GDPR (General Data Protection Regulation)

Omhandler Sensitive Data Exposure

Verden er inddelt I 3 dele, USA hvor data er en handelsvare, Kina hvor staten kontrollere data, EU hvor data reguleres.

Regler for hvordan man skal overholde dataregler.

Sikre sig for at private data ikke kommer ud i verdenen.

Hvem man skal informere hvis det går galt.

Dækker alle firmaer der arbejder med data i eller med EU er dækket af GDPR.

I EU har man som data subject rettigheder:

* Breach notifications are made mandatory (72 hours).
* Right to access.
* Right to be forgotten.
* Data portability in a ‘commonly use and machine reliable format’.

GDPR – penalties

* Up to 4% of annual global turnover.
* E20.000.000.
* Whichever is greater!
* Revenue of Facebook was $40.654.000.000 in 2017, so turnover was much higher.
* 4% of revenue is E1.391.738.381.

GDPR – Privacy by default

* The condition for consent has been strengthened.
* It must be as easy to withdraw consent.

GDPR – Privacy by design

* In its core a system should use a little private data as possible.
* Controllers should only process the absolutely necessary data and limit the access to personal data.

GDPR – Data Protection Officers

* Must be appointed on the basis of professional qualifications.
* May be a staff member or an external service provider (Outsource).
* Contact details must be provided to the relevant DPA (Data Protection Agency, Datatilsynet).
* Must be provided appropriate resources to carry out tasks and maintain der expert knowledge.
* Must report directly to the highest level of management.
* Must not carry out other tasks that may result in a conflict of interest.

GDPR – What is personal data

* Personal data is any information that relates to an identified or identifiable living individual.
* For example: Name and surname, address, email, card numbers, location data, IP, cookie ID, data held by other parties that identify the person.

GDPR – Data Controller

* The data processor processes personal data only on behalf of the controller.
* The data processor is usually a third-party external company.
* E.g. A payroll company, IT service companies.

A3 Sensitive Data Exposure (Kryptering)

At man kommer til at afsløre data som man ikke skulle have afsløret.

Hvordan man krypterer data

Encrypting your information

* Is like putting your valuables in a safe.
* It is a second layer in case your house is broken into (If someone gets the encrypted data, they hopefully can’t use it or understand what it is).

Base64 encoding

* Fungere ved at man tager 6 bit og laver dem om til et tegn, så tager man de næste 6 bit osv…
* We often use binary information as characters (e.g. emails with attachments)
* INGEN encryption ingår, base64 er beregnet til komprimering af data størrelse.
* DETTE ER IKKE EN ENCRYPTION

Cesar rot code

* An old method potentialy used by cesar where you rotate the aplphabet by a number of letters, 3 in this case so that A becomes D ovs…

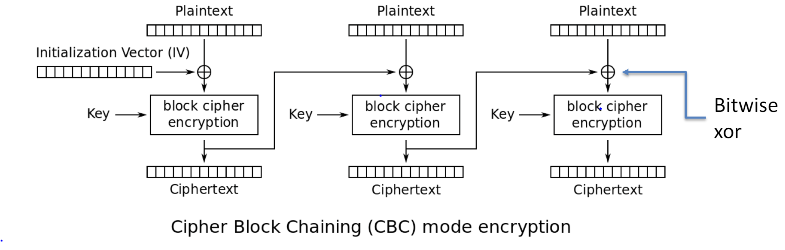
Substitution ciphers

* Simple substitution
* Polyaplhabetic substitution (more alphabets)
* Polygrapic substitution (larger numbe of characters, blev brugt af tyskerne i WW2 ved ENIGMA)
* The one-time pad (hvis man skifter seperationen mellem substitutions vilkårligt efter hver bogtav, stort set umulig at bryde)

