Cryptography Notes

Rasmus Kirk Jakobsen Mikkel Skafsgaard Berg

December 15, 2023

Note that these notes are based on the 2023v3 version of the cryptography book.

Contents

Curriculum

Background material:

Chapters 2+3 (preliminary probability theory and math), Section 5.4 (optimistic results on key exchange), Section 6.3 (diff and lin analysis), Proof of Theorem 7.8 (if you can get the secret exponent you can factor), Section 7.6.1 (factoring algorithms), Section 9.6 (discrete log algorithms).

Example Templates:

Here are some details on what you might for instance cover in each the exam subject. But do not misunderstand this in the sense that you have to follow the templates below, they really are just examples…

Information theory and Cryptography:

Definition of perfect secret security, why you need as many keys at plaintexts to have perfect security. Definition of entropy, and proof of some of the inequalities properties it satisfies. Unicity distance (but be careful, this may take a lot of time, so test this beforehand)

Symmetric (secret-key) cryptography:

What a crypto-system is (the three algorithms) You can describe DES or AES - but you can also just give a high-level description of what a block cipher is. Definitions of PRF and CPA security. Specification of CBC or CTR modes (or both), proofs of CPA security for CBC or CTR mode (or both). Perhaps a brief talk about stream ciphers and how to make one from a block cipher.

Public-key cryptography from Factoring:

What a public-key cryptosystem is. Basic spec of RSA, maybe proof that decryption works. Then some selection of the following: How to make RSA be CPA secure (the PCRSA scheme, and the result that computing the least significant bit is as hard as inverting RSA). How to generate keys and Miller-Rabin primality testing, how to get CCA security: OAEP and the intuition on why it works.

Public-key cryptography based on discrete log and LWE:

The DL, DH and DDH problems, and how they relate. The El Gamal cryptosystem and proof that it is secure if DDH is hard. Then some example of groups we can use, can be a subgroup of $\mathbb{Z}_p^*$, or you can talk about elliptic curves. You can also put less emphasis on El Gamal, for instance skip the example groups and go to LWE instead, define the problem and the cryptosystem and do the proof from the exercise that decryption works under a certain assumption about the noise distribution.

Symmetric authentication and hash functions:

Definition of collision-intractable hash functions. Then a selection of: construction from discrete log, proof that collision-intractable implies one-way, construction and proof that we can get any size input from fixed size input. Finally, MAC schemes, definition of CMA security, CBCMAC and EMAC security result for EMAC. Maybe a brief mention of HMAC.

Signature schemes:

Definition of signatures schemes and of CMA security. The Schnorr signature scheme, you can do many details here, such as the proof that you cannot cheat the underlying interactive game with better than 1/q probability, and the full story on how you derive the signature scheme from the interactive game. Or you can just do the spec of the scheme, giving you time for something else, such as RSA+hash signatures and the proof that secure hash + secure signature scheme is secure. Or you can do the one-time signatures based on hash functions and the proof that they are secure.

Basic Facts from Probability Theory

Theorem 2.4 (Jensen's inequality): Let $p_1, ..., p_n$ be a probability distribution, that is, $\sum_i p_i = 1$ and $0 \leq p_i \leq 1$. Then for any concave f and any $x_1, .., x_n$, we have:

$$\sum_{i=1}^{n} p_i f(x_i) \leq f(\sum_{i=1}^{n} p_i x_i)$$

Furthermore, if f is strictly concave, equality holds **iff** all the x_i's are equal.
TLDR: $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$

Note: *Concave* means that $\frac{f(a)+f(b)}{2} \leq \frac{a+b}{2}$.
TLDR: $f'' = 0$.

Note: *Strictly concave* means that $\frac{f(a)+f(b)}{2} < \frac{a+b}{2}$ for all $a \neq b$.
TLDR: This basically means that graph of the function always curves and is never linear.

