Seleção PBAD/LabSEC - Etapa final

Gustavo Zambonin
Alexandre Augusto Giron
{gustavo.zambonin,alexandre.giron}@posgrad.ufsc.br

1 Introdução

Esta etapa do processo de seleção consiste em implementar uma aplicação capaz de realizar algumas operações comuns à segurança da computação. O objetivo do desafio não é avaliar o grau de conhecimento do aluno, mas sim, como o aluno se comporta ao se deparar com um desafio.

Aqui serão levados em consideração alguns fatores, tais como, quanto o aluno se dedicou para resolver do desafio, quanto do desafio foi resolvido, qual o comportamento que teve quando surgiu dúvidas, sua organização do código, entre outros fatores.

As operações que a aplicação deverá conter são as seguintes:

- obter o resumo criptográfico de um documento;
- gerar chaves;
- gerar certificados;
- gerar repositório de chaves;
- gerar uma assinatura digital;
- verificar uma assinatura digital.

No capítulo seguinte, serão apresentados alguns conceitos básicos, necessários para o bom entendimento da implementação do desafio.

Desde já, agradecemos o esforço do autor original deste desafio, Lucas Ferraro.

2 Conceitos básicos de criptografia

Segundo Menezes *et al.* [MVvO96], criptografia é o estudo de técnicas matemáticas relacionadas a aspectos da segurança da informação, tais como confidenciabilidade, integridade, autenticação e certificação. De todos os objetivos da segurança da informação citados anteriormente, os quatro acima formam uma base, na qual os demais são derivados deles.

- **Confidencialidade** é um serviço usado para manter o conteúdo da informação para todos que tenham autorização para tal. O segredo é sinônimo de confidencialidade e privacidade. Existem várias abordagens para proporcionar confidencialidade, que vão desde a proteção física até algoritmos matemáticos que tornam os dados ininteligíveis.

- **Integridade** é um serviço que trata da alteração não autorizada de dados. Para assegurar a integridade dos dados, é preciso ter a capacidade de detectar a manipulação dos dados por terceiros não autorizados. Essa manipulação se refere a ações como inserção, exclusão e substituição.
- **Autenticação** é um serviço relacionado à identificação. Essa função se aplica a ambas as entidades e a própria informação. Duas partes que começam uma comunicação devem se identificar uma à outra.
- **Não-repúdio** é um serviço que evita uma entidade de negar autorizações anteriores ou ações. Quando surgir alguma disputa em que uma entidade não admita que autorizou que certa ação fosse tomada, é necessário uma forma de resolver o problema, então uma terceira entidade confiável é convocada para resolver a disputa.

A criptografia é caracterizada por três diferentes dimensões. A primeira é o tipo de operação usado para transformar o texto plano, ou seja, o texto original, em texto cifrado. O segundo é a quantidade de chaves usadas e o terceiro seria pela forma que o texto plano é processado. Para esse desafio, o ponto mais importante é o segundo, pois diferencia a criptografia simétrica da assimétrica, ou de chaves públicas. A seguir será apresentado um breve conceito de criptografia simétrica e uma forma de distribuição de chaves confiável para comunicação segura entre duas entidades distintas. Após, será visto o conceito de criptografia assimétrica e suas aplicações.

2.1 Criptografia simétrica

Segundo Paar *et al.* [PP11], a melhor maneira de definir a criptografia simétrica de forma bem simples é através do exemplo abaixo.

Tem-se dois usuários, Alice e Bob, que querem se comunicar através de um canal inseguro. O termo canal pode soar um pouco abstrato, mas é apenas uma forma geral para dizer que eles querem se comunicar, por exemplo, pela internet. O verdadeiro problema começa quando um terceiro usuário com más intenções aparece na situação, Oscar, que deseja roubar informações da comunicação entre Alice e Bob. É claro que existe diversas maneiras de Alice e Bob se comunicarem sem serem ouvidos, entretanto eles estão em seus escritórios e desejam enviar um ao outro documentos secretos e ninguém, além deles pode saber do que se trata o conteúdo, como na Figura 1.

