outputs the tuple (runtime complexity, space complexity). Can you think of a another procedure where such a tradeoff can be made?
8. Prove that the S_5-box of DES is not a linear function.

S_5		Middle 4 bits of input															
		0000	**0001**	**0010**	**0011**	**0100**	**0101**	**0110**	**0111**	**1000**	**1001**	**1010**	**1011**	**1100**	**1101**	**1110**	**1111**
Outer bits	**00**	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

9. Dan Boneh mentions in his lectures (on slide 37) that there is an easy attack on DESX in time 2^{120}. How does the attack work?
10. Given a closed physical system, the laws of physics limit the total amount of information the system is able to store. There is also a limit on how much computation can by performed by a system in a given amount of time, that is, a limit on the speed of information processing. Use an estimate of this limit (see `https://en.wikipedia.org/wiki/Bremermann's_limit`, `https://en.wikipedia.org/wiki/MargolusLevitin_theorem`, and the references on those pages) and an estimate of the amount of computation required to run an efficient AES-256 encoding algorithm to find a rough guess of the expected time to brute-force a AES-256 key if we convert the entire mass of the solar system into a computer operating at the ideal information processing limit. If the problem turns out not to be feasible even with our new solar system computer, how many star systems (galaxies, superclusters...) will we need to harvest and incorporate into our optimal computer before we can brute-force an AES-256 key? Alternatively, if our solar system brute-forcer doesn't take an unreasonable amount of time, how many AES keys could we find with it over the course of our voyage to a new, habitable solar system?