PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Safe Vault, sistema para proteção de pastas e arquivos

Thiago Lages de Alencar

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC

DEPARTAMENTO DE INFORMÁTICA

Curso de Graduação em Engenharia da Computação

Rio de Janeiro, mês de ano

Thiago Lages de Alencar

Safe Vault, sistema para proteção de pastas e arquivos

Proposta de Projeto Final 1, apresentado para **Bacharelado em Ciência da Computação** da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Anderson Oliveira da Silva.

Rio de Janeiro

Setembro de 2018.

1. Introdução

Serviços de armazenamento como Google Drive[12], Dropbox[13] e Onedrive[14], permitem ao usuário o conforto de acesso remoto às suas informações, sem se preocupar com a manutenção recursos físicos. Porém existem usuários que se recusam a armazenar informações sensíveis nesses provedores, pois consideram a proteção insuficiente. Portanto observa-se que, a confidencialidade é uma grande preocupação quando se fala de armazenamento na internet[1][2].

O armazenamento na nuvem pode aumentar os riscos de comprometimento das informações de diversas maneiras, como por exemplo: vazamento de informação, furto de conta, fraudes, uso indevido, etc[2][3]. Esses riscos podem ser reduzidos se os dados forem protegidos utilizando protocolos de segurança. Tais protocolos dão ao usuário controle sobre quem pode acessar os dados, e isso garante a proteção das informações do usuário.

Diante desse cenário, o sistema proposto visa a funcionar como uma ferramenta para o usuário armazenar de forma segura, seus arquivos em serviços de armazenamento ou no próprio dispositivo que está utilizando. A ideia é que o sistema seja apenas uma ponte para o usuário garantir a segurança de seus dados nos serviços de armazenamento na nuvem.

2. Situação Atual

Grande parte dos serviços de armazenamento oferecem o armazenamento não criptografado, o que implica no usuário confiar que suas informações vão ter a confidencialidade respeitada. Não existe certeza dessa confidencialidade, pois se alguém obter a credencial de acesso do usuário, toda a informação no provedor do serviço vai estar comprometida.

Provedores e empresas de terceiros que garantem a segurança, costumam armazenar a chave que protege as informações do usuário[6]. Isso delega o controle do usuário para os provedores e empresas, não resolvendo o problema de confidencialidade[1].

O sistema proposto dá ao usuário a chave capaz de proteger a informação, assim garantindo a confidencialidade das informações. Uma desvantagem é que o serviço de armazenamento não poderá fornecer recursos como: busca, modificação, inserção e remoção sobre as informações seguras[1]. Essas facilidades devem ser implementadas pela ferramenta que controla a proteção dos dados.

Para prover a segurança das informações do usuário, deve-se utilizar algoritmos de criptografia. A informação deve ser protegida por cifragem de forma que mesmo se uma pessoa maliciosa obtiver os dados cifrados, será muito difícil decifrar sem ter a chave de criptografia[2]. A segurança da informação é garantida pelos seguintes princípios básicos: *Integridade*, *Autenticidade*, *Confidencialidade e Disponibilidade*.

Integridade envolve proteger a informação de alterações sem permissão explícita do proprietário[2]. O estado da informação quando resgatada, deve ser igual ao estado quando gerada[5]. Algoritmos de resumo de mensagem (digest) são utilizados para verificar a integridade das informações[2].

Autenticidade consiste em identificar corretamente um usuário ou computador[2]. É uma maneira de medir o grau de confiança de que a origem da informação é a mesma que ela alega ser[7]. É necessário verificar a autenticidade em todo processo de identificação e transferência de informação. Através da assinatura digital pode-se validar a autenticidade das informações[2].

Confidencialidade garante que apenas as pessoas às quais a informação é destinada conseguem compreendê-la. Utilizando algoritmos de criptografia, é possível mascarar a informação original, através de cifragem. A segurança depende da garantia de segredo da chave utilizada no algoritmo.

Disponibilidade visa a manter a informação disponível para uso sempre que houver necessidade dela. O serviço de armazenamento compartilha desse princípio básico.

Tecnologías Utilizadas

O desenvolvimento do projeto será feito utilizando a linguagem de programação *Java*, versão *SE 10*, e os provedores criptográficos disponíveis na *Java Cryptography Architecture* (JCA).

O ambiente de desenvolvimento será o *Eclipse*, versão *Eclipse Photon 27 June 2018*.

3. Proposta e Objetivos do trabalho

O trabalho visa a implementar uma ferramenta que permita garantir a segurança da informação do usuário em serviços de armazenamento, de forma a garantir os princípios de integridade, autenticidade e confidencialidade. A ferramenta é separada em três partes: *Engine*, *Plugin* e *Interface*.

A Engine será responsável por aplicar todos os protocolos de segurança nas informações do usuário, receber requisições das interfaces e fazer requisições aos plugins. A Engine deve ser única para todas as plataformas e deve interagir com as interfaces de usuário dos diferentes sistemas operacionais e com os serviços de armazenamento (através de plugin). Para garantir a segurança da informação, a Engine fará uso de algoritmos de hash (digest), padrões de assinatura digital e algoritmos criptográficos simétricos e assimétricos.

O Plugin deve interagir com o serviço de armazenamento para realizar tarefas requisitadas pela Engine. Cada Plugin fica responsável por se comunicar com um dos provedores de armazenamento. A Engine faz requisições padronizadas independente do Plugin para executar operações básicas no serviço de armazenamento, como: criar arquivo, deletar arquivo, ler arquivo, escrever arquivo, criar pasta, listar pasta e deletar pasta. Um Plugin padrão será implementado para interagir com o sistema de arquivos local na máquina do usuário.