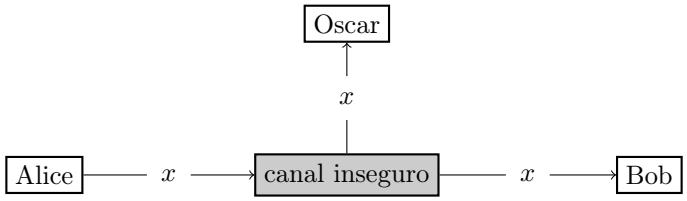


Figura 1: Comunicação insegura.

Uma das soluções para Alice e Bob seria criar uma chave secreta e trocar essa chave por um canal seguro. Assim, usando um algoritmo simétrico de criptografia Alice poderia cifrar os dados originais com essa chave secreta e enviar os dados cifrados para Bob, que conhecendo o algoritmo usado para cifrar poderia usar a mesma chave, já combinada anteriormente, e decifrar o dado para assim ter o texto original de forma segura. Dessa forma Oscar pode até interceptar a

mensagem, porém não poderá decifrá-la, pois não tem conhecimento da chave secreta e não teria como decifrar o texto, como na Figura 2.

Se Alice e Bob não tivessem como combinar uma chave secreta por um meio seguro, essa solução não poderia ser utilizada. Em razão disso, começaram a ser elaborados protocolos para troca de chaves, sendo que um dos primeiros foi o protocolo de Diffie-Hellman, que mostra como usar a criptografia simétrica com duas chaves diferentes. Também a partir desse problema e da ideia de Diffie- Hellman, surgiu a ideia da criptografia assimétrica e da distribuição de chaves por uma entidade confiável de forma hierárquica.

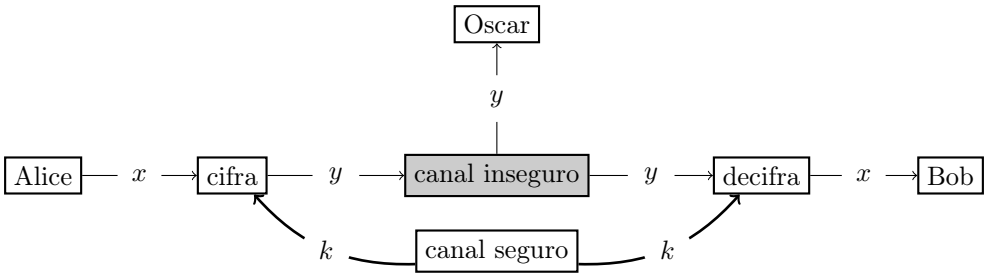


Figura 2: Criptografia simétrica.

2.2 Criptografia assimétrica

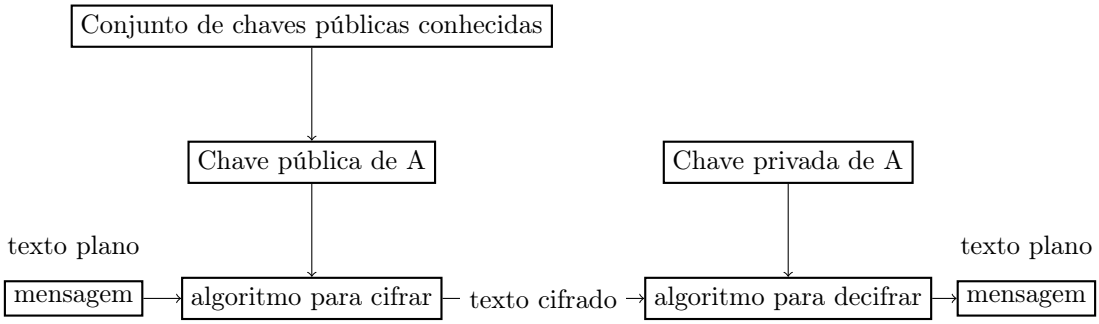
Segundo Stallings [Sta16], o conceito de criptografia assimétrica, também conhecida por criptografia de chave pública, envolve uma tentativa de resolver alguns dos problemas mais difíceis associados à criptografia simétrica. Um deles é a distribuição de chaves, outro seria a falta de proteção na comunicação entre duas entidades.