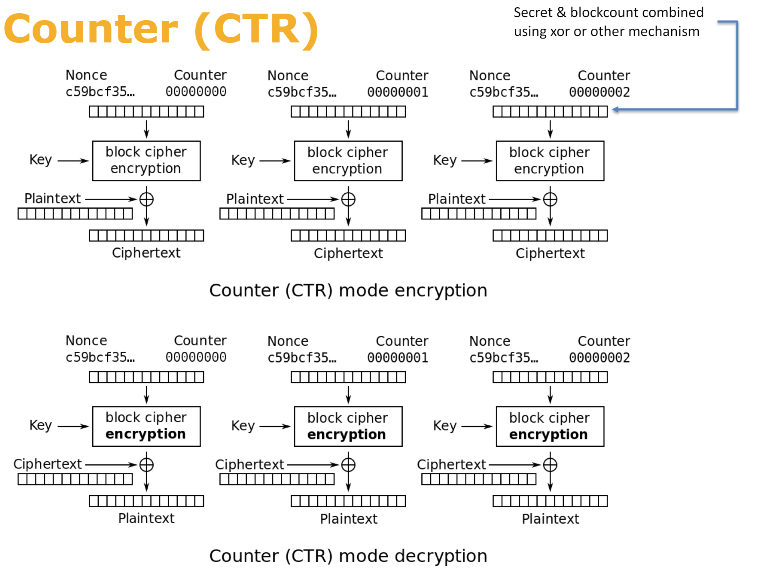
EAS algorithm

* 
* The heart of the algorithm can “only” encrypt blocks which are 128, 196, 256 bits long (16,24or32 bytes)
* To make it work on real data, we need to break plain text into blocks and ecrypt each block.

Cipher Block Chaining

* 
* Hvis man tager den samme nøgle og XOR 2 gange får man den oprindelige text.
* Hvis du ændrer en bit men XOR ændres der mange steder i Ciphertxten

Counter (CTR)

* En hash går kun en vej, derfor sikre vi os at den samme hash giver det samme output.
* Her har man ikke brug fro en decrypter da man kan counter encrypte sig vej tilbage til plaintext.
* 
* Three inputs: a Plaintext, Key, b Nonce (NumberOnce, generated by random generator and saved to disk, Don’t push to github)
* One output: c Chiphertext
* a XOR b = c, c XOR b = a

Exercises

* Exercise 0. Design a system for handling signatures for new parties.
  + You can sign up for at most one party  
      
    In correspondence to the GDPR rules you could design a system where each you can sight up your signature for a party through you CPR-number, protected by nem-id so CPR-numbers cant be abused. The system will then take that cpr-number, authorized by nem-id, and count it as a personal signature for the chosen party.
  + Signatures are binding can't be withdrawn  
      
    To bind signatures so they ant be withdrawn, we could count the signatures remotely from the party so ensure that no changes or tempting with signatures can occur.
* Exercise 1
  + The first thing we will do is to look at Cesarean rotation. I am not sure it was ever used, but it is good for making text which can't be read by the named eye, but is easy to crack.  
      
    Cesar ROT code is an old method potentially used by cesar where you rotate the alphabet by a number of letters, 3 in this case so that A becomes D ovs…
  + In the sample code, find out how many letters to rotate the hidden message. Notice, the rotator can rotate both forwards and backwards (though that is not really necessary - why?)  
      
    By running the Week-8 example code we can find that the number of rotations is 9:  
    ROT encoded: "Tvjri kyflxyk Tcvfgrkir yru r dfjk tyridzex efjv";  
    9 alphabet rotations: “Cesar thought Cleopatra had a most charming nose”  
      
    The rotation back and forwards and backwards is not real necessary since we can just repeat around the alphabetical characters.
* Exercise 2
  + This exercise uses an "incredible dumb and stupid" password scheme, as the actual input is a number between 1000 and 9999. You must find out which one. Your attack form here is brute force (trying them all).  
      
    Breaking the cifer: "NmxhSedpfrayg4OsgKNWSjVp68E0xa76H5bOa+LEgt3fvVWPM/QHX48ySecVpyEPO/xVRaa2URbzEglWugmPpji8Q6ClwoMYHmX6qtimZ7I=". If you run a for loop like: “for (int i = 1000; i <= 9999; i++)” and run all the key combinations through, you can brute force your way to several string combinations. Finding the right one can be done by filtering/matching for alphabetic and numerical redable charactors in an if statement inside the loop, that prints out the decrypted message: “Mobile was Internet 2.0. It changed everything. Crypto is Internet 3.0.” and the key combination “passwordabcd7345”.
* Exercise 3
  + This exercise is about writing a small toy editor which allow you to store small texts in encrypted format. The save function is there, but you have to write the load.  
      
    To load funktion is already there but need a decrypter to read the encoded text. This encoded text i mainly made up of two components, a 16-byte array and AES implementation for cipher initialization containing a key, the 16-byte array and the massage. These are both encoded in base64 and then concatenated to a string separated by “:”.  
      
    Every time something is saved to the .txt file, a secureRandom implementation for a new Initialization vector is generated. This means that in order to decrypt the data we need to load, we first need to split the “:” separated base64 encryption up and decode them. Out of this we will get 16-byte array we used when encrypting the data, as well as the decrypted combination of the key, 16-byte array and the massage from the cypher initialization. Here we can then initialize a new cipher and combined with the secret key that the user used as super secret (The above button/hash) decrypt the complete data from our ciphertext into plaintext.