Note: For this course, we only apply this to log() which is strictly concave. △

Information theory and Cryptography (Chapter 5)

Disposition (Kirk):

- Perfect Security
- Entropy
- Unicity Distance

Disposition (Berg):

Perfect Security

Below we have the definition for perfect security:

Definition 5.1: A cryptosystem has perfect security if for all $x \in \mathcal{P}$ and $y \in \mathcal{C}$, it holds that $P[x|y] = P[x]$.
TLDR: Information about the ciphertext gives you *no* information about the plaintext. △

Theorem - $|\mathcal{K}| \geq |\mathcal{C}| \geq |\mathcal{P}|$: If you have perfect security then $|\mathcal{K}| \geq |\mathcal{C}| \geq |\mathcal{P}|$.
TLDR: If you have perfect security your key can not be shorter than your ciphertext, which cannot be shorter than your plaintext. △

Proof:

- $|\mathcal{C}| \geq |\mathcal{P}|$: This is true for all crypto systems in order for decryption to function correctly.
- $|\mathcal{K}| \geq |\mathcal{C}|$: For a fixed plaintext x must be able to hit every ciphertext y, otherwise an adversary could conclude that $E(x) \neq y$ and therefore learn information from y.

Therefore, given perfect security, you have $|\mathcal{K}| \geq |\mathcal{C}| \geq |\mathcal{P}|$. □

Entropy

> *As a further illustration of the intuition behind entropy, consider the following thought experiment: suppose you get access to an oracle that will magically tell you if you will live to be more than 110 years old. One would naturally expect that the probability p of getting "yes" as the answer is very small, while the probability* $1 - p$ *of "no" is close to 1.*

Definition 5.6: Let X be a random variable that takes values $x_1, ..., x_n$ with probabilities $p_1, ..., p_n$. Then the entropy of X, written $H(X)$, is defined to be:

$$H(X) = \sum_{i=1}^{n} p_i \log_2(1/p_i)$$

TLDR: If an event A occurs with probability p and you are told that A occurred, then you have learned $\log_2(1/p)$ bits of information. △

The entropy $H(X)$ can be described as:

- How many bits we need to send on average to communicate the value of X.
- The amount of uncertainty you have about X before you are told what the value is.

Theorem 5.7: For a random variable X taking n possible values, it holds that $0 \leq H(X) \leq \log_2(n)$. Furthermore, $H(X) = 0$ **iff** one value X has probability 1 (and the others 0). $H(X) = log_2(n)$ **iff** it is uniformly distributed, i.e., all probabilities are $1/n$.
TLDR: If the entropy of X is 0 there is no uncertainty, meaning that we know the value of X. If the entropy of X is 1 then the uncertainty of X is highest meaning that all possible values of X have the same probability. △

Proof: We need to prove the following:

- $H(X) > 0$
  - $H(X)$ is defined as a product of positive sums, therefore $H(X)$ is also positive.
- $H(X) = 0$ **iff** a single $p_i = 1$ and all other $p_j = 0$
  - The function $f(p) = p\log(1/p)$ is only 0 if $p = 0 \vee p = 1$. This coupled with the fact that probabilities must sum up to one means that $H(X) = 0$ **iff** a single $p_i = 1$ and all other $p_j = 0$.
- $H(X) < \log_2(n)$
  - log is a concave function ($\log'' = 0$) therefore we can use Theorem 2.4 (Jensen's inequality):

$$H(X) = \sum_{i=1}^{n} p_i \log_2(1/p_i) \leq \log_2(\sum_{i=1}^{n} p_i \cdot 1/p_i) = \log_2(n)$$

    Thus $H(X) < \log_2(n)$.
- $H(X) = \log_2(n)$ **iff** X is uniformly distributed.
  - Since log_2 is strictly concave then from Theorem 2.4 we know that $H(X) = \log_2(n)$ **iff** all p_i are equal i.e. X is uniformly distributed. □

Conditional Entropy

Definition 5.9: Given the above definition of $H(X \mid Y = y_j)$, we define the conditional entropy of X given Y to be:

$$H(X \mid Y) = \sum_j P[Y = y_j] H(X \mid Y = y_j)$$

Entropy of Random Variables in Cryptography

Theorem 5.11: For any cryptosystem with deterministic encryption function, it holds that:

$$H(K \mid C) = H(K) + H(P) - H(C)$$

TLDR: Answers how much uncertainty remains about the key given the ciphertext

Unicity Distance

Definition - Redundancy: Given a language L and a plaintext space $\mathcal{P}$, the *redundancy* of the language is the amount of superflous information is contained, on avarage in the language L.