Na criptografia simétrica uma chave precisa ser estabelecida entre as duas entidades por um canal seguro, como foi demonstrado na Figura 2, na qual apenas com a existência do canal seguro seria possível ocorrer uma comunicação segura. Mesmo assim, se o problema dessa distribuição de chaves fosse resolvido, o número de chaves existentes iria ser imenso, pois imagine que para cada entidade e seus pares comunicantes deveria existir uma chave diferente, o que tornaria totalmente inviável. Ainda existe um outro fator a ser levado em consideração, o não-repúdio.

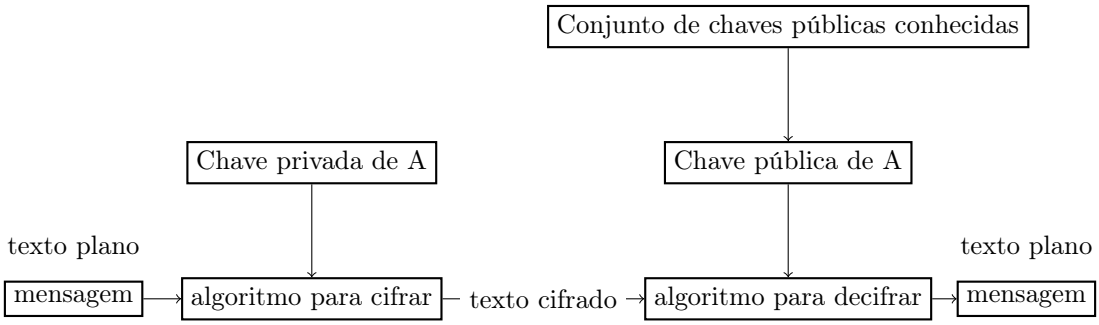
Para resolver esses problemas, Diffie e Hellman [DH76] e Merkle [Mer79] tiveram propostas revolucionárias, baseadas na seguinte ideia: não é necessário que a chave usada para cifrar a mensagem seja secreta. A parte essencial é que o receptor da mensagem pode decifrar apenas com uma chave secreta. Para concretizar isso, o receptor teria de ter publicado uma chave anteriormente para que o emissor pudesse usá-la para cifrar a mensagem. Assim, o receptor teria duas chaves, ou um par de chaves, onde uma é a chave pública e a outra, é a secreta [PP11]. Ainda de acordo com Stallings, o algoritmo de criptografia assimétrica traz uma importante característica, a de ser computacionalmente inviável de determinar a chave privada pela chave pública relacionada.

Stallings define que um esquema de criptografia assimétrica é baseado em seis ingredientes, como visto na Figura 3.

- **texto plano** é a mensagem em sua forma original e legível, também pode ser um dado qualquer. Essa é a entrada para o algoritmo responsável por cifrar.
- **algoritmo para cifrar** faz várias transformações com o texto plano para deixá-lo ilegível.



Sistema de criptografia assimétrica para mensagem secreta



Sistema de criptografia assimétrica para autenticação

Figura 3: Criptografia assimétrica.

- **par de chaves** são dois ingredientes da criptografia assimétrica, a chave pública e a chave privada, sendo que uma é usada para cifrar e a outra para decifrar. O algoritmo para cifrar depende do tipo de chave imposto como entrada.
- **texto cifrado** é a saída do algoritmo usado para cifrar, aqui o texto é totalmente ilegível. Esse texto, ou dado, é totalmente dependente do texto plano e da chave usada para cifrar.
- **algoritmo para decifrar** aceita um texto cifrado e sua chave correspondente para produzir o texto plano.

