Week 2

Andre Rodrigues - up201505639
Pedro Antunes - up201507254

Consolidation questions

1 - What are the practical limitations of the One-Time-Pad?

O One-Time Pad faz XOR de cada bit do plaintext com um bit da chave. A chave não pode ser reutilizada porque:

P1 XOR K = C1

P2 XOR K = C2

C1 XOR C2 = P1 XOR K P2 XOR K = P1 XOR P2

E basta saber P1 ou P2 para podermos descobrir todos os ciphertexts encriptados com esta chave, sem termos de descobrir a chave.

As limitações práticas do One-Time Pad são:

- A troca da chave. A troca da chave tem de ser feita num ambiente totalmente seguro e de forma presencial pois a chave é a única coisa que garante segurança à cifra.

- A chave tem de ter um tamanho muito grande pois cada letra da chave (ou cada bit) corresponde a uma letra da mensagem (ou a um bit da mensagem), ou seja, a chave tem de ter um tamanho tão grande quanto a mensagem o que a torna muito difícil de memorizar e guardar e outro ponto relacionado é o facto de ser preciso uma nova chave cada vez que ciframos daí usualmente o tamanho da chave ter de ser maior do que o tamanho da mensagem para que permita cifrar várias mensagens.

2 - Why is deterministic encryption insecure for most applications?

Cifra determinística é o oposto de cifra probabilística e diz que um algoritmo criptográfico produz sempre o mesmo ciphertext para um determinado plaintext e chave, independente do número de execuções do algoritmo.

O problema deste tipo de cifra é que um atacante pode reconhecer os plaintexts correspondes aos ciphertexts sem saber a chave. O atacante consegue fazer isto pois o algoritmo é público e sabendo qual é o algoritmo pode submeter plaintexts e obter os ciphertexts respetivos até encontrar um ciphertext compatível com o que o atacante procura.

3 - Why is the following encryption scheme IND-CPA secure?

•Let E be a block cipher, say AES, andKa random key
•To encrypt M sample R from the set of blocks
•Return C= ($M \oplus E(K, R)$, R)

Este esquema de encriptação é IND-CPA seguro porque E(K, R) retorna bits aleatórios, que depois iremos usar para fazer XOR com o plaintext e assim obter uma string de bits que não conseguimos distinguir de uma string de bits aleatória.

Como o objetivo de segurança do “IND” é que o ciphertext não pode ser distinguido de uma bit string aleatória, ou seja, o ciphertext tem de parecer aleatória então conseguimos isso como descrito no passo anterior.

Já o modelo de atacante “CPA” diz que o atacante pode pode executar queries de cifra para plaintexts a sua escolha e consegue observar o ciphertext resultante.

O modelo descrito também respeita esta propriedade porque introduz um bit aleatoria R a cada iteração do algoritmo (por isso que passamos R para que o ciphertext possa ser decifrado), fazendo com que o atacante não consiga saber a que plaintext corresponde aquele ciphertext pois o mesmo plaintext terá vários ciphertexts.

4 - What advantages/disadvantages of public-key encryption over symmetric encryption?

Vantagens:

- A chave não precisa de ser partilhada previamente, para poder estabelecer uma comunicação, qualquer um pode enviar informação usando a chave publica do destinatário. E para além disso se estivermos a falar de um contexto de redes então a comunicação usada para estabelecer a chave é inseguro logo a chave pode estar comprometida o que leva a ter de haver uma troca de chaves pessoalmente.

- A chave não é conhecida por ambas as partes, ou seja, aumenta a segurança da informação pois caso a chave seja comprometida só um lado da comunicação é que fica exposto.

- Consegue fornecer assinaturas digitais para confirmar a autenticidade do destinatário da mensagem. As assinaturas digitais também permitem o não repúdio pois cada mensagem estará ligada a um emissor caindo sobre ele a sua responsabilidade. E outra vantagem das assinaturas digitais é permitir saber se a mensagem foi alterada no percurso.

Desvantagem:

- Ambos os processos de cifrar e decifrar são mais lentos comparado com as cifras simétricas.

- Não consegue usar a password usada na autenticação para provar a identidade do recipiente, ou seja, o emissor não tem como saber se aquela chave pública é mesmo do destinatário pretendido.

Guided practical assignment

1 - Use the Sage web interface to interpret the following script

EncObject = ATTACKATDAWN

Ciphertext = DWWDFNDWGDZQ (shift 3)

Plaintext = ATTACKATDAWN

2 - Sage scripts can also be run directly in the console by creating a Python file with the following contents

$ sage -python caesar.py

3 - We can use the python cryptography library (`pyca/cryptography`) to compute one block with AES.

$ python one_block_AES.py

$ openssl enc -aes-128-ecb -nopad -d -K c1c14b577637a9f3b39460ed506ba0dd -in ciphertext.bin

attack at dawn!!

-aes-128-ecb -> diz que o ciphertext foi cifrado com o algoritmo AES com o modo ECB e com nível de segurança de 128 bits.
-nopad -> diz que o ciphertext não possui padding
-d -> significa que pretendemos decifrar
-K -> a chave privada usada para cifrar
-in -> ciphertext

4 - We can use openSSL in the console to invert the block.

```
openssl enc -aes-128-ecb -nopad -d -K <key_in_hex> -in ciphertext.bin
```

Can you check if inversion was correct? Explain the options used in this command.

Programming assignment

O IND-CPA significa que fornecemos dois plaintexts e recebemos um ciphertext de um dos plaintexts e temos de adivinhar a qual plaintext corresponde o ciphertext.

Neste caso, o servidor que é onde está implementado o IND-CPA reutiliza a chave, ou seja, envia-mos para o servidor o plaintext m0 no qual ele nos devolve o ciphertext c0 e fazemos o mesmo para outro plaintext c1, obtendo m1. Depois no desafio iremos usar as duas mensagens usadas anteriormente, m0 e m1, onde nos será dado o c1 ou c0 obtido anteriormente, isto porque a chave é a mesma pois caso não fosse os resultados de encriptar o m0 ou m1 duas vezes dariam cifras diferentes. Ora agora basta comparar a cifra que o servidor nos deu com as cifras obtidas no primeiro passo para sabermos qual o plaintext correspondente e fazer uma hipótese que estará correta em todas as iterações.

Mostrando o que já sabíamos que para quebrar o one-time-pad é necessário reutilizar a chave.