1 Examples Sheet Questions

1.1 Historic Ciphers

> **Ex 1.** Decipher the shift cipher text
> LUXDZNUAMNDODJUDTUZDGYQDLUXDGOJDCKDTKKJDOZ

There are two critical observations:

1. We do now know distribution of the source (although we can make guesses)
2. The key space is only 26 (25 excluding identity)

In light of these two observations, I brute force the decryption by writing a program which outputs all 26 candidate plaintext messages and by inspection select the one which is most likely.

```
["".join(chr((ord(c) + i - 65) % 26 + 65) for c in s) for i in range(26)]
```

Get a better listing environment (or get pdftex working locally)

By inspection, the most likely plaintext is FORXTHOUGHXIXDOXNOTXASKXFORXAIDXWEXNEEDXIT.

> **Ex 2.** How can you break the any transposition cipher with $\lceil \log_a n \rceil$ chosen plaintexts, if a is the size of the alphabet and n is the size of the permutation block length?

Send the base-a representation of each of each index. This requires $\lceil \log_a n \rceil$ messages. Then we can reinterpret the characters received at index j as integers to read off the index i which maps to it.

1.2 Perfect Secrecy

> **Ex 3.** Show that the shift cipher provides unconditional security if $\forall K \in \mathbb{Z}_{26} : \mathbb{P}(K) = 26^{-1}$ for plaintexts $M \in \mathbb{Z}_{26}$.

A cipher provides "perfect secrecy" if receiving ciphertext gives the interceptor *no information* about the plaintext, *even if they have infinite compute*. More formally, we have that:

$$\forall M \in \mathcal{M}, C \in \mathcal{C}. \mathbb{P}(M \mid C) = \mathbb{P}(M)$$

where

$$P(M \mid C) = \frac{\mathbb{P}(M) \cdot \sum_{\{K \mid M = \mathsf{Dec}_K(C)\}} \mathbb{P}(K)}{\sum_K \mathbb{P}(K) \cdot \mathbb{P}(\mathsf{Dec}_K(C))}$$

I prove that the LHS of this expression is equal to the RHS for a shift-cipher:

$$\begin{aligned}
P(M \mid C) &= \frac{\mathbb{P}(M) \cdot \sum_{\{K \mid M = \mathsf{Dec}_K(C)\}} \mathbb{P}(K)}{\sum_K \mathbb{P}(K) \cdot \mathbb{P}(\mathsf{Dec}_K(C))} && \text{Bayes Rule} \\
&= \frac{\mathbb{P}(M) \cdot \sum_{\{K \mid M = \mathsf{Dec}_K(C)\}} \frac{1}{26}}{\frac{1}{26} \cdot \sum_K \mathbb{P}(\mathsf{Dec}_K(C))} && \forall K \in \mathbb{Z}_{26} : \mathbb{P}(K) = 26^{-1} \\
&= \frac{\mathbb{P}(M) \cdot \frac{1}{26}}{\frac{1}{26} \cdot \sum_M \mathbb{P}(M)} && \text{shift cipher a bijection} \\
&= \frac{\mathbb{P}(M) \cdot \frac{1}{26}}{\frac{1}{26} \cdot 1} && \text{probabilities sum to 1} \\
&= \mathbb{P}(M) && \text{as required}
\end{aligned}$$

> **Ex 4.** Show that if an encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ over a message space $\mathcal{M}$ is *perfectly secret* if and only if

(*a*) for every probability distribution over $\mathcal{M}$, every message $M \in \mathcal{M}$, and every ciphertext $C \in \mathcal{C}$ with $\mathbb{P}(C) > 0$ we have

$$\mathbb{P}(C \mid M) = \mathbb{P}(C)$$

(*b*) for every probability distribution over $\mathcal{M}$, every message pair $M_0, M_1 \in \mathcal{M}$, and every ciphertext $C \in \mathcal{C}$ with $\mathbb{P}(C) > 0$ we have

$$\mathbb{P}(C \mid M_0) = \mathbb{P}(C \mid M_1)$$

(*a*) This is proved by a simple rearrangement of the condition for perfect secrecy:

