Sistema de Comunicação Anônima com Utilização de Advanced Encryption Standard(AES) e Rivest-Shamir-Adleman(RSA).

Andrey Carvalho, Eduardo Britto da Costa, Morgana Silva Prado, e Rodrigo Valente Bernardes
Curso de Eng. da Computação Universidade Tecnológica Federal do Paraná.

***Resumo*—Neste artigo é proposto a implementação de um sistema de comunicação instantânea criptografado para resolver um possível problema de interceptação de mensagens.**

***Index Terms*—AES - Comunicação - Criptografia - RSA - Segurança**

I. Problema

A interceptação de mensagens em sistemas de comunicação online.

II. Introdução

ESTE relatório tem por fim descrever as etapas da realização do projeto assim como os produtos resultantes dessas etapas. Na seção II será apresentada as ferramentas e procedimentos utilizados para a implementação do sistema proposto. Na seção III serão apresentados os resultados obtidos, ou seja, relatos do funcionamento do sistema. Na quarta e última seção serão apresentados os apontamentos feitos após o término do desenvolvimento do sistema. Esses apontamentos incluem quais foram as funcionalidades propostas que tiveram implementação bem sucedida, quais não tiveram, quais outras funcionalidades poderiam ser implementadas posteriormente, pontos fracos e pontos fortes do sistema desenvolvido. Ao final do relatório estão as referências utilizadas para o desenvolvimento do projeto.

III. Método

O projeto de chat anônimo possui alguns módulos. O projeto foi dividido entre comunicação do servidor, comunicação e interação com o banco de dados, sistema de criptografia de dados, e interface de usuário. O controle das versões se deu por meio da ferramenta GitHub, e as implementações foram concebidas utilizando a linguagem Python de programação.

O projeto possui 20 arquivos no total, sem exclusões. Alguns não são parte da implementação do sistema, mas estão presentes na ferramenta de controle de versões. Abaixo segue uma breve descrição dos principais objetivos de cada arquivo funcional no projeto.

- .gitignore - mostra os arquivos que serão desconsiderados da implementação do projeto.
- src/DDL_BD.sql - contém a implementação do banco de dados a ser utilizado no sistema.
- src/dml.sql - contém a implementação de população do banco.
- src/main.py - implementada a inicialização do sistema.
- src/server.py - implementa funções de **thread** para o login e para a lista de amigos online, assim como funções de funcionamento do próprio servidor.
- src/modules/conversationDAO.py - contém a implementação das **querys** utilizadas para alterações dos dados de conversas no banco de dados.
- src/modules/crypto.py - implementa as funções da parte de criptografia, onde estão as funções de criptografia e descriptografia em RSA, por exemplo.
- src/modules/database.py - implementa as funções inerentes ao próprio banco de dados, como o **commit**, por exemplo, além de implementar método de conexão com o banco.
- src/modules/key_setDAO.py - contém os mé-

todos que realizam alterações no na tabela **key_set** do banco de dados.
- src/modules/message.py - contém os métodos de classe **model** padrão além dos métodos de criptografia e descriptografia de mensagem.
- src/modules/messageDAO.py - contém os métodos de alteração da tabela **message** no banco de dados.
- src/modules/user.py - apresenta os métodos que irão utilizar os métodos da classe **CryptoEngine** para gerar as chaves assimétricas e distribuí-las para os usuários, além de gerar um diretório com **hash** para armazenar cada conversa.
- src/modules/userDAO.py - essa classe contém os métodos que serão utilizados para alteração de dados da tabela **chat_user**. Cada método possui uma rotina que implementa **querys sql**.
- src/modules/user_relationDAO.py - possui métodos que implementam **querys sql** assim como a classe UserDAO, porém possui como alvo de alteração as tabelas **user_relation**, **chat_user** e **relation_type**.
- src/modules/gui/chat_gui.py - esse arquivo contém as classes chat_window e sender_thread. Na primeira classe estão os métodos que implementam a janela de chat do usuário assim como as ações de cada componente da mesma. A segunda classe contém o método que estabelece conexão com o servidor e exibe as mensagens enviadas.
- src/modules/gui/friends_gui.py - a classe friends_list possui os métodos de inicialização dos componentes da janela da lista de amigos e as ações abrir conversa e fechar/sair. A classe chat_friend_thread implementa a inicialização da janela como **thread**.
- src/modules/gui/login.py - seus métodos implementam a inicialização da janela **login** como **thread** assim como seus componentes e também implementa a validação dos dados de login, verificando antes de proceder com a conexão.
- /src/modules/utils/files.py - contém os métodos de leitura e escrita em arquivo, que serão utilizados na classe ChatUser, por exemplo.

