TÉCNICO LISBOA

Fundamentos da Programação

Primeiro Projecto

30 de Setembro de 2013

Encriptação de mensagens

Pretende-se com este trabalho a realização em Python de uma versão simplificada de um algoritmo de encriptação.

A criptografia (do grego "kryptos", oculto, e "graphein", escrever) estuda formas de ocultar informação. Julga-se que já os Gregos Antigos e os Espartanos usavam formas de encriptação, para enviar mensagens durante as campanhas militares. Nos dias de hoje, a criptografia é essencial em aplicações que utilizam os cartões Multibanco ou de crédito e o comércio electrónico.

Actualmente existem vários algoritmos de encriptação, continuando matemáticos e informáticos a trabalhar no desenvolvimento de algoritmos que se pretendem cada vez mais seguros.

Um algoritmo de encriptação transforma um texto normal num texto encriptado. Naturalmente, para recuperar o texto original será necessário aplicar-lhe um outro algoritmo que faça a operação inversa. Assim, estes algoritmos existem sempre aos pares, constituindo aquilo a que se chama uma cifra. Para decifrar um texto encriptado será necessário não só o algoritmo correcto, como também a *chave* usada na encriptação. Por exemplo, uma forma muito simples de encriptar um texto é substituir cada letra pela letra que está i posições mais adiante no alfabeto. Será fácil escrever algoritmos que encriptem um texto, e que decifrem um texto encriptado, mas só será possível decifrar um texto encriptado se for conhecida a chave usada na encriptação, neste caso o valor de i.

No exemplo dado não é difícil, por meio de tentativas, descobrir qual a chave usada. Assim, a cifra apresentada não poderia ser usada numa situação do mundo real, pois seria muito fácil de descobrir. Com a existência de computadores cada vez mais rápidos, cada vez é maior a necessidade de desenvolver cifras sofisticadas, que não sejam fáceis de descobrir.

1 O algoritmo RSA

O algoritmo RSA, de Rivest, Shamir e Adleman, os seus criadores em 1977, permite encriptar textos com uma chave pública. Isto quer dizer que tanto o algoritmo de encriptação como a chave usada podem ser conhecidos por todos. Evidentemente, existe outra chave, a chave privada, sem a qual não é possível decifrar textos encriptados usando a chave pública. Descreve-se em seguida o algoritmo RSA de uma forma simplificada, começando por descrever a forma de gerar as chaves pública e privada.

1.1 Geração das chaves

Tanto a chave pública como a chave privada correspondem a pares de inteiros; representamos a primeira por (m, e) e a segunda por (m, d). Os passos a seguir para a geração destas chaves são:

1. Escolhem-se de modo arbitrário dois números primos, p e q. Estes números têm que ser mantidos secretos e, quanto maiores forem, mais seguro é o algoritmo.
2. O primeiro elemento de ambas as chaves é dado por
$$m = p \times q$$
3. Seja $n = (p-1) \times (q-1)$.
   (a) Escolhe-se o segundo elemento da chave pública, e, de forma a que as seguintes condições sejam verificadas[1]:
   - $1 < e < n$
   - $mdc(e, n) = 1$

   (b) Determina-se o segundo elemento da chave privada, d, de forma a que, para algum inteiro k, se tenha
$$d \times e = 1 + k \times n$$

1.2 Encriptação

Dada uma mensagem, esta é primeiro transformada num inteiro, N. A mensagem encriptada é dada por[2]:

$$C = N^e \, mod \, m$$

Dada uma mensagem encriptada C, a mensagem original é obtida da seguinte forma:

$$N = C^d \, mod \, m$$

Uma vez que o resto da divisão inteira de qualquer inteiro por m é inferior a m, tanto a mensagem original como a mensagem encriptada terão de ser inferiores a m.

2 Exemplo

Comecemos por determinar uma chave pública e a correspondente chave privada. O primeiro passo corresponde a escolher dois números primos. Escolhemos os números primos $p = 61$ e $q = 127$. Temos:

$m = 7747$

[1] Em que $mdc(e, n)$ representa o máximo divisor comum entre e e n, significando que e e n são co-primos.
[2] Em que $a \, mod \, b$ representa o resto da divisão inteira de a por b.

$n = 7560$

Suponhamos que era escolhido $e = 11$. Trata-se de uma escolha válida, pois $1 < 11 < 7560$ e $mdc(11, 7560) = 1$. Como $4811 \times 11 = 1 + 7 \times 7560$, podemos escolher $d = 4811$.

Estão assim determinadas a chave pública $(7747, 11)$ e a chave privada $(7747, 4811)$.

Suponhamos agora que pretendemos encriptar a mensagem $N = 100$. A mensagem encriptada é $C = 100^{11}\ mod\ 7747 = 2276$.

A chave privada permite, a partir da mensagem encriptada $C = 2276$ obter a mensagem original: $N = 2276^{4811}\ mod\ 7747 = 100$.