$$\begin{aligned}
\forall M \in \mathcal{M}, C \in \mathcal{C}.\, \mathbb{P}(M \mid C) = \mathbb{P}(M) && \text{definition of perfect secrecy} \\
\forall M \in \mathcal{M}, C \in \mathcal{C}.\, \frac{\mathbb{P}(C \mid M) \cdot \mathbb{P}(M)}{\mathbb{P}(C)} = \mathbb{P}(M) && \text{Bayes Rule} \\
\forall M \in \mathcal{M}, C \in \mathcal{C}.\, \mathbb{P}(C \mid M) \cdot \mathbb{P}(M) = \mathbb{P}(C) \cdot \mathbb{P}(M) && \text{multiplying by } \mathbb{P}(C) \\
\forall M \in \mathcal{M}, C \in \mathcal{C}.\, \mathbb{P}(C \mid M) = \mathbb{P}(C) && \text{dividing by } \mathbb{P}(M)
\end{aligned}$$

(*b*) This is proved by direct application of the above property:

$$\begin{aligned}
\forall M, C.\, \mathbb{P}(C \mid M) = \mathbb{P}(C) && \text{proved above} \\
\forall C.\, \mathbb{P}(C \mid M_1) = \mathbb{P}(C) && \text{universal generalisation} \\
\forall C.\, \mathbb{P}(C \mid M_0) = \mathbb{P}(C) && \text{universal generalisation} \\
\forall C.\, \mathbb{P}(C \mid M_0) = \mathbb{P}(C \mid M_1) && \text{transitivity of } =
\end{aligned}$$

1.3 Semantic Security

1.4 Block Ciphers

Ex 5. If the round function f in a Feistel construction is a pseudo-random function, how many rounds r are at least necessary to build a pseudo-random permutation? What test can you apply to distinguish a Feistel structure with $r-1$ rounds (with high probability) from a random permutation?

Do this!

Ex 6. Using a given pseudo-random function $F : \{0,1\}^{100} \to \{0,1\}^{100}$, construct a pseudo-random permutation $P : \{0,1\}^{300} \to \{0,1\}^{300}$ by extending the Feistel princigle appropriately.

Do this!

Ex 7. What happens to the ciphertext block if all bits in both the key and plaintext block of DES are inverted?

Do this!

Ex 8. given a hardware implementation of the DES encryption function, what has to be modified to make it decrypt?

Do this!

1.5 Modes of Operation

Ex 9. In the CBC mode of operation, the initial vector (IV) chosen uniformly at random, using a secure source of random bits. Show that the CBC would not be CPA secure if the initial vector could be anticipated by the adversary, for example because it is generated instaed of using a counter or a time-stamp.

Do this!

Ex 10. Explain for each of the discused modes of operation (ECB, CBC, CFB, OFB, CTR) of a block cipher how decryption works.

Do this!

Ex 11. A sequence of plaintext blocks $M_1, \ldots, M_8$ is encrypted using DES into a sequence of ciphertext blocks. Where an IV is used, it is numbered C_0. A transmission error occurs and one bit in ciphertext block C_3 changes its value. As a consequence, the receiver obtains after decryption a corrupted plaintext block sequence $M'_1, \ldots, M'_8$. For the discused modes of operation (ECB, CBC, CFB, OFB, CTR), how many bits do you expect to be wrong in each block M''_i?

Do this!

Ex 12. Your opponent has invented a new stream-cipher mode of operation for 128-bit key AES. He thinks that OFB could be improved by feeding back into the key port rather than the data prot of the ASE chip. He therefore sets $R_0 = K$ and generates the key stream by $R_{i+1} = E_{R_i}(R_0)$. Is this better or worse than OFB?

Do this!

Ex 13. A programmer wants to use CBC in order to protect both the integrity and confidentiality of network packets. She attaches a blcok of zero bits M_{n+1} to the end of the plaintext $M_1 \parallel \ldots \parallel M_n$ as redundancy, then encrypts with CBC. At the receiving end, she verifes that the added redundant bits are still all zero after CBC decryption. Does this test ensure the integrity of the transferred message?

Do this!

1.6 Message Authenticity

Ex 14. Show that CTR mode is not CCA secure.

Do this!

1.7 Authenticated Encryption

Ex 15. Your colleagues have inented a new authenticated encryption scheme that they call AES-CBC+CMAC. Their key generating function outputs a 128-bit AES key K, adn their encryption function outputs $C \parallel T = \mathsf{Enc}_C(M) \parallel \mathsf{Mac}_C(M)$, where $\mathsf{Enc}_K(M)$ shall be the AES-CBC encryption of M with key K (with random IV each time), and $\mathsf{Mac}_C(M)$ shall be the AES-CMAC of M with key K. Show that this construct lacks CPA security.