Existem ainda passos essenciais para esse sistema, que Stallings reporta como segue:

1. Cada usuário gera um par de chaves para ser usado para cifrar e decifrar mensagens;
2. Cada usuário põe uma das chaves (note que pode ser qualquer uma delas, mas apenas uma) em um repositório público ou qualquer local acessível ao público. Essa é chamada de chave pública. A chave restante é a privada, a qual ninguém deve ter acesso, como foi visto na Figura 3. Cada usuário mantém uma coleção de chaves públicas obtida dos outros usuários.
3. Se uma entidade A desejar mandar uma mensagem confidencial para entidade B, então A precisa usar a chave pública de B para cifrar a mensagem.

4. Quando B recebe a mensagem, ele a decifra usando sua chave privada. Nenhum outro usuário pode decifrar a mensagem porque apenas B conhece sua chave privada.

Com esse procedimento, todos os participantes têm acesso a todas as chaves públicas, e as chaves privadas são geradas localmente por cada participante e não podem ser distribuídas. Enquanto a chave privada de um usuário está intacta, comunicações envolvendo esse usuário são seguras. A qualquer momento, algum dos usuários pode trocar seu par de chaves e substituir a chave pública antiga pela nova.

Stallings ainda cita que a criptografia assimétrica pode ser usada em três classes de uso:

- **cifrar e decifrar**: o emissor cifra a mensagem com a chave pública do destinatário;
- **assinatura digital**: o emissor assina a mensagem com sua chave privada, visto com mais detalhes na Subseção 2.4;
- **distribuição de chaves**: existem vários protocolos e serviços de trocas de chaves, mas o que será visto nesse desafio será uma estrutura hierárquica para distribuição e controle das chaves, que será detalhado na Subseção 2.6.

2.3 Resumo criptográfico

Segundo Ferguson *et al.* [FSK11], resumo criptográfico, também conhecido por função de resumo criptográfico, é uma função que recebe uma entrada arbitrária de bits ou bytes, e transforma em uma saída de tamanho fixo. Um uso típico da função de resumo criptográfico é na assinatura digital, onde, ao invés de assinar a mensagem m, que pode ter um tamanho de alguns milhões de bytes, pode-se aplicar primeiro uma função de resumo criptográfico e deixar a mensagem m com um tamanho fixo e muito menor, por exemplo com 1024 bits. Essa aplicação $H(m)$, pode ser chamada de resumo criptográfico de m e oferece um desempenho muito maior ao processo de assinatura digital. Ainda de acordo com Ferguson *et al.*, para esse processo ser seguro, deve ser computacionalmente inviável produzir duas mensagens m_1 e m_2 iguais a partir de entradas diferentes. Isso tornaria a função de resumo criptográfico insegura.

Uma função de resumo criptográfico, ou simplesmente, uma função de hash H precisa ter as seguintes propriedades [Sta16]:

- H tem que ser aplicável a um bloco de dados x de qualquer tamanho;
- H precisa produzir uma saída de tamanho fixo;
- $H(x)$ é relativamente fácil de ser computada para qualquer x, fazendo com que as implementações de hardware e software sejam práticas;
- Para qualquer valor de resumo criptográfico h, é impossível descobrir o bloco de dados x que o gerou. Isso pode ser considerado como uma função de propriedade *one-way*, ou seja, é impossível descobrir o valor que gerou o resultado tomando apenas o resultado como base de busca;
- Para qualquer x, deverá ser computacionalmente inviável encontrar um bloco $y \neq x$ tal que $H(y) = H(x)$.

O resumo criptográfico, por ser uma forma canônica de representar um bloco de dados de tamanho arbitrário, pode ser usado como uma forma de garantir a integridade de uma mensagem, pois é possível enviar a mensagem por completo a um destinatário e junto com ela enviar um resumo criptográfico dessa mensagem. Assim, antes de ler a mensagem recebida, o destinatário,

conhecendo a função de resumo criptográfico usada, poderia aplicar essa função à mensagem e assim conferir se o resumo criptográfico recém gerado é o mesmo que o enviado.