$$R_L = \frac{\log(|\mathcal{P}|) - H_L}{\log(|\mathcal{P}|)} = 1 - \frac{H_L}{\log(|\mathcal{P}|)}$$

H_L is a measure of the number of bits of information each letter contains in the language L, on average. For English, we have that H_L is (very approximately) 1.25 bits per letter.

$$H_L = \lim_{n \mapsto \infty} H(P_n)/n$$

TLDR: A language contains redundancy, which is how much duplicate information there is on avarage in the language.
Example: The following sentance displays redundancy in english:

> *"cn y rd th fllwng sntnc, vn f t s wrttn wtht vcls?"*

△

Definition - Spurious Keys: If an adversary has a ciphertext y that he wants to decrypt, he can try all keys and see if y decrypts to meaningful english. If y decrypts to meaningful english under the *wrong* key, then that key is said to be a *spurious key.*
TLDR: A spurious key is a key that *seems* to be the correct key for a ciphertext but is not. △

Definition - Number of Spurious Keys: The average number of spurious keys, taken over all choices of ciphertexts of length n:

$$sp_n = \sum_{\boldsymbol{y} \in \mathcal{C}^n} P[y](|K(\boldsymbol{y})| - 1) = \sum_{\boldsymbol{y} \in \mathcal{C}^n} P[y]|K(\boldsymbol{y})| - 1$$

Given a ciphertext $\boldsymbol{y}$, we use $K(\boldsymbol{y})$ to denote the set of keys that are possible given this ciphertext. More precisely, a key K is in this set if decryption of $\boldsymbol{y}$ under K yields a plaintext that could occur with non-zero probability:

$$K(\boldsymbol{y}) = \{K \in \mathcal{K} \mid P[D_K(\boldsymbol{y} > 0)]\}$$

TLDR: This formula for sp_n describes the average number of spurious keys of a ciphertext $\boldsymbol{y}$ of length n.

△

Definition 5.12: The unicity distance n_0 of a cryptosystem is the minimal length of plaintexts such that $spn_0 = 0$, if such a value exists, and ∞ otherwise.
TLDR: The unicity distance tells you how many times you can encrypt something where multiple keys seem to be valid keys. △

Theorem 5.13: Assume we have a cryptosystem with deterministic encryption function, where the plaintext and ciphertext alphabets have the same size ($|\mathcal{C}| = |\mathcal{P}|$), and where keys are uniformly chosen from $\mathcal{K}$. Assume we use the system to encrypt sequences of letters from language L. Then

$$n_0 \geq \frac{\log(|\mathcal{K}|)}{R_L \log(|\mathcal{P}|)}$$

TLDR: If we reuse keys, our unconditional security will always be gone, once we encrypt enough plaintext under the same key. The only exception is the case where $R_L = 0$ which leads to n_0 being ∞. Which makes sense, if every sequence of characters is a plaintext that can occur, the adversary can never exclude a key. △

Proof: We start by unfolding the definition of $H(K \mid C_n)$ using Definition 5.9:

$$H(K \mid C_n) = \sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{C}_n}} P[C_n = \boldsymbol{y}] H(K \mid C_n = \boldsymbol{y})$$

First, note that given some ciphertext $\boldsymbol{y}$, the key K will have some conditional distribution, but of course only values in $K(\boldsymbol{y})$ can occur. Therefore $H(K|C_n = \boldsymbol{y}) \leq \log_2(|K(\boldsymbol{y})|)$:

$$\begin{aligned}
H(K \mid C_n) &\leq \sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{C}_n}} P[C_n = \boldsymbol{y}] \log_2(|K(\boldsymbol{y})|) \\
&\leq \log_2 \left(\sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{C}_n}} P[C_n = \boldsymbol{y}] |K(\boldsymbol{y})| \right) && \text{(Definition 2.4 - Jensen's Inequality)} \\
&\leq \log_2(sp_n + 1) && \text{(Definition - Number of Spurious Keys)}
\end{aligned}$$

