Universidade da Beira Interior
Faculdade de Engenharia
Departamento de Informática

Propostas para Trabalhos de Grupo
Team Work Proposals

Sistemas de Software Seguros
Secure Software Systems

Departamento de Informática
Department of Computer Science
Universidade da Beira Interior
University of Beira Interior

Pedro R. M. Inácio
inacio@di.ubi.pt
2016/17

Conteúdo

1 Introdução

Introduction

As seguintes propostas de trabalho foram elaboradas no contexto da unidade curricular de Sistemas de Software Seguros do curso de mestrado em Engenharia Informática da Universidade da Beira Interior, e servem primariamente o propósito de melhorar a destreza dos seus executantes no que diz respeito ao desenho e desenvolvimento de sistemas de software em geral, à inclusão de aspetos de segurança durante a fase de projeto dessas aplicações, ao manuseamento e correta utilização de mecanismos da criptografia, e ao teste do sistema final em particular. Como tal, estas propostas não replicam necessariamente, ou em todo o detalhe, aplicações das tecnologias actualmente em uso.

The following proposals were elaborated within the context of the subject of Secure Software Systems of the masters degree on Computer Science and Engeneering of the University of Beira Interior, and their primary purpose it to improve the dexterity of the students concerning the design and development of software systems in general, the inclusion of security aspects during the engineering phase of those applications and the handling and correct utilization of cryptographic mechanisms, and to the testing of the resulting system in particular. As such, they may not all replicate real life applications of nowadays technologies.

Note também que os estudantes são livres de submeter uma nova proposta, desde que esteja em concordância com os objetivos da unidade curricular. Adicionalmente, a proposta deverá ser discutida com o regente antes de ser aceite como um tópico de trabalho de equipa válido.

Notice also that the student is free to submit a new proposal, as long as it is in accordance with the objectives of this subject. Additionally, the proposal has to be discussed with the Professor prior to being accepted as a valid teamwork topic.

Como de resto discutido durante as aulas, as equipas devem ter entre três e quatro elementos, e a implementação destas propostas deve ser complementada com um relatório em que se discutem as dificuldades encontradas ao longo do trabalho e se justificam todas as escolhas tomadas. Os trabalhos serão discutidos oralmente no final do semestre, recorrendo a um pequeno conjunto de diapositivos.

As discussed in the classes, the teams should be formed by three or four students, and the implementation of these proposals should be complemented with a report in which the difficulties faced along the work are discussed and the choices that have been taken are justified. The works will be orally discussed at the end of the semester, resorting to a small set of slides.

As aplicações podem ser implementadas recorrendo a qualquer estúdio ou ambiente de desenvolvimento integrado, sendo esse pormenor deixado ao critério do estudante. São também livres na escolha da plataforma alvo ou tecnologias *web* envolvidas, se aplicáveis, a não ser que os meios e tecnologias a utilizar estejam diretamente indicados na proposta). Só terão de ser levados em conta os dois apontamentos que se seguem: (i) não presuma nem esteja à espera que o Professor seja capaz de responder a todas as questões específicas às tecnologias que escolheu (por exemplo, não espere que ele saiba como trabalhar simultaneamente com o Netbeans, Eclipse, Android Studio, Xcode, etc.) e, (ii) a aplicação deve estar a funcionar corretamente na data de entrega, independentemente das ferramentas ou tecnologias utilizadas. Alguns dos problemas que a equipa encontrar durante o desenvolvimento destes projetos podem já ter sido tratados ou resolvidos nas aulas, e é livre de reutilizar, se aplicável, qualquer pedaço de código ou recurso desenvolvido ao longo das mesmas.

The applications can be implemented resorting to any development studio or integrated development environment of your choice. The team is also free to decide the programming language or the web technology, if applicable, that is going to be used in the development of the project, unless the means or technologies that should be used are directly specified in the proposal. Nonetheless, the following two remarks should be taken into consideration: (i) do not expect the Professor to be able to answer to all technology specific questions (e.g. do not expect him to know how to simultaneously work with Netbeans, Eclipse, Visual Studio, Android Studio, Xcode, etc.) and, (ii) the application should be correctly functioning by the time it is delivered. Some of the problems the team may face during the development of these projects may have been already addressed during the practical classes and,

as such, the team is free to reuse, if applicable, any code developed along those classes.

