Criptologia
C-Academy

Aula Prática 1

Sumário
Introdução ao tema da criptografia de chave simétrica através do desenvolvimento de tarefas dedicadas ao manuseamento das cifras clássicas mais conhecidas. Discussão de diversos conceitos e termos do jargão da criptografia.

Pré-requisitos:
Algumas das tarefas propostas a seguir requerem o uso de *software* para efetuar cálculos e o acesso a um sistema com compilador de programas escritos em linguagem de programação C. Sugere-se, assim, o uso de uma distribuição comum de Linux, onde todas estas condições estarão provavelmente preenchidas.

1 A Cifra de César

A cifra original de César (chamada assim por ter sido usada pelo imperador romano Júlio César) usava uma translação fixa de 3 letras para a direita do alfabeto. Contudo, de uma forma geral, se considerarmos que as mensagens a cifrar são todas constituídas pelas letras do alfabeto com 26 letras e a cada uma atribuirmos um valor inteiro de 0 a 25, i.e.,
`A B C D E F G H I J K L M N O P Q R S T U V W X Y Z`
`0 1 2 3 4 5 6 7 8 ...`,
então a cifra de cada letra da mensagem é dada por $E(k, M_i) = M_i + k \bmod 26$, enquanto que a decifra é definida por $D(k, M_i) = M_i - k \bmod 26$, em que $k = 1, 2, ..., 26$ e M_i representa a letra i da mensagem $M \in \mathcal{M}$.

A cifra original de César era, portanto, dada por $E(M_i) = M_i + 3 \bmod 26$ e a decifra por $E(M_i) = M_i - 3 \bmod 26$ (i.e., $k = 23$).

Tarefa 1 ***Task 1***

Cifre a palavra `OLA` usando a cifra original de César.
Q1.: Qual o resultado? _____

Tarefa 2 ***Task 2***

Decifre agora o criptograma seguinte, sabendo que a chave utilizada foi 10:
Criptograma: `LOXPSMKOYWKSYB`

Texto limpo: _____

Tarefa 3 ***Task 3***

Decifre o criptograma seguinte, mas desta feita apenas sabendo que as três letras mais comuns na Língua Portuguesa são o `A`, o `E` e o `O`. **Nota:** por comodidade, deixaram-se os espaços e as pontuações na frase,

cifrando-se apenas as letras do alfabeto indicado antes.

Criptograma:
`J HVM NVGBVYJ, LPVIOJ YJ OZP NVG`
`NVJ GVBMDHVN YZ KJMOPBVG!`

Q2.: Para o alfabeto especificado em cima, quantas chaves diferentes se podem definir?
☐ 1. ☐ 5. ☐ 10. ☐ 25. ☐ 26. ☐ 32.

Q3.: Em média e mesmo que não soubesse nada acerca das frequências relativas das letras do alfabeto do texto limpo, de quantas tentativas precisava para encontrar o texto limpo original?
☐ -1. ☐ π. ☐ 5. ☐ 12,5. ☐ 13 ☐ 25. ☐ 1500.

Na verdade, o que fez na tarefa anterior foi atacar a cifra de acordo com um modelo de ataque. **Q4.: Ainda que não seja especialista na área, faça um esforço para tentar identificar o modelo de ataque que utilizou:**
☐ *Ciphertext-only attack (COA)*
☐ *Known-plaintext attack (KPA)*
☐ *Chosen-plaintext attack (CPA)*
☐ *Adaptive chosen-plaintext attack (CPA2)*
☐ *Chosen-ciphertext attack (CCA)*
☐ *Adaptive chosen-ciphertext attack (CCA2)*
☐ *Side-channel attack*.

2 A Cifra de Vigenère

A cifra de Vigenère, assim designada também devido ao seu criador, é um pouco mais segura que a cifra de César. Enquanto que na cifra de César, a chave de cifra é apenas um número que denota a deslocação, na cifra de Vigenère, a chave de cifra é uma palavra ou uma série de caracteres. Para cifrar uma mensagem, repete-se a chave de cifra tantas vezes quanto necessário para se perfazer o tamanho do texto-limpo, e depois somam-se (módulo 26, neste caso), as letras do texto-limpo com a chave para se obter o criptograma. Por exemplo, se a chave de cifra for `OLA` e o texto-limpo for `ESTAAULAEUMASECA`, o criptograma obtinha-se da seguinte forma:

```
 ESTAAULAEUMASECA
+OLAOLAOLAOLAOLAO
=SDTOLUZLEIXAGPCO
```

Repare que, neste caso e ao contrário do que acontece para a cifra anterior, a mesma letra pode ser cifrada de formas diferentes, se estiver em partes diferentes da mensagem. Por exemplo, a letra `E` é transformada em `S` e em `P` em diferentes partes da mensagem anterior.