Now we want to simplify $H(K \mid C_n)$. We start by applying Theorem 5.11:

$$H(K \mid C_n) = H(K) + H(P_n) - H(C_n)$$

Observe that $H(C_n) \geq log(|C|^n) = n\log(|\mathcal{P}|)$. Moreover, recalling the definition on H_L, let us assume that we take n large enough so that $H(Pn) \approx nH_L$.

$$\begin{aligned} H(P_n) &\approx nH_L \\ &\approx n(\log(|\mathcal{P}|)(1 - R_L)) \quad \text{(Definition - Number of Spurious Keys)} \end{aligned}$$

Now we try to find $H(K \mid C_n)$

$$\begin{aligned} H(K \mid C_n) &= H(K) + H(P_n) - H(C_n) \\ &\geq H(K) + H(P_n) - n\log(|\mathcal{P}|) && \text{(From our observation of } H(C_n)\text{)} \\ &\approx H(K) + n\log(|\mathcal{P}|)(1 - R_L) - n\log(|\mathcal{P}|) && \text{(From our estimate of } H(P_n)\text{)} \\ &= H(K) + n\log(|\mathcal{P}|) - n\log(|\mathcal{P}|)R_L - n\log(|\mathcal{P}|) \\ &= H(K) - n\log(|\mathcal{P}|)R_L \\ &= \log(|\mathcal{K}|) - n\log(|\mathcal{P}|)R_L && \text{(Theorem 5.7, } K \text{ is uniform)} \\ H(K \mid C_n) &\geq \log(|\mathcal{K}|) - n\log(|\mathcal{P}|)R_L \end{aligned}$$

Combining our equations, setting $sp_n = 0$ and solving for n:

$$\begin{aligned} \log(|\mathcal{K}|) - n\log(|\mathcal{P}|)R_L &\leq \log_2(sp_n + 1) \\ \log(|\mathcal{K}|) - n\log(|\mathcal{P}|)R_L &\leq \log_2(0 + 1) \\ n\log(|\mathcal{P}|)R_L &\leq \log(|\mathcal{K}|) \\ n &\leq \frac{\log(|\mathcal{K}|)}{\log(|\mathcal{P}|)R_L} \end{aligned}$$

So $n_0 \leq \frac{\log(|\mathcal{K}|)}{\log(|P|)R_L}$. □

Symmetric (secret-key) cryptography (Chapter 4.1 + 6)

Disposition (Kirk)

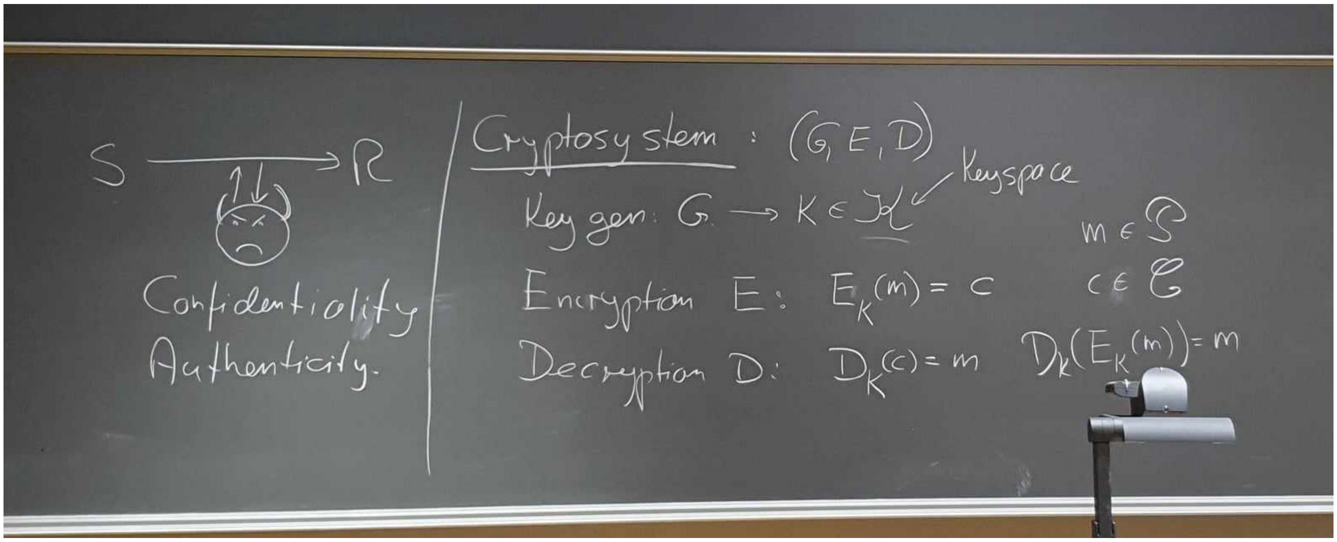
- Definitely AES
- PRF and CPA security
- CBC/CTR
- CPA security proof for CBC/CTR
- At least understand the block cipher…