Agora, ele poderia ler a mensagem e ter certeza sobre sua integridade. Essa forma de uso do resumo criptográfico acrescentada à criptografia assimétrica é o que fundamenta a assinatura digital. Dessa forma o emissor usa sua chave privada para cifrar o resumo criptográfico da mensagem a ser enviada e o receptor confere a mensagem enviada usando a chave pública do emissor para recriar o resumo criptográfico, e assim ter a certeza que a mensagem é autêntica e íntegra.

2.4 Assinatura digital

Em situações em que não existe confiança mútua entre o emissor e o receptor, algo além da autenticação é necessário. A forma mais atrativa de solucionar isso é com assinatura digital, que é análoga à assinatura manuscrita [Sta16]. Para tanto, a assinatura digital deve conter as seguintes propriedades:

- deve verificar o autor da assinatura;
- tem que autenticar o conteúdo no momento da assinatura;
- precisa ser verificável por terceiros, para resolver disputas.

Com isso, a assinatura digital inclui a função de autenticação. Stallings diz que com essas propriedades é possível formular os seguintes requisitos para a assinatura digital:

- a assinatura precisa ser um padrão de bits, que depende do conteúdo que está sendo assinado;
- a assinatura precisa usar alguma informação única para o emissor, para prevenir a falsificação do conteúdo e da autoria;
- precisa ser relativamente fácil gerar uma assinatura digital;
- precisa ser relativamente fácil reconhecer e verificar uma assinatura digital;
- precisa ser computacionalmente inviável forjar a assinatura digital, tanto construindo uma nova mensagem para uma assinatura já existente, quanto construindo uma assinatura fraudulenta para uma mensagem qualquer;
- precisa ser prático guardar uma cópia da assinatura digital.

Um método básico para criar uma assinatura digital seria primeiro gerar o resumo criptográfico dos dados a serem assinados e depois o assinante teria de aplicar sua chave privada a esse resumo. Dessa forma seria possível verificar a autenticidade da assinatura e a integridade dos dados assinados. Para tal verificação, basta aplicar a mesma função de resumo criptográfico H aos dados assinados e obter h', que deve ser igual ao resumo criptográfico cifrado pelo assinante.

Para decifrar tal resumo, é necessário conhecer a chave pública do assinante e o algoritmo usado para cifrar. Feito isso, o resultado deve ser o resumo criptográfico h dos dados assinados. Caso $h = h'$ está comprovada a integridade e autenticidade da assinatura digital; caso contrário, essa assinatura foi falsificada e não deve ser confiável. As assinaturas digitais têm muitas aplicações na segurança da informação, incluindo autenticação, integridade de dados e não-repúdio.

Uma das mais importantes aplicações das assinaturas digitais é a certificação de chaves públicas em grandes grupos. Certificação é um meio para uma terceira parte confiável ligar a identidade

de um usuário à sua chave pública. Assim, algum tempo depois, outras entidades podem verificar a autenticidade da chave pública sem a necessidade de uma terceira parte [MVvO96]. Isso será abordado com mais detalhes na Subseção 2.5.

2.5 Certificado digital

Segundo Menezes *et al.*, um certificado digital é uma estrutura de dados que consiste de uma parte de dados e uma parte de assinatura. A parte de dados contém um texto legível, incluindo, no mínimo, a chave pública e um texto identificando a entidade associada a essa chave pública. Nota-se que essa entidade pode ser uma pessoa. A parte da assinatura consiste em uma assinatura digital feita por uma Autoridade Certificadora (AC, explicada na Subseção 2.6.1) sobre a parte de dados da entidade requerente, assim unindo a identidade da entidade com sua respectiva chave pública.

Para gerar um certificado digital, a princípio é necessário seguir os seguintes passos:

1. escolher qual será a AC emissora do certificado;
2. solicitar a essa AC a emissão de um certificado digital, informando os dados do solicitante, tal como seu nome e sobrenome (os dados aqui dependem muito da política usada para a certificação), e esses formarão a parte de dados do certificado digital;
3. em alguns casos, a AC pode solicitar que o usuário solicitante vá até uma autoridade de registro, a qual tem a incumbência de validar os dados informados na solicitação do certificado;
4. caso todos os dados do solicitante estiverem de acordo, a autoridade de registro envia a requisição para a AC, que emite o certificado do solicitante.

