Segurança de Sistemas Computacionais
2016/2017, 2º Semestre

Lab 6
Programação em Java/JCA – JCE (Java Cryptographic Extension)

Criptografia Assimétrica:

- **Assinaturas digitais com métodos criptográficos assimétricos**
- **Distribuição e estabelecimento de chaves simétricas de sessão usando métodos criptográficos assimétricos**
- **Acordo de estabelecimento de chaves de Diffie-Hellman**

Exemplos / Demonstrações e Exercícios

Verifique o código que lhe é fornecido para a aula prática (lab6).

1. Verifique o código na diretoria DigitalSignatures, nomeadamente o código **PKCS1SignatureExample.java**
   Com base na observação do código e análise experimental da sua modificação, tente realizar as seguintes alíneas e tirar as respetivas conclusões.

   a) O código mostrado demonstra como fazer uma assinatura digital e como fazer para sua verificação (usando neste caso o algoritmo RSA). Verifique qual a função de síntese subjacente à assinatura (lembrando-se da computação de uma assinatura digital quando se usa o método RSA). Ensaie outras configurações de assinaturas, de acordo com o suporte que tem nos provedores JAVA instalados.

   b) Verifique o tamanho das assinatura geradas. O tamanho é o esperado ? O que determina esse tamanho ? Porquê ?

   c) Em que condições é que o valor da assinatura é sempre diferente em diferentes corridas do programa ? Verifique experimentalmente o seu racional.

   d) Com base no código mostrado, altere-o de forma a produzir e verificar uma assinatura digital usando o algoritmo DSA, gerando para este caso as respetivas chaves em várias dimensões.

2. Verifique agora os restantes materiais que ilustram a utilização de criptografia assimétrica tendo em vista duas técnicas que podem ser interessantes

   2.1) **AESWrapRSAExample.java**
   Neste exemplo a ideia é proteger chaves privadas (neste caso RSA) de forma a que estas sejam processadas estando cifradas com um algoritmo simétrico (neste caso AES). Verifique que este tipo de

construção pode ser facilmente portada para proteger chaves privadas DSA. Tente compreender as vantagens de utilizar na prática este tio de construções.

2.1) Inspirando-se no exemplo anterior como criar envelopes de chave pública para distribuição e estabelecimento de chaves de sessão (para utilização posterior de um algoritmo simétrico) para confidencialidade num canal de comunicação ? Para o efeito veja como base de partida o exemplo em **KeyExchangeRef, RSAKeyExchange.java**

3. Verifique agora como pode programar acordos de estabelecimento de chaves com o método **Diffie-Hellman** (ver diretoria **Diffie-Hellman**)

3.1) Verifique o exemplo **TwoWayDHExample.java** na diretpria Diffie-Hellman. De acordo com o seu conhecimento teórico do método de Diffie-Hellman, tente compreender o código fornecido de modo a comprovar experimentalmente o modelo do acordo de Diffie-Hellman entre duas entidades.

3.2) A partir do anterior e verificando agora o exemplo **ThreeWayDHExample.java** veja como é possível estender o acordo a 3 principais, verificando a partir daí como poderia estender a N principais envolvidos num estabelecimento de uma chave de sessão comum. De acordo com o seu conhecimento teórico, verifique que o resultado experimental obtido é expectável de acordo com as propriedades matemáticas subjacentes às computações que têm lugar no algoritmo de Diffie-Hellman.

3.3) Tente modificar o exemplo 3.1 de modo a programar um acordo autenticado de Diffie-Hellman, de modo a não ficar vulnerável a ataques do tpo “homem no meio”. Recorde o contexto da aula teórica sobre esse ataque quando o acordo de Diffie-Hellman não é autenticado. Para a realização prática pode usar assinaturas digitais RSA ou DSA.