A Interface é o meio de comunicação do usuário com a Engine, e vice-versa. Através dela o usuário descobre o estado das informações na nuvem e solicita ações à Engine. Todas as interfaces de usuário, independentemente de sistema operacional, solicitam as mesmas ações de forma padronizada para a Engine.

4. Plano de Ação

Para fornecer a segurança da informação aos usuários, será necessário estudar os conceitos de integridade, autenticidade e confidencialidade[2], e as técnicas que garantem a funcionalidade desses conceitos, como: *resumo de mensagem*, *assinatura digital*, *criptografia simétrica* e *assimétrica*[2]. A Engine será construída com base na *Java Cryptography Architecture* (JCA)[9]. Os Plugins requerem o estudo das API dos diversos serviços de armazenamento[10][11]. O processo de criação do software é incremental e engloba as seguintes atividades:

1. Estudo dos conceitos: integridade, autenticidade, confidencialidade, resumo de mensagem, assinatura digital, envelope digital, criptografia simétrica e assimétrica.
2. Teste de algoritmos criptográficos simétricos e assimétricos. Cifrar e decifrar informações.
3. Teste de resumo de mensagem e assinatura digital.
4. Teste de armazenamento e recuperação de informação cifrada na máquina do usuário, armazenar localmente.
5. Estudo da Google Drive API.
6. Teste armazenamento e recuperação de informação cifrada no serviço Google Drive.
7. Teste criptografar e armazenar no Google Drive.
8. Padrões e boas práticas que vão ser utilizada no código do sistema.
9. Diagramas da engine, plugin e interface.
10. Desenvolvimento da engine.
11. Teste da interação da engine com as informações.
12. Desenvolvimento do plugin padrão, armazenamento local.
13. Teste da interação da engine com o plugin padrão.
14. Desenvolvimento do plugin do Google Drive.
15. Teste da interação da engine com o plugin do Google Drive.
16. Desenvolvimento da interface, *Java Graphic User Interface* (GUI).
17. Teste da interface com a engine, utilizando o plugin padrão.

18. Teste da interface com a engine, utilizando o plugin do Google Drive.

A previsão de execução dessas atividades é 7 semanas de estudo e 11 semanas de desenvolvimento, conforme mostrado no cronograma a seguir:

	SEMANA																	
ATIVIDADE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	x	x	x	x														
2		x	x	x														x
3		x	x	x														x
4				x	x													x
5					x	x	x											
6						x	x											
7							x											
8								x	x	x		x		x		x	x	
9									x	x		x		x		x		
10										x	x							
11											x							x
12												x	x					
13													x					x
14														x	x			
15															x			x
16																x	x	
17																	x	x
18																		x

5. Referências bibliográficas

[1] YUSUF HAIDER M, SIVA SELVAN. **Confidentiality Issues in Cloud Computing and Countermeasures: A Survey**. Departamente of Computer Science and Engineering. Manipal Institute of Technology (Manipal University). Disponível em: <https://www.researchgate.net/publication/305689086_Confidentiality_Issues_in_Cloud_Computing_and_Countermeasures_A_Survey>

[2] ALFREDO KURY ABÍLIO, ANDRÉ CONCEIÇÃO DA GRAÇA, CRISTIANO PEDRO DA SILVA, MAURO SÉRGIO DOS SANTOS AMORIM. **Comunicação Segura na Internet: Métodos, Infra-estrutura de Chaves Públicas e Padrões**. Trabalho de Conclusão de Curso.

[3] PRADEEP KUMAR TIWARI. **Cloud Computing Security Issues, Challenges and Solution**. International Journal of Emerging Technology and Advanced Engineering, 2012. Disponível em:

<https://www.researchgate.net/publication/271522943/download>

[4] MAHESH U. SHANKARWAR, AMBIKA V. PAWAR. **Security and Privacy in Cloud Computing: A Survey**. CSE Department, SIT, Symbiosis International University, 2015. Disponível em: <https://www.researchgate.net/publication/282925703/download>

[5] CAROLINA GWOZDZ POERSCH, GIULLYAN METZKER KUNTZE. **Modelo de Coleta e Análise de Evidências em Sistemas Computacionais**. Universidade do Sul de Santa Catarina, 2010. Disponível em: <https://www.riuni.unisul.br/bitstream/handle/12345/3229/100889_Carolina.pdf?sequence=1>

[6] Google Cloud. **How Google Uses Encryption to Protect Your Data**. Acessado em: 13 Set. 2018. Disponível em:

<https://storage.googleapis.com/gfw-touched-accounts-pdfs/google-encryption-whitepaper-gsuite.pdf>

[7] SEAN PEISERT, ED TALBOT, TOM KROEGER. **Principles of Authentication**. California, USA, 2013. Disponível em: <https://www.nspw.org/papers/2013/nspw2013-peisert.pdf>

[8] Oracle. **The Java™ Tutorials**. Acessado em: 13 Set. 2018 <https://docs.oracle.com/javase/tutorial/index.html>

[9] Oracle. **Java Cryptography Architecture (JCA) Reference Guide**. Acessado em: 13 Set. 2018 <https://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html>

[10] Google. **Google Drive APIs REST**. Acessado em: 13 Set. 2018 <https://developers.google.com/drive/api/v3/about-sdk>

[11] Google. **Google API Client Libraries**. Acessado em: 13 Set. 2018 <https://developers.google.com/api-client-library/>

[12] Google Drive. **Google**. Disponível em: <https://www.google.com/drive/>

[13] Dropbox. **Dropbox Inc**. Disponível em: <https://www.dropbox.com>

[14] OneDrive. **Microsoft**. Disponível em: <https://onedrive.live.com>