É da responsabilidade da equipa a pesquisa de detalhes específicos à solução dos problemas propostos, assim como de maneiras de testar a validade dessas soluções. Para além de incluir a solução do problema no relatório, deve também ser descrito o modo como foi validado o que foi feito (se aplicável) bem como alguma teoria de suporte.
The team has the responsibility to search for specific details concerning the proposal at hands and for ways to test and validate the implemented functionalities or solutions for the identified problems. Besides including the solution to the problem in the report, the means used to validate the developed work (if applicable) should also be described, along with some supporting theory.

O relatório deve conter uma ou mais secções (e.g., no capitulo de Engenharia de Software) em que são discutidos claramente os pontos relativos à segurança tidos em conta durante a fase de desenho e projeção da aplicação. Deve inclusivamente conter uma modelação prévia dos ataques a que a aplicação ou sistema desenvolvido pode estar afeto ou endereça (esta modelação é alvo de estudo numa aula – possivelmente a aula 8 ou 9 – desta unidade curricular), bem como a descrição dos testes feitos para garantir que a aplicação cumpria os objetivos de segurança. É obrigatória a inclusão do modelo do sistema, do modelo de ataque e das propriedades que devem ser garantidas, bem como uma discussão de como é que o sistema desenvolvido as garante.
The report shall contain one or more sections (e.g., in the chapter devoted to Software Engineering) for clearly discussing security related aspects that were taken into account during the design and projection phase of the application. It should also models for the attacks to which the application or system may be exposed to or that the implementation addresses (this modeling is the subject of study in one fo the lectures – probably lecture 8 or 9 – of this course unit), as well as a description of the tests that were performed to ensure that the application meets the security objectives. It is mandatory to include the system and threat models, and the security properties that should assured by design, along with a discussion on how does the system assures those properties.

1.1 Estrutura do Relatório
Structure for the Report

De modo a facilitar a estruturação do documento técnico que deve acompanhar o código e a aplicação desenvolvida no âmbito do trabalho a desenvolver, fica aqui uma sugestão para a estrutura do relatório. O relatório final é, contudo, responsabilidade da equipa, e pode conter mais ou menos capítulos ou secções do que as que são aqui indicadas, desde que apropriada e justificada:

1. Resumo
   Que é constituído por 2 frases onde introduzem o tema.
   2 frases (máximo) onde dizem como abordaram o tema.
   1 ou 2 frases onde se referem os objetivos ou resultados alcançados.
   Sugestão: ver `https://www.lightbluetouchpaper.org/2007/03/14/how-not-to-write-an-abstract/`
2. Introdução
   (a) Descrição do Problema
   Pequena secção onde descrevem o trabalho por palavras vossas. Não copiem o enunciado.
   (b) Constituição do Grupo
   (c) Organização do documento
   Exemplo:
   Este relatório está dividido em 5 capítulos principais:
   - O primeiro capítulo (Introdução) descreve o problema a tratar e os objetivos mais importates a alcançar...
   - O segundo capítulo (Desenvolvimento) ....

- ...
- ...

3. Engenharia de Software e da Segurança
   No início de cada capítulo deve ser dito como o capítulo está estruturado.
   (a) Análise dos Requisitos e Casos de Uso
   (b) Outros Diagramas (Diagramas de Classes, Diagramas Entidade Relacionamento, Componentes, etc.)
   (c) Modelo do Sistema, Modelo de Ataques e Propriedades de Segurança
   (d) Modelação de Ataques
   (e) Conclusão
4. Implementação
   (a) Ferramentas e Tecnologias Utilizadas
   (b) Escolhas de Implementação
   Explicar porque é que escolheram fazer de uma maneira, e não de outra.
   (c) Detalhes de Implementação
   (d) Manual de Instalação
   Secção simples a indicar como se compila (se aplicável) ou instala o sistema.
   (e) Manual de Utilização
   Secção simples onde se ilustra como se usa o sistema implementado.
   (f) Conclusão
5. Testes ao Sistema
   Este capítulo...
   (a) Testes de Segurança
   Descrever os testes de segurança a efetuar.
   (b) Resultados
   Descrever os resultados dos testes de segurança.
   (c) Conclusão
6. Reflexão Crítica e Problemas Encontrados
   Este capítulo...
   (a) Objetivos Propostos vs. Alcançados
   Descrever os objetivos que eram propostos e quais os que foram alcançados com exatidão.
   (b) Divisão de Trabalho pelos Elementos do Grupo
   Indicar as tarefas que cada membro do grupo fez.
   (c) Problemas Encontrados e Reflexão Crítica
   Problemas encontrados e resolvidos (ou não) durante a implementação.
   Refletir sobre o que foi conseguido e sobre o que poderia ser melhorado a nível pessoal e de trabalho de equipa. Apontar os problemas principais.
   (d) Conclusão
7. Conclusões e Trabalho Futuro
   (a) Conclusões Principais
   Texto muito analítico onde descrevem o que de melhor tiram deste trabalho, em termos técnicos. Sejam analíticos e sucintos.
   (b) Trabalho Futuro
   O que ficou por implementar.
8. Bibliografia