Tarefa 4 ***Task 4***

Cifre a mensagem `TIO MANEL TINHA UMA QUINTA` com a chave de cifra `AULA`.

Criptograma: _____

Q5.: Quantas chaves de cifra existem com 4 letras diferentes?
☐ 4 ☐ $25 \times 24 \times 23 \times 22$ ☐ $25!$
☐ $4!$ ☐ $26 \times 25 \times 24 \times 23$ ☐ $26!$

Q6.: Qual, ou quais, as famílias de chaves que transformam a cifra de Vigenère numa cifra de César?
☐ Chaves com letras todas iguais.
☐ Chaves com uma só letra.
☐ Chaves com duas letras apenas.
☐ Chaves com todas as letras diferentes.

Tarefa 5 *Task 5*

Decifre o criptograma seguinte (ou encontre a chave de cifra), sabendo que a cifra utilizada foi a cifra de Vigenère, a primeira palavra é `ISTO` e a chave de cifra tem 3 letras:
Criptograma
`JUWP G IBELM`

Texto-limpo: _____

Q7.: Faça novamente um esforço para tentar identificar o modelo de ataque que utilizou para quebrar a cifra desta vez:
☐ *Ciphertext-only attack (COA)*
☐ *Known-plaintext attack (KPA)*
☐ *Chosen-plaintext attack (CPA)*
☐ *Adaptive chosen-plaintext attack (CPA2)*
☐ *Chosen-ciphertext attack (CCA)*
☐ *Adaptive chosen-ciphertext attack (CCA2)*
☐ *Side-channel attack*.

3 Cifra de Substituição

Na cifra de substituição, a chave de cifra é simplesmente a definição de uma tabela de correspondências de cada letra do alfabeto que se está a utilizar para a respetiva cifra dessa letra. Por exemplo, a chave seguinte determina que todos os `A` do texto-limpo sejam transformados em `S` no criptograma, e todos os `B` em `Q`, etc.:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
S Q T U H J I B Y K A V L C W E Z N R M X G F P D O
```

A operação de decifra consiste em simplesmente olhar para a correspondência no sentido contrário.

Tarefa 6 *Task 6*

Construa um programa em linguagem C para cifrar e decifrar usando a cifra de substituição. Considere começar com o programa incluído a seguir:

```c
#include<stdio.h>

void encrypt(char * in, char * out, char * key, int size){
  int i = 0;
  for(i = 0; i < size; i++){
     int iFound = 0;
     int j = 0;

     while( iFound == 0 )
       if( in[i] == key[j] )
          iFound = 1;
       else
          j++;

     out[i] = key[26+j];
  }
}

int main(){

  char key[2*26] = {
    'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W
        ','X','Y','Z',
    'S','Q','T','U','H','J','I','B','Y','K','A','V','L','C','W','E','Z','N','R','M','X','G','F
        ','P','D','O'
  };

  char plaintext[12]  = "OLACOMOESTA";
  char ciphertext[12] = "XXXXXXXXXXX";
  char plaintext2[12] = "XXXXXXXXXXX";