As implementações foram realizadas utilizando editores de texto como VIm em sistemas operacionais base Linux. A utilização de outro sistema operacional ou editor de texto não necessariamente afetaria o andamento do projeto.

IV. Resultados Obtidos

A. Sistema Desenvolvido

ESTA sessão irá apresentar os resultados obtidos com o desenrolar do trabalho. E como proposto foi realizado o desenvolvimento de um sistema de comunicação instantâneo que fosse criptografado.

Na Figura 1 é apresentado o login do sistema, onde os usuários devem colocar seu nome e sua senha. No momento que o login é realizado um par de chaves (RSA) é criada, caso o usuário não exista, um novo usuário é criado em disco assim como a sua chave privada, e a chave pública é salvado no banco de dados.

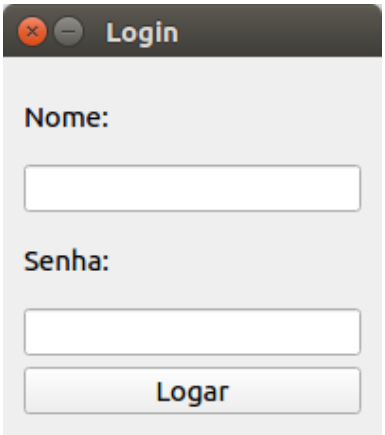

Figura 1. Área de Login do sistema

Já na Figura 2 é apresentado os usuários que estão utilizando o sistema e que podem ser selecionados para se iniciar a troca de mensagem. Sendo assim, para abrir a janela de comunicação será feita a troca de mensagens com Teste 0, que está selecionado na Figura 2.

Figura 2. Área de amigos de um usuário do sistema

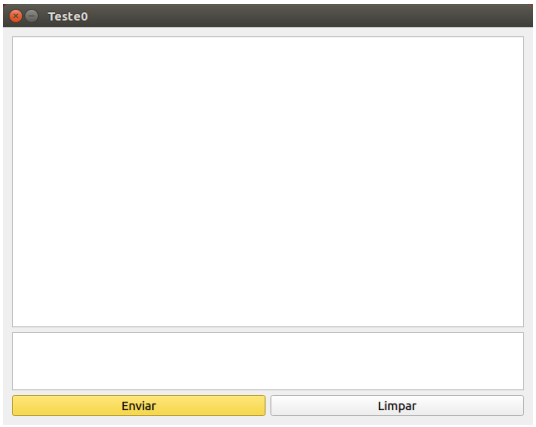

Figura 3. Área de amigos do sistema

E por fim, a Figura 3 é demostrado o chat com o usuário Teste 0, as mensagens que são compartilhadas nessa conversa são criptografadas com AES (Advanced Encryption Standard), e todas as mensagem serão armazenadas no banco de dados e no fim da comunicação todas essas mensagem armazenadas anteriormente, serão apagadas do banco de dados, tendo então uma troca de mensagens instantânea.

B. Testando o Sistema

FOI utilizado o software Wireshark para realihttps://pt.sharelatex.com/project/5b2135bffd7b7422734d29c3 zar testes no sistema, fazendo um sniffer nos pacotes trocados.

Realizando esses testes, foi possível ver que os pacotes trocados no momento do login do usuário utilizam criptografia que é visto pela Figura 4. Com a análise do pacote é possível ver a criptografia gerada é uma criptografia assimétrica (RSA), e com isso todos os dados do usuário estão protegidos e são gerados dois pares de chaves.

Figura 4. Teste de login no Sistema.

Como foi feito no teste no login, foi realizado um teste no momento de requisição dos usuários que utilizam o sistema e estes dados são retornados do servidor de forma criptografada como pode ser visto na Figura 5, de forma que nenhuma pessoa consiga identificar dados dos outros usuários.