Nota: o relatório final não deve conter páginas em branco e o corpo do documento não deve ultrapassar as 20 páginas. Todas as figuras devem ser comentadas textualmente.

1.2 Notas a ter em Conta na Elaboração de um Bom Relatório
Remarks Concerning the Elaboration of the Report

1. Comecem sempre por incluir uma introdução onde descrevam o problema a tratar, o contexto e a estrutura do documento.
2. Elaborem bem no esqueleto do documento (secções, subsecções, etc.). Uma boa estruturação do relatório é 90% do caminho para obter um bom trabalho.
3. Caso tenham efetuado trabalho de pesquisa, incluam no relatório todas as referências.
4. Sejam breves e sucintos, mas elaborem nos detalhes que acharam importantes.
5. Procurem resolver todos os problemas que enfrentarem no tempo que possuem. Caso tal se demonstre impossível, discutam o falhanço com o mesmo afinco que discutiriam o sucesso.
6. Procurem incluir formas que ilustrem melhor o trabalho, nomeadamente gráficos, figuras ou tabelas.
7. Sejam pontuais. Um relatório entregue depois do prazo tem menos valor.
8. Implementem mecanismos que tenham aprendido nas aulas, só para mostrar que estudaram a matéria ou estiveram presentes.
9. Façam documentos com qualidade. Prestem atenção às regras da Língua em que escreve, nomeadamente pausas, sintaxe e semântica.

A folha de cálculo em `http://www.di.ubi.pt/~inacio/projeto/req-relatorio-v10.ods` também pode conter algumas dicas interessantes para a elaboração do relatório. Responder `SIM` a cada um dos critérios aí mencionados é meio caminho andado para uma boa nota na parte que se refere ao relatório.

1.3 Entrega do Trabalho
Delivery of the Works

Cada grupo deve entregar o trabalho na plataforma *Moodle* até às 23:55 do dia de entrega do trabalho e os nomes dos ficheiros deve seguir a especificação incluída na secção respetiva da unidade curricular, também no *Moodle*. O conjunto de elementos a entregar deve incluir: (i) o código fonte do sistema de software desenvolvido; (ii) scripts de instalação dos vários componentes; (iii) toda a documentação que o acompanhar, nomeadamente o relatório. Podem considerar disponibilizar contentores ou máquinas virtuais com os sistemas desenvolvidos pré-instalados em repositórios *online*. Por cada dia de atraso na entrega do trabalho, descontam-se 0,5 valores (aos 5).

1.4 Sugestão de Alinhamento para a Apresentação
Suggested Lineup for the Presentation

Como dito anteriormente, a defesa do trabalho deve ser acompanhada por um breve conjunto de diapositivos (nunca mais do que 10). A apresentação deve rondar os 15 minutos. Devem considerar fazer um conjunto de 7 a 9 slides para guiar o discurso com o seguinte alinhamento (adaptem):

- 1 diapositivo com o titulo do trabalho e elementos do grupo;
- 1 diapositivo com os objetivos do trabalho;
- 1 diapositivo dedicado à Engenharia do Software e da Segurança;

- 1 diapositivo dedicado(s) à implementação;
- 1 diapositivo dedicado à apresentação da aplicação (este diapositivo é só para lembrar para fazer o *switch* para um demonstrador real ou para um video da aplicação a correr);
- 1 diapositivo dedicado(s) à implementação;
- 1 diapositivo com a análise critica;
- 1 diapositivo dedicado aos objetivos alcançados / conclusões e trabalho futuro.
- 1 diapositivo a dizer `Bem haja pela atenção. Perguntas.`

A descrição das propostas é bastante breve e por vezes desprovida de detalhes, para que possam procurar alguns detalhes junto do docente e, simultaneamente, analisar e decidir sobre as soluções alternativas.