A verificação de um certificado digital é feita da mesma maneira que a verificação de uma assinatura digital. Nota-se que a chave pública usada para a verificação será a chave pública da AC emissora do certificado digital. Essa verificação citada pode implicar em um resultado negativo, caso o certificado tenha sido emitido pela AC, mas seja invalidado por alguma razão. Um dessas razões é o certificado ter expirado, ou seja, ter chegado ao fim o tempo previsto de validade. Outra maneira do certificado estar inválido é caso ele tenha sido revogado. Segundo Stallings, existem três possibilidades para a revogação do certificado, que são:

- o usuário avisa que sua chave privada foi comprometida;
- o usuário não é mais certificado pela AC emissora;
- o certificado da AC emissora foi comprometido.

Para manter um registro que algum certificado foi revogado, as ACs guardam essa informação de revogação em uma estrutura chamada de lista de certificados revogados (LCR), que está descrita na Subseção 2.6.2.

2.6 Infraestrutura de chaves públicas

Citando Ferguson *et al.*, uma infraestrutura de chaves públicas (ICP) é uma infraestrutura que permite a um usuário reconhecer a quem uma chave pública pertence.

A estrutura clássica de uma ICP é baseada em hierarquia de autoridades, parecida com uma árvore de estrutura de dados. A ideia aqui é ter um ponto de confiança, que normalmente é uma

autoridade certificadora, que por ser de confiança e ser o nó inicial dessa estrutura, é chamada de AC-Raiz.

Além da AC-Raiz, que é a autoridade máxima dessa estrutura, existem as ACs intermediárias e ACs finais, essas últimas responsáveis pela emissão dos certificados digitais dos usuários finais dessa infraestrutura.

Quando alguma entidade deseja validar um certificado digital de outra entidade, ela precisa conhecer a AC emissora do certificado de tal entidade. Conhecido isso, a entidade pode verificar a assinatura aposta no certificado usando a chave pública da autoridade emissora. Esse mesmo processo tem que ser realizado para a AC emissora e a AC superior a essa emissora, para verificar se a tal AC emissora é válida. Isso se repete até que a AC-Raiz seja alcançada, e quando isso ocorre, a verificação termina, pois a chave pública da AC-Raiz é publicada por outros meio de segurança; por exemplo, a ICP-Brasil é publicada no Diário Oficial da União. Nota-se que esta é uma forma simplificada para esse processo de validação, pois o mesmo envolve outros passos que vão além da verificação das assinaturas nos certificados.

2.6.1 Autoridade certificadora

Segundo Menezes *et al.*, a autoridade certificadora (AC) é uma entidade confiável cuja assinatura sobre o certificado digital comprova que a chave pública pertence à entidade sujeita no certificado.

2.6.2 Lista de certificados revogados

As listas de certificados revogados são estruturas em forma de lista a qual as ACs, de todos os níveis, mantém as informações sobre qual certificado emitido por ela, em algum momento, está revogado. Essas listas são publicadas periodicamente e são assinadas pela própria AC emissora.

De acordo com Stallings, a única forma que as ACs tem para identificar um certificado presente na LCR é o *serial number* do certificado, que é um identificador numérico. Esse é emitido pela AC junto ao certificado de um usuário, e é através desse identificador numérico que são feitas as buscas por certificados revogados dentro da LCR.

3 Etapas do desafio

Dados os conceitos básicos, então, é chegada a hora de implementar. Com o intuito de auxiliar no desenvolvimento das tarefas do desafio foi elaborado um fluxo "aconselhável" da ordem de implementação. Note que não é obrigatório segui-lo, mas pode ser mais fácil compreender o que está sendo feito dessa forma.

Arquivos de interesse:

- `hiring-0.2-src.tar.bz2`, entregue junto a este documento. Contém uma estrutura para auxiliar no desenvolvimento, vista em detalhes na Seção 4;
- `artefatos/textos/textoPlano.txt` é um arquivo de texto que será usado para cifrar, resumir e assinar ao longo das etapas, encontrado dentro do arquivo acima. **Deverá conter seu nome e matrícula da graduação**.