3 Trabalho a desenvolver

Deverá escrever um programa que permita a encriptação e subsequente decifração de mensagens usando o algoritmo RSA. Para tal siga, os seguintes passos:

1. **(3 val.)** Escreva uma função `calcula_e`, de um argumento (n) correspondente a um inteiro positivo, que devolve *o menor* inteiro e, tal que:
   (a) $1 < e < n$;
   (b) e é co-primo com n.
2. **(3 val.)** Escreva uma função `calcula_d`, de dois argumentos (e, n) correspondentes a inteiros positivos, que devolve *o menor* inteiro, d, tal que existe um inteiro, k, que verifica a igualdade $d \times e = 1 + k \times n$. O modo de calcular k fica ao seu critério.
3. **(4 val.)** Escreva uma função `encripta`, que recebe três argumentos (N, i, j) correspondentes a inteiros positivos, e
   (a) gera uma chave pública, utilizando como primos iniciais o i-ésimo e o j-ésimo primos.
   (b) encripta a mensagem correspondente ao inteiro N, usando a chave pública gerada no passo anterior.

   A função `encripta` deve devolver um inteiro, correspondente à mensagem encriptada. Se o valor de N for igual ou superior ao valor de m (este valor é o produto do i-ésimo primo pelo j-ésimo primo), a sua função deve gerar um erro de valor (`ValueError`) com a seguinte mensagem: "`encripta: a mensagem tem de ser inferior a` $\langle m \rangle$", onde $\langle m \rangle$ deverá ser substituído pelo valor de m.
4. **(4 val.)** Escreva uma função `decifra`, que recebe três argumentos (C, i, j) correspondentes a inteiros positivos, e
   (a) gera uma chave privada, utilizando como primos iniciais o i-ésimo e o j-ésimo primos.
   (b) decifra a mensagem usando a chave privada gerada no passo anterior.

A função `decifra` deve devolver um inteiro, correspondente à mensagem decifrada. Se o valor de C for igual ou superior ao valor de m (este valor é o produto do i-ésimo primo pelo j-ésimo primo), a sua função deve gerar um erro de valor (`ValueError`) com a seguinte mensagem: "`decifra: a mensagem tem de ser inferior a` $\langle m \rangle$", onde $\langle m \rangle$ deverá ser substituído pelo valor de m.

Apresenta-se de seguida um exemplo de interacção com o seu programa, onde são usados os mesmos valores que os usados no exemplo da Secção 2:

```
>>> calcula_e(7560)
11
>>> calcula_d(11, 7560)
4811
>>> encripta(100, 18, 31)
2276
>>> decifra(2276, 18, 31)
100
>>> encripta(7747, 18, 31)
encripta: a mensagem tem de ser inferior a 7747
```

4 Classificação

A avaliação da execução será feita através de um sistema automático para entrega de projectos designado Mooshak. Existem vários testes configurados no sistema. O tempo de execução de cada teste está limitado, bem como a memória utilizada. Só poderá efectuar uma nova submissão pelo menos 15 minutos depois da submissão anterior. Só são permitidas 10 submissões em simultâneo no sistema, pelo que uma submissão poderá ser recusada se este limite for excedido. Nesse caso tente mais tarde.

Os testes considerados para efeitos de avaliação podem incluir ou não os exemplos disponibilizados no enunciado, além de um conjunto de testes adicionais. O facto de um projecto completar com sucesso os exemplos fornecidos no enunciado não implica, pois, que esse projecto esteja totalmente correcto, pois o conjunto de exemplos fornecido não é exaustivo. É da responsabilidade de cada grupo garantir que o código produzido está correcto.

Não será disponibilizado qualquer tipo de informação sobre os casos de teste utilizados pelo sistema de avaliação automática. Os ficheiros de teste usados na avaliação do projecto serão disponibilizados na página da disciplina após a data de entrega.

A nota do projecto será baseada nos seguintes aspectos:

1. Execução correcta (14 valores, distribuídos conforme indicado).

   Esta parte da avaliação é feita recorrendo a um programa de avaliação automática que sugere uma nota face aos vários aspectos considerados.

2. Facilidade de leitura, nomeadamente abstracção procedimental, nomes bem escolhidos, qualidade (e não quantidade) dos comentários e tamanho das funções (5 valores).

3. Estilo de programação (1 valor).

5 Condições de realização e prazos

A entrega do 1º projecto será efectuada exclusivamente por via electrónica. Deverá submeter o seu projecto através do sistema Mooshak, até às **23:59** do dia **22 de Outubro de 2013**. Projectos em atraso não serão aceites seja qual for o pretexto.

Deverá submeter um único ficheiro com extensão .py contendo todo o código do seu projecto. O ficheiro de código deve conter em comentário, na primeira linha, os números e os nomes dos alunos do grupo, bem como o número do grupo.

No seu ficheiro de código não devem ser utilizados caracteres acentuados ou qualquer carácter que não pertença à tabela ASCII. Isto inclui comentários e cadeias de caracteres. Programas que não cumpram este requisito serão penalizados em três valores.

Duas semanas antes do prazo de entrega, serão publicadas na página da cadeira as instruções necessárias para a submissão do código no Mooshak. Apenas a partir dessa altura será possível a submissão por via electrónica. Nessa altura serão também fornecidas a cada um as necessárias credenciais de acesso. Até ao prazo de entrega poderá efectuar o número de entregas que desejar, sendo utilizada para efeitos de avaliação a última entrega efectuada. Deverá portanto verificar cuidadosamente que a última entrega realizada corresponde à versão do projecto que pretende que seja avaliada. Não serão abertas excepções.

Pode ou não haver uma discussão oral do trabalho e/ou uma demonstração do funcionamento do programa (será decidido caso a caso).

Projectos iguais, ou muito semelhantes, serão penalizados com a reprovação na disciplina. O corpo docente da cadeira será o único juiz do que se considera ou não copiar num projecto.