2 Vote como Paga (*codenamed*: BitVote)

2.1 Breve Introdução

A construção de um sistema de votos eletrónicos é dos que mais desafios apresenta, sendo difícil preencher, no mundo virtual e simultaneamente, todos os requisitos inerentes ao procedimento de votação tradicional (e manual). Este projeto tem como objetivo a construção de um sistema parecido com o da Bitcoin, mas para votos eletrónicos. A ideia é a de que, em vez de moedas, cada Bitcoin deve ser vista como um voto e que, antes de uma votação, seja dada exatamente 1 moeda a cada votante. Durante a votação, cada votante pode votar no candidato que deseja, transferindo-lhe o seu voto. É claro que todos os votos devem ser anónimos e intransmissíveis.

O sistema deve ser totalmente descentralizado, exatamente como na Bitcoin, e, portanto, depender do facto de todos os votantes contribuírem para confirmar os votos uns dos outros através de um processo parecido com mineração (*mining*), mas sem recompensa. Apesar da confirmação dos votos ser feita à medida que estes são feitos, deve ser estudada a forma do sistema garantir que ninguém sabe quem vai à frente antes da votação terminar. Este sistema pode ser feito para *Desktops* ou *smartphones* (e.g., Android), ou para ambos.

2.2 Funcionalidades Básicas

- O sistema permite a configuração de uma votação eletrónica com, pelo menos, dois candidatos.
- O sistema permite votos em branco ou nulos.
- As chaves privadas dos votantes são guardadas de forma segura.
- O sistema garante que os votos são anónimos.
- É garantido que os votos são intransmissíveis.

2.3 Funcionalidades Avançadas

- O sistema é multi-plataforma (i.e., o mesmo conjunto de aplicações funciona em diferentes sistemas operativos sem muitas modificações/configurações).

- É dada a garantia de que o vencedor só é conhecido após a votação terminar, e não durante a votação, embora a
- São usadas primitivas da criptografia em curvas elípticas.
- O sistema funciona em *smartphones*.
- Outras funcionalidades relevantes no contexto da segurança do sistema e que o favoreçam na nota.

3 Autenticador Centralizado Baseado em Certificados para o *Desktop* ou para Dispositivos Móveis (*codenamed*: TheCollector)

3.1 Breve Introdução

Hoje em dia, qualquer utilizador de um computador tem de saber lidar com, e gerir, uma ou mais credenciais de acesso, sendo as combinações de *nomes de utilizador*/*palavras-passe* as que mais habitualmente são utilizadas. Para ajudar a lidar com esta informação sensível existem programas que, no fundo, concretizam chaveiros virtuais, onde todas as credenciais estão guardadas por uma palavra ou frase-chave mestra. No fundo, o que esses programas fazem é guardar as credenciais cifradas com uma chave derivada da chave mestra. Atualmente, os próprios *browsers*, por motivos óbvios, integram chaveiros deste género com funcionalidades muito apelativas. O objetivo deste trabalho é construir um chaveiro em cuja autenticação é feita mediante certificados digitais ou, alternativamente, mediante um protocolo de conhecimento zero. Este chaveiro pode ser uma aplicação para a máquina local (os utilizadores confiam mais neste tipo de chaveiros) *Desktop* ou dispositivo móvel. Se optarem por uma aplicação *Desktop*, considerem utilizar o cartão do cidadão ou um dispositivo móvel (neste último caso, consultar o artigo *Strong Authentication with Quick Response Codes*) como forma de conseguir autenticação na aplicação. Caso optem por construir uma aplicação para telemóvel, considerem usar fatores como *someWHERE you are* e *something you know* para conseguir desbloquear o chaveiro. Ainda neste último caso, considerem produzir uma pequena aplicação para o *Desktop*, cujo único propósito é gerar códigos (e.g., *Quick Response* (QR)) para destrancar o chaveiro do telemóvel.