3.1 Primeira etapa — obter o resumo criptográfico de um documento

Basta obter o resumo criptográfico do documento `textoPlano.txt`.

Os pontos a serem verificados para essa etapa ser considerada concluída são os seguintes:

- obter o resumo criptográfico do documento, especificado na descrição dessa etapa, usando o algoritmo de resumo criptográfico conhecido por SHA-256;
- armazenar em disco o arquivo contendo o resultado do resumo criptográfico, em formato hexadecimal.

3.2 Segunda etapa — gerar chaves assimétricas

A partir dessa etapa, tudo que será feito envolve criptografia assimétrica. A tarefa aqui é parecida com a etapa anterior, pois refere-se apenas a criar e armazenar chaves, mas nesse caso será usado apenas um algoritmo de criptografia assimétrica, o ECDSA.

Os pontos a serem verificados para essa etapa ser considerada concluída são os seguintes:

- gerar um par de chaves usando o algoritmo ECDSA com o tamanho de 256 bits;
- gerar outro par de chaves, mas com o tamanho de 512 bits. Note que esse par de chaves será para a AC-Raiz;
- armazenar em disco os pares de chaves em formato PEM.

3.3 Terceira etapa — gerar certificados digitais

Aqui você terá que gerar dois certificados digitais. A identidade ligada a um dos certificados digitais deverá ser a sua. A entidade emissora do seu certificado será a AC-Raiz, cuja chave privada já foi previamente gerada. Também deverá ser feito o certificado digital para a AC-Raiz, que deverá ser autoassinado.

Os pontos a serem verificados para essa etapa ser considerada concluída são os seguintes:

- emitir um certificado digital autoassinado no formato X.509 para a AC-Raiz;
- emitir um certificado digital no formato X.509, assinado pela AC-Raiz. O certificado deve ter as seguintes características:
  - `Subject` deverá ser o seu nome;
  - `SerialNumber` deverá ser o número da sua matrícula;
  - `Issuer` deverá ser a AC-Raiz.
- armazenar em disco os certificados emitidos em formato PEM;
- as chaves utilizadas nessa etapa deverão ser as mesmas já geradas.

3.4 Quarta etapa — gerar repositório de chaves seguro

Essa etapa tem como finalidade gerar um repositório seguro de chaves assimétricas. Esse repositório deverá ser no formato PKCS#12. Note que esse repositório é basicamente um tabela de espalhamento com pequenas mudanças. Por exemplo, sua estrutura seria algo como `<Alias, <Certificado, Chave Privada>>`, onde o *alias* é um nome amigável dado a uma entrada da estrutura, o certificado e chave privada devem ser correspondentes à mesma identidade. O *alias* serve como elemento de busca dessa identidade. O PKCS#12 ainda conta com uma senha, que serve para cifrar a estrutura (isso é feito de modo automático).

Os pontos a serem verificados para essa etapa ser considerada concluída são os seguintes:

- gerar um repositório para o seu certificado/chave privada com senha e *alias* de acordo com as constantes fornecidas;
- gerar um repositório para o certificado/chave privada da AC-Raiz com senha e *alias* de acordo com as constantes fornecidas.

3.5 Quinta etapa — gerar uma assinatura digital

Essa etapa é um pouco mais complexa, pois será necessário que implemente um método para gerar assinaturas digitais. O padrão de assinatura digital adotado será o *Cryptographic Message Syntax* (CMS). Esse padrão usa a linguagem ASN.1, que é uma notação em binário, assim não será possível ler o resultado obtido sem o auxílio de alguma ferramenta. Caso tenha interesse em ver a estrutura da assinatura gerada, então, recomenda-se o uso da ferramenta `dumpasn1`.