Do this!

1.8 Secure Hash Functions

Ex 16. Explain the purpose of the collision-resistance requirement for the hash function used in a digital signature scheme.

Do this!

Ex 17. Your colleagues urgently need a collision-resistant hash function. Their code contains already an existing implementation of ECBC-MAC, using a block cipher with 256-bit block size. Therefore, they suggest to use ECBC-MAC with fixed keys $K_1 =$

$K_2 = 0^\ell$ as a hash function. Show that this construction is not even pre-image resistant.

Do this!

Ex 18. Show how the DES block cipher can be used to build a 64-bit hash function. How difficult is it to find collisions for your construct?

Do this!

2 2014 Paper 8 Question 11

https://www.cl.cam.ac.uk/teaching/exams/pastpapers/y2014p8q11.pdf

(*a*) Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a private-key encryption scheme that operates on fixed-length messages from $\mathcal{M} = \{0,1\}^m$. Briefly explain a game that a user $\mathcal{U}$ of Π must be ble to win against any polynomial-time adversary $\mathcal{A}$ with probability $\frac{1}{2} - \epsilon$ (where ϵ is "negligible" with growing key length) for Π to be able to claim to offer "indistinguishable multiple encryptions under chosen-plaintext attack" (CPA security).

Do this!

(*b*) Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a private-key encryption scheme that offers CPA security and operates on fixed-length messages $M \in \mathcal{M} = \{0,1\}^m$ with keys $K \in \mathcal{K} = \{0,1\}^\ell$. We use it to construct a new encryption scheme $\Pi' = (\mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}')$. In which of the following cases is Π' also CPA secure? Explain your answer.

(*i*) $\mathsf{Enc}'_K(M) = \mathsf{Enc}_K(M \oplus 1^m)$

Do this!

(*ii*) $\mathsf{Enc}'_K(M) = \mathsf{Enc}_K(M) \parallel \mathrm{LSB}(M)$

Do this!

(*iii*) $\mathsf{Enc}'_K(M) = \mathsf{Enc}_K(M) \parallel \mathrm{LSB}(K)$

Do this!

(*iv*) $\mathsf{Enc}'_K(M) = \mathsf{Enc}_{0^\ell}(M)$

Do this!

(*c*) While reviewing an implementation of AES-CBC, you discover that it simply uses the last ciphertext block from the previously encrypted message as the IV value C_0 for encrypting the message. The implementation's author argues that as long as the IV of the very first message was chosen uniformly at random, all resulting subsequent ciphertext blocks will also be distributed uniformly at random, and therefore make good IVs. Why is this construction nevertheless not CPA secure?

Do this!

https://www.cl.cam.ac.uk/teaching/exams/pastpapers/y2011p4q8.pdf

3 2011 Paper 4 Question 8

(*b*) Your colleagues used a pseudo-random function $f : \{0,1\}^{64} \to \{0,1\}^{64}$ in order to construct a permutation $g : \{0,1\}^{192} \to \{0,1\}^{192}$. The argument and return values of g are split into three 64-bit registers, respectively: $g(X_1, X_2, X_3) = (Y_1, Y_2, Y_3)$. The output of g is calculated as $Y_2 = f(X_1) \oplus X_2 \oplus f(X_3)$, $Y_1 = X_1 \oplus f(Y_2)$, and $Y_3 = X_3 \oplus f(Y_2)$, where $\oplus$ denotes bit-wise exclusive or.

(*i*) Show that g is indeed a permutation.

Do this!

(*ii*) Show how an attacker who does not know f can efficiently distinguish g from most random permutations, after evaluating g on two different inputs.

Do this!

(*iii*) After you point out this shortcoming to your colleagues, they propose an improved variant $g'(X_1, X_2, X_3) = (Z_1, Z_2, Z_3)$ that adds another round to g: $Z_1 = Y_1$, $Z_2 = f(Y_1) \oplus Y_2 \oplus f(Y_3)$, and $Z_3 = Y_3$.
Show how this variant still does not fix the problem of efficient distinguishability from most random permutations.

Do this!