3.2 Funcionalidades Básicas

- O chaveiro eletrónico integra mecanismos de autenticação forte (assinaturas digitais ou CHAP).
- As credenciais de acesso são guardadas com uma cifra de chave simétrica de qualidade (e.g., AES), que deve depender do segredo de autenticação, e mudar sempre que o chaveiro for aberto!
- A integridade das credenciais de acesso é garantida por um mecanismo adequando de autenticação da origem da informação (e.g., HMAC).
- O sistema integra um mecanismo básico de geração de palavras-passe aleatoriamente.

3.3 Funcionalidades Avançadas

- A autenticação é conseguida através de um protocolo de conhecimento zero.

- O chaveiro eletrónico permite a configuração simples de parâmetros de segurança (e.g., o tamanho da chave de cifra a utilizar no AES, a função de *hash* a utilizar no HMAC ou assinatura digital, etc.).
- A integridade da base de dados de credenciais é garantida por um mecanismo de assinatura digital.
- A aplicação integra-se com o próprio sistema operativo para o qual foi desenhada, por exemplo, permitindo transferir um nome de utilizador ou palavras-passe para o *clipboard*, apagando-o automaticamente após algum tempo.
- Outras funcionalidades relevantes no contexto da segurança do sistema e que o favoreçam na nota.

4 Autenticador Centralizado Baseado em Certificados para a Web (*codenamed*: TheWebCollector)

4.1 Breve Introdução

Hoje em dia, qualquer utilizador de um computador tem de saber lidar com, e gerir, uma ou mais credenciais de acesso, sendo as combinações de *nomes de utilizador*/*palavras-passe* as que mais habitualmente são utilizadas. Para ajudar a lidar com esta informação sensível existem programas que, no fundo, concretizam chaveiros virtuais, onde todas as credenciais estão guardadas por uma palavra ou frase-chave mestra. No fundo, o que esses programas fazem é guardar as credenciais cifradas com uma chave derivada da chave mestra. Atualmente, os próprios *browsers*, por motivos óbvios, integram chaveiros deste género com funcionalidades muito apelativas. O objetivo deste trabalho é construir um chaveiro em cuja autenticação é feita mediante certificados digitais ou, alternativamente, mediante um protocolo de conhecimento zero. Este chaveiro pode ser uma Web App (+ *plug-in*), que permite, por exemplo, que um utilizador tenha acesso às suas credenciais através de *browsers* em qualquer lugar, desde que consigam configurar certificados digitais para autenticação mútua usando o protocolo *Transport Layer Security* (TLS) ou a implementar a lógica, do lado do cliente, para levar a cabo um protocolo de conhecimento zero. Considerem o uso do cartão do cidadão (e da sua funcionalidade de autenticação) como forma de desbloquear o chaveiro automaticamente. Considerem também a construção de um *plug-in* para um *browser* (e.g., Firefox) que permita, entre outras funcionalidades que considerem úteis, o preenchimento automático de formulários de autenticação em qualquer página através da vossa *web app*. O cenário de utilização é o seguinte:
Considerem que um utilizador fez o seu registo na vossa *web app* (ainda por desenvolver), tendo configurado o seu cartão do cidadão ou o seu dispositivo móvel (consultar o artigo *Strong Authentication with Quick Response Codes*) como forma de autenticação na aplicação. Numa segunda interação, ao abrir o seu *browser* e ao navegar até, e.g., site do Facebook, carrega no botão do *plug-in* da vossa *web app*. O *plug-in* pede o PIN de autenticação do cartão do cidadão e, após autenticação bem sucedida, preenche os campos de autenticação automaticamente.

4.2 Funcionalidades Básicas

- O chaveiro eletrónico integra mecanismos de autenticação forte (assinaturas digitais ou CHAP).
- As credenciais de acesso são guardadas com uma cifra de chave simétrica de qualidade (e.g., AES), que deve depender do segredo de autenticação, e mudar sempre que o chaveiro for aberto!

- A integridade das credenciais de acesso é garantida por um mecanismo adequando de autenticação da origem da informação (e.g., HMAC).
- O sistema integra um mecanismo básico de geração de palavras-passe aleatoriamente.

