Instituto Superior Técnico, Tagus Park

**Disciplina de Aplicações e Implementações de Algoritmos Criptográficos**

MERC: 1º ano, 2º semestre, 2012/13

65963 – David Dias Data: 22/03/2013, Sexta-feira

63520 – Fábio Domingos

AES – Implementação em Java

# **Abstract:** Implementação em Java do algoritmo de cifra AES e analise da sua performance face aos vários modos de cifra (ECB e CBC)

# Introdução

O AES(1998) é uma algoritmo de cifra simétrica baseado na cifra Rijandael, que vem para substituir o DES(1977). Os 4 passos fundamentais do AES são a Substituição de Bytes (SubBytes), troca de linhas (ShiftRows), mistura de colunas(MixColumns) e o XOR com a chave(AddRoundKey), especifica por cada Ronda.

Este documento apresenta a implementação desenvolvida em Java para chaves com 128 bits e para os modos de cifra de blocos ECB e CBC.

# Funcionamento

A implementação desenvolvida apresenta uma interface:

* **void init( int Cipher. int CMode, byte[] key);**
* **byte[] update(byte[] plaintext);** e **byte[] update(byte[] plaintext, int inputlen);**
* **byte[] doFinal(byte[] plaintext);** e **byte[] doFinal(byte[] plaintext, int inputlen);**

A invocação deste metodos pode ser feita da seguinte forma:

AES [–e/d] [–ecb/cbc/ctr] [–f Keyfile] [–k Key] [–i inputfile] [–o outputfile]

* **-e/d:** Define o modo de cifra que queremos utilizar. e – cifra, d - decifra
* **–ecb/cbc/ctr:** Define o tipode de cifra de bloco a utilizar
* **–f Keyfile:** Especifica o local onde se encontra a chave que será usada na cifra/decifra.
* **–k Key:** Especifica uma chave para a cifra/decifra através da linha de comandos
* **–i inputfile:** Indica o local onde se encontra o ficheiro que queremos cifrar/decifrar
* **–o outputfile:** Indica o local onde queremos guardar o ficheiro cifrado/decifrado

Para o correcto funcionamento desta interface deverá ser escolhida apenas uma das opções –f ou –k.

Ao ser iniciada a interface, a função **init** gera de imediato todas as chaves de ronda que iram ser utilizadas para cifrar/decifrar os diferentes blocos ao longo do processo.

A função **update** terá como função principal partir o ficheiro a cifrar/decifrar em varios blocos, cifrando/decifrando cada um deles sequencialmente.

A função **dofinal** irá agrupar todos os blocos cifrados/decifrados pela função update, fazendo o padding do ultimo bloco no final de todo o processo.

Gerar as chaves para todas as rondas à priori e esta ficam guardadas até a cifra terminar

10 cycles of repetition for 128-bit keys

melhor velocidade a partir indexação dos padrões no ECB

# Analise de performance

Analise ao ECB

Analise ao CBC

Fazer gráficos no Excell com os tempos do crypto do API do Java e o nosso para ambos os casos

# Propostas de aumento de performance

# Conclusões