Disposition (Berg)

- Symmetric Cryptosystem definition (chap 4.1)
- DES definition
  - Feistel ciphers
- AES definition
- (High level definition of Block Ciphers)
- PRF security
- CPA security
- CBC mode
- CTR mode
- Stream ciphers
- (Differential & Linear Cryptanalysis)

Notes

Symmetric Cryptosystems


$S \longrightarrow R$
Confidentiality
Authenticity.
Cryptosystem : (G, E, D)
Keyspace
Key gen: $G \longrightarrow K \in \mathcal{K}$
$m \in \mathcal{P}$
Encryption E: $E_K(m) = c$
$c \in \mathcal{C}$
Decryption D: $D_K(c) = m$
$D_K(E_K(m)) = m$

(Optionally something related to confidentiality and authenticity)

For a symmetric system, there are 3 finite sets given; the key space κ, the plaintext space ρ and the ciphertext space $\mathcal{C}$.

Public-key cryptography from Factoring (Chapter 7 & 8)

Disposition (Kirk)

Disposition (Berg)

Notes

Public-key cryptography based on discrete log and LWE (Chapter 9 & 10, definition of CPS security in chapter 8)

Disposition (Kirk)

Disposition (Berg)

Notes

Symmetric authentication and hash functions (Chapter 11)

Disposition (Kirk)

Disposition (Berg)

Notes

Signature schemes (Chapter 12)

Disposition (Kirk)

Disposition (Berg)

Notes

Appendix

CPA

$$\begin{aligned}\epsilon &= \epsilon' + \left(\frac{\mu}{n}\right)^2 \cdot \frac{1}{2^n} \\ &= \epsilon' + \frac{\mu^2}{n \cdot 2^n}\end{aligned}$$

Solving for $1 = |\epsilon - \epsilon'|$:

$$\begin{aligned}1 &= |\epsilon - \epsilon'| \\ 1 &= \frac{\mu^2}{n \cdot 2^n} \\ n \cdot 2^n &= \mu^2 \\ \sqrt{n \cdot 2^n} &= \mu \\ \sqrt{n} \cdot 2^{n/2} &= \mu\end{aligned}$$

So if we encrypt much less than $2^{n/2}$ we are safe. We disard $\sqrt{n}$ since it is insignificant compared to $2^{n/2}$.

□

CPA

$$\begin{aligned}P[M_j] &= P[M_j|M_{j-1}]P[M_{j-1}] + P[M_j|\neg M_{j-1}]P[\neg M_{j-1}] && \text{(Law of Total Probability)} \\ &= \frac{P[M_j, M_{j-1}]}{P[M_{j-1}]}P[M_{j-1}] + P[M_j|\neg M_{j-1}]P[\neg M_{j-1}] && \text{(Bayes rule)} \\ &= P[M_j, M_{j-1}] + P[M_j|\neg M_{j-1}]P[\neg M_{j-1}] \\ &= P[M_j]P[M_{j-1}] + P[M_j|\neg M_{j-1}]P[\neg M_{j-1}] \\ &\leq P[M_{j-1}] + P[M_j|\neg M_{j-1}]\end{aligned}$$