4.3 Funcionalidades Avançadas

- A autenticação é conseguida através de um protocolo de conhecimento zero.
- O chaveiro eletrónico permite a configuração simples de parâmetros de segurança (e.g., o tamanho da chave de cifra a utilizar no AES, a função de *hash* a utilizar no HMAC ou assinatura digital, etc.).
- A integridade da base de dados de credenciais é garantida por um mecanismo de assinatura digital.
- A aplicação integra-se com o *browser* para o qual foi desenhada de uma forma muito amigável para o utilizador, permitindo, por exemplo, preencher automaticamente os campos dos formulários de autenticação pré-configurados na *Web App*.
- O sistema integra um gerador de palavras-passe bastante elaborado e que permite, por exemplo, derivar palavras-passe de outras já utilizadas anteriormente e especificar os tipos de caracteres a utilizar.
- Outras funcionalidades relevantes no contexto da segurança do sistema e que o favoreçam na nota.

5 Um *Messenger* de Mensagens Curtas Cifradas para Dispositivos Móveis (*codenamed*: SecureShortMessageService)

5.1 Breve Introdução

São inúmeras as aplicações e sistemas existentes na atualidade para comunicação via mensagens curtas. Algumas dessas aplicações gabam-se da segurança que oferecem e servem-se dessa bandeira para conseguir mais utilizadores. Algumas aplicações de mensagens, focando nichos especiais, garantem o anonimato dos interlocutores, outras que as mensagens podem ser mais tarde apagadas com prova dada, etc.

O objetivo deste trabalho é construir um sistema de software que permita a troca segura de mensagens curtas entre utilizadores registados (mas eventualmente anónimos). Este sistema pode ser direcionado para dispositivos móveis ou para a web (ou ambos), e pode ser suportado pela Cloud. Para além da utilização de mecanismos do estado-da-arte para segurança das comunicações, deve o grupo considerar a integração de mecanismos de *oblivious transfer* (em que um emissor pode negar saber que mensagens enviou ou não) e protocolos de conhecimento zero para a autenticação.

5.2 Funcionalidades Básicas

- As aplicações integram mecanismos de autenticação forte (assinaturas digitais ou CHAP);
- As comunicações são seguras com algoritmos de cifra de chave simétrica seguros, corretamente implementados;

- A integridade das mensagens é assegurada um mecanismo adequando de autenticação da origem da informação (e.g., HMAC).
- O sistema garante que cada comunicação usa chaves de cifra e integridade efémeras, trocadas com o protocolo de acordo de chaves Diffie-Hellman (com mensagens assinadas digitalmente, para evitar ataques de homem no meio).

5.3 Funcionalidades Avançadas

- A autenticação é conseguida através de um protocolo de conhecimento zero;
- É usada criptografia de chave pública sobre curvas elípticas;
- Existe a possibilidade de fazer *oblivious transfer* (já há alguma aplicação deste género a permitir isto?);
- Outras funcionalidades relevantes no contexto da segurança do sistema e que o favoreçam na nota.

6 Moeda Criptográfica Controlada Centralmente (*codenamed*: fr€coin)

6.1 Breve Introdução

O objetivo principal deste projeto é o de criar uma moeda criptográfica. Ao contrário da BitCoin, esta moeda deve ser controlada centralmente, i.e., deve existir um sistema central (Entidade Central) que controla a quantidade de moeda existente e todos os gastos. Quando um novo utilizador se regista, deve ser gerado um par de chaves (e.g., RSA), cuja chave pública identifica as suas transações no sistema. Se um utilizador quiser enviar dinheiro a outro utilizador, deve criar um documento em que coloca a sua chave pública, a quantidade de freecoins que está a enviar e a chave pública do destinatário. O documento é assinado pelo emissor e enviado para a Entidade Central, que verifica se a transação se pode dar (i.e., se o emissor existe e tem dinheiro suficiente e se o destinatário existe). Caso a transação possa ser feita, assina também o documento e envia-o para o destinatário. Esta entidade central vai construindo um caderno de transações com todas as transações criadas até à data. Este caderno só é conhecido por essa entidade. Cada utilizador só deve conhecer (de forma legítima) as transações que efetuou ou as que refletem dinheiro que recebeu.

A criação da moeda deve ser feita come se sugere a seguir. A cada 30 segundos, a entidade central vai atribuir 1 fr€coin a quem lhe fizer lhe resolver primeiro um desafio. O desafio é simples: a entidade central gera um valor aleatório com b bits e envia esse valor para a rede. O primeiro utilizador que lhe enviar um ficheiro cujos primeiros b bits do valor de *hash* do SHA256 são iguais aos do desafio, ganha a moeda.