Os pontos a serem verificados para essa etapa ser considerada concluída são os seguintes:

- gerar um assinatura digital usando o algoritmo de resumo criptográfico SHA-256 e o algoritmo de criptografia assimétrica ECDSA;
- o assinante será você. Então, use o PKCS#12 recém gerado para seu certificado e chave privada;
- assinar o documento `textoPlano.txt`, onde a assinatura deverá ser do tipo "anexada", ou seja, o documento estará embutido no arquivo de assinatura;
- gravar a assinatura em disco.

3.6 Sexta etapa — verificar uma assinatura digital

Por último, será necessário verificar a integridade da assinatura recém gerada. Note que o processo de validação de uma assinatura digital pode ser muito complexo, mas aqui o desafio será simples. Para verificar a assinatura será necessário apenas decifrar o valor da assinatura (resultante do processo de cifra do resumo criptográfico do arquivo `textoPlano.txt` com as informações da estrutura da assinatura) e comparar esse valor com o valor do resumo criptográfico do arquivo assinado. Como dito na fundamentação, para assinar é usada a chave privada, e para decifrar (verificar) é usada a chave pública.

Os pontos a serem verificados para essa etapa ser considerada concluída são os seguintes:

- verificar a assinatura gerada na etapa anterior, de acordo com o processo descrito, e apresentar esse resultado.

4 Implementação

A implementação deverá seguir algumas regras que serão levadas em conta na avaliação do desafio. O desafio estará comprometido se as regras não forem respeitadas. Essas regras são as seguintes:

- o desafio, codificado em Java, é apresentado como um projeto Maven. Sugere-se que a IDE utilizada seja o IntelliJ IDEA. A estrutura do projeto **não deve** divergir significativamente da sugerida;

- junto ao código, um **relatório** elaborado a partir das atividades desenvolvidas, ligando a fundamentação apresentada anteriormente com o código escrito. Note que as soluções para o desafio são conhecidas, mas este documento avaliará seu compreendimento destes conceitos. O documento deverá estar **em formato PDF, na mesma pasta deste**;
- a entrega deverá ser feita através da plataforma na qual este desafio foi fornecido. O arquivo entregue deverá **obrigatoriamente** ser o resultado do comando `mvn package`;
- o prazo de entrega é em duas semanas após o recebimento deste documento;
- se detectado plágio, o candidato estará **expulso** do processo de seleção. Caso alguma referência for utilizada, é necessário citá-la como parte da documentação do código.

Para ajudar a guiar no desenvolvimento do desafio foi elaborada uma estrutura contendo as possíveis classes, pacotes e locais para guardar os artefatos gerados durante a implementação. Nessa estrutura são dadas algumas dicas de como deve-se proceder para a boa execução do desafio.

Durante a implementação, se houver qualquer dúvida, problema, ou outra coisa que o impeça de continuar o desafio, entre em contato da forma que achar melhor. Lembre-se que esperamos que o candidato traga perguntas específicas e que demonstrem que ao menos tentou implementar as etapas propostas. Quanto mais genérica for a pergunta, mais genérica será a resposta. Geralmente as respostas virão na forma de referências a algo que responda a questão.

Os contatos são aqueles dados no início desse documento e/ou na plataforma específica para o processo seletivo. E caso ache realmente necessário conversar pessoalmente, então apareça no Laboratório ou use uma sala de videoconferência. Apenas envie um e-mail confirmando a disponibilidade. Boa sorte!

Referências

[DH76] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, September 1976.

[FSK11] N. Ferguson, B. Schneier, and T. Kohno. *Cryptography Engineering: Design Principles and Practical Applications*. 1st edition, 2011.

[Mer79] R. C. Merkle. *Secrecy, Authentication, and Public Key Systems*. PhD thesis, Stanford University, June 1979.

[MVvO96] A. J. Menezes, S. A. Vanstone, and P. C. van Oorschot. *Handbook of Applied Cryptography*. 1st edition, 1996.

[PP11] C. Paar and J. Pelzl. *Understanding Cryptography: A Textbook for Students and Practitioners*. 1st edition, 2011.

[Sta16] W. Stallings. *Cryptography and Network Security: Principles and Practice*. 7th edition, 2016.