  printf("%s\n",plaintext);
  encrypt(plaintext,ciphertext, key,11);
  printf("%s\n",ciphertext);
  decrypt(ciphertext,plaintext2, key,11);
  printf("%s\n",plaintext2);
}
```

Q8.: Depois de analisar a sua definição, consegue dizer quantas chaves diferentes suporta a cifra de substituição?

☐ 5 ☐ $5!$ ☐ $25!$ ☐ $26!$ ☐ 2^{80} ☐ $2^{\log_2 5}$

Um computador moderno consegue efetuar cerca de 2^{26} operações compostas num segundo. **Q9.: Acha que esse computador conseguia testar exaustivamente (i.e., por *brute force*) todas as chaves possíveis para a cifra analisada em tempo útil?**

☐ Sim, conseguia mas demorava algumas horas.
☐ Sim, conseguia nas calmas. Curte!
☐ Não, não conseguia.

Q10.: Tendo em conta o que fez e estudou até esta parte do guia, esta cifra parece-lhe segura?

☐ Sim, parece-me ser segura.
☐ Em termos de número de chaves, parece-me ser segura, mas em termos de facilidade de ataque, não.

Q11.: Esta cifra é vulnerável a ataques em que se conhece parte do texto-limpo associado a um criptograma ou em que o texto-limpo associado a um criptograma tem propriedades estatísticas notáveis?

☐ Sim, é vulnerável em ambas as situações.
☐ É vulnerável apenas na primeira situação.
☐ É vulnerável apenas na segunda situação.
☐ Não é vulnerável em nenhum dos casos.

A **Enigma**[1] era uma máquina que implementava uma cifra de substituição polialfabética através do encadeamento de 3 rotores (que podiam ser escolhidos de um conjunto de 5). Na sua forma mais simples (sem o chamado *dashboard*), o número máximo de chaves (combinações) suportadas era de $A_3^5 \times 26^2 \times 26^3 = 712882560$:

- A_3^5 é o número de arranjos possíveis na escolha de 3 em 5 rotores;
- 26^2 é o número de posições possíveis para os saltos entre os rotores (o rotor do meio podia iterar após o primeiro rotor chegar à letra `A`, ou à letra `B`, `C`, etc.);
- 26^3 é o número de posições iniciais dos rotores (cada rotor podia começar numa de 26 letras).

Esta máquina suportou as comunicações alemãs durante bastante tempo, e motivou também imensa investigação na sua criptanálise. Na altura, a máquina constituía um desafio, porque tentar todas as 712882560 combinações manualmente e para cada mensagem era uma tarefa difícil e morosa, para além de sujeita a erros. **Q12.: Quanto tempo demoraria um computador atual a tentar essas combinações?**

_____

_____

Sugestão: experimente fazer um programa que conte até 712882560 e verifique o tempo que demora.

4 One Time Pad

A *one time pad* é conhecida como a cifra simétrica com segurança perfeita, embora tenha outros defeitos.

Tarefa 7 ***Task 7***

Pegue numa moeda e atire-a 16 vezes ao ar (faça isso com o devido cuidado e respeito). Por cada lançamento, aponte um `0` ou um `1` num ficheiro de texto conforme saia cara ou coroa.

Q13.: Quantos `0`s sairam? _____

Q14.: Quantos `1`s sairam? _____

Q15.: Quantas vezes saiu a combinação `00`? _____

Q16.: Quantas vezes saiu a combinação `01`? _____

Q17.: Se lhe dissessem o que saiu das 6 primeiras vezes, conseguia adivinhar o que ía sair na sétima?
☐ Não, não ia.
☐ Com uma probabilidade de 1/6, sim, ía.

Q18.: A sequência que resultou desta experiência vai de encontro ao conceito que tem de aleatoriedade?
☐ Sim, vai.
☐ Não, não vai.
☐ Nunca pensei nisso, mas vai.

[1] É possível ver a enigma a funcionar em http://enigmaco.de/enigma/enigma.html e encontrar bastante informação útil em https://plus.maths.org/content/exploring-enigma. A página https://observablehq.com/@tmcw/enigma-machine tem outra representação interessante, embora simplificada.

Tarefa 8 ***Task 8***

Considere que estava a tentar transmitir uma mensagem em binário a um(a) colega seu(ua). A mensagem era `0000000100000001`. **Q19.: Esta mensagem parece-lhe aleatória ou fácil de prever?**
☐ Aleatória. ☐ Fácil de prever.

Analise a tabela seguinte, que define a operação de $\oplus$ (xor ou ou exclusivo):

$\oplus$	0	1
0	0	1
1	1	0

Use a tabela para calcular o xor da mensagem a transmitir com a sequência que gerou durante a experiência da moeda. Observe as caracteristicas do resultado guiando-se pelas seguintes questões.

Q20.: Quantos `0`s tem o resultado do xor? _____

Q21.: Quantos `1`s tem o resultado do xor? _____

Q22.: Quantas vezes tem a combinação `00`? _____

Q23.: Quantas vezes tem a combinação `11`? _____

Q24.: A sequência resultante parece-lhe ser aleatória?
☐ Sim, de facto parece. ☐ Não, não parece.

Q25.: Se enviar a sequência resultante do xor ao(à) seu(ua) colega, este(a) consegue recuperar o texto-limpo da mensagem? De que forma?
- ☐ Atirando também a moeda ao ar, registando os resultados e fazendo o xor ao contrário.
- ☐ Atirando também a moeda ao ar, registando os resultados e fazendo o xor da mesma forma.
- ☐ É impossível obter o texto-limpo de volta, a não ser que também lhe envie a sequência resultante da minha experiência.
- ☐ É impossível obter o texto-limpo de volta, independentemente das condições.

5 Exploração do OpenSSL

Nesta e nas próximas aulas, o OpenSSL revelar-se-á um recurso extremamente útil no contexto da utilização e estudo de ferramentas criptográficas, entre outras. As próximas tarefas estão desenhadas de forma a explorar, ainda que de forma superficial, este recurso. Este conhecimento será aprofundado ao longo de vários guias.

Tarefa 9 ***Task 9***

Inicie o seu computador no sistema operativo Fedora e abra um *browser*. Use a Internet para responder à seguinte questão. **Q26.: O que é o OpenSSL?**

_____

_____

Q27.: Procure saber se o OpenSSL é importante nos dias de hoje e se, apesar de ser um recurso que tem a ver com a segunça da informação, foi a base de alguma vulnerabilidade crítica nos últimos tempos.

☐ O OpenSSL? E isso existe?

☐ O OpenSSL é importante mas não há nada a reportar acerca de bugs severos, em termos de segurança, na sua implementação.

☐ O OpenSSL é muito importante e (eishh!) continha um mega *bug* que ía acabando com a Internet.

Procure continuar a responder acertadamente recorrendo agora ao manual do OpenSSL.

Q28.: Como é que normalmente se pode aceder ao manual de um comando Linux ou Unix *like*?

☐ Não faço a mínima ideia.

☐ Procurando o manual do sistema operativo na gaveta, e abrindo-o na página relativa ao comando.

☐ Escrevendo `> __________` no terminal...

Q29.: Há alguma diferença entre OpenSSL (devidamente capitalizado) e `openssl` (em `monospace`)?

☐ Sim, há... um é um _____ e o outro é um _____.

☐ *Nope*... as duas designações referem-se exatamente à mesma *toolkit*.

Q30.: Pode usar o OpenSSL para gerar chaves assimétricas?

☐ Nunca experimentei, mas penso que sim.
☐ Nunca experimentei, mas penso que não.

Q31.: Pode usar esta ferramenta para lidar com e-mail cifrado?

☐ Claro. ☐ Não.

Q32.: E para gerar *timestamps*?

☐ Também dá. ☐ Não.

Q33.: E para verificar o MD5 de determinado ficheiro?

☐ FAZ TUDO! ☐ Não, não dá...

Q34.: Pode usar o OpenSSL para fazer o pequeno almoço?

☐ Dá, e pergunta como queremos os ovos.
☐ Não, mas de resto faz tudo...

Tarefa 10 ***Task 10***

Construa o comando OpenSSL que lhe permite gerar **10 bytes** aleatórios de qualidade em hexadecimal.

_____

Tarefa 11 ***Task 11***

Na linha de comandos (terminal), escreva `openssl` e prima `enter`. **Q35.: Acha que ainda está na linha de comandos?**

☐ Sim. ☐ Não.

Justifique. _____

Escreva `help` na *shell* que deve ter disponível depois do passo anterior. **Q36.: Acha que `help` é um comando/opção do OpenSSL?**

☐ Sim. ☐ Não.

Q37.: Quantos são os comandos principais (*standard*) que tem à disposição?

☐ 1	☐ 45	☐ 46	☐ 47
☐ πr^2	☐ 50	☐ 101110_2	☐ $2E_{16}$
☐ 101111_2	☐ 30_{16}	☐ 1201_3	☐ 48
☐ 110000_2	☐ $2F_{16}$	☐ 1210_3	☐ 49

Q38.: Das seguintes, quais correspondem a opções existentes para o comando `openssl enc`?

☐ -in <file>	☐ -out <file>	☐ -pass <arg>
☐ -e	☐ -d	☐ -a/-base64
☐ -α	☐ -r	☐ -45 <file>
☐ -k	☐ -kfile	☐ -md
☐ -S	☐ -K/-iv	☐ -[pP]
☐ -bufsize <n>	☐ -nopad	☐ -breakfast

Nota: para responder a esta questão, experimentou escrever `enc -help`?

Q39.: Será que o OpenSSL também consegue comprimir e descomprimir ficheiros?

☐ Só não fala como as pessoas!

☐ Não, visto que mesmo o encadeamento de comandos `> man enc | grep compress` não devolve qualquer resultado...