6.2 Funcionalidades Básicas

- O registo de um novo utilizador é feito de forma segura.
- Os pares de chaves são gerados do lado do utilizador.
- Todas as comunicações entre utilizadores e entidade central são feitas de forma segura (e.g., cifradas e protegidas por mecanismos de integridade).
- O sistema permite as transações conforme descritas na breve descrição.
- O sistema implementa a funcionalidade de geração de novas moedas.

6.3 Funcionalidades Avançadas

- O sistema suporta transações completamente anónimas entre utilizadores (i.e., geração de um par de chaves por cada transação);
- O sistema suporta autenticação mútua (cliente **e** servidor).
- A entidade central controla a emissão da moeda, ajustando o grau de dificuldade do problema à velocidade com a rede o resolve.
- A entidade central é também uma autoridade certificadora, e o sistema passa a estar suportado por uma infrastrutura de chave pública.
- O sistema é implementado com criptografia sobre curvas elípticas.
- Outras funcionalidades relevantes no contexto da segurança do sistema e que o favoreçam na nota.

7 Sistema de Sincronização de Ficheiros Seguro (*code-named*: TUDOIGUAL)

7.1 Breve Introdução

O objetivo principal deste projeto é criar um sistema de sincronização de ficheiros seguro. O sistema deve ser constituido por pelo menos duas componentes: uma aplicação cliente e outra aplicação servidor. Após se instalar a aplicação cliente num sistema, deve ser possível indicar uma diretoria cujos conteúdos são sincronizados com outra diretoria noutro computador. O servidor medeia as comunicações entre as duas máquinas.

O sistema deve garantir a segurança dos ficheiros armazenados na diretoria e transmitidos entre máquinas. Por exemplo, ao ser colocado um ficheiro dentro da diretoria num computador, esse ficheiro deve ser cifrado imediatamente. Deve também ser calculada uma assinatura digital do ficheiro, guardada localmente. O servidor deve ser notificado de um novo ficheiro na diretoria e, caso haja outras máquinas com a diretoria configurada e ligadas, deve motivar a troca do ficheiro entre essas duas máquinas. A ideia é que o conteúdo das diretorias seja igual sempre que as duas máquinas estejam ligadas. Obviamente, as aplicações cliente deve ter a certeza de estar a comunicar com a aplicação servidor correta, e a configuração de vários computadores deve ser feito de forma segura também, o que significa que deve haver alguma forma de registo (na primveira utilização) e autenticação (nas vezes seguintes) do utilizador que configura a máquinas.

7.2 Funcionalidades Básicas

- Na primeira utilização, a aplicação cliente gera um par de chaves para criptografia de chave pública (e.g., RSA). A chave privada desse par deve ser guardada apenas no servidor;
- Na primeira utilização, a aplicação cliente permite o registo de um novo utilizador junto do servidor;
- O registo de um novo utilizador é feito de uma forma segura (e.g., as comunicações são cifradas, é trocada uma chave efémera Diffie-Hellman ou usadas chaves RSA geradas aquando da instalação do servidor);
- Quando um ficheiro é colocado na diretoria de sincronização, é gerada uma chave de cifra e o ficheiro é cifrado com uma cifra de chave simétrica por blocos do estado da arte num modo adequado;

- A chave de cifra simétrica gerada no ponto anterior é cifrada com a chave RSA mencionada no primeiro ponto;
- Ao retirar um ficheiro da diretoria, deve ser feita autenticação do utilizador no servidor e decifrada a chave de cifra que permite decifrar o ficheiro;
- Ao colocar um ficheiro na diretoria, é calculada a sua assinatura digital com recurso ao servidor.

7.3 Funcionalidades Avançadas

- O sistema suporta a sincronização, via servidor, entre duas aplicações em dois computadores distintos;
- Ao transmitir um ficheiro entre dois computadores, este é partido em vários bocados, transmitidos individualmente com códigos MAC, que são verificados à chegada;
- O sistema é implementado com criptografia sobre curvas elípticas em vez de RSA;
- Outras funcionalidades relevantes no contexto da segurança do sistema e que o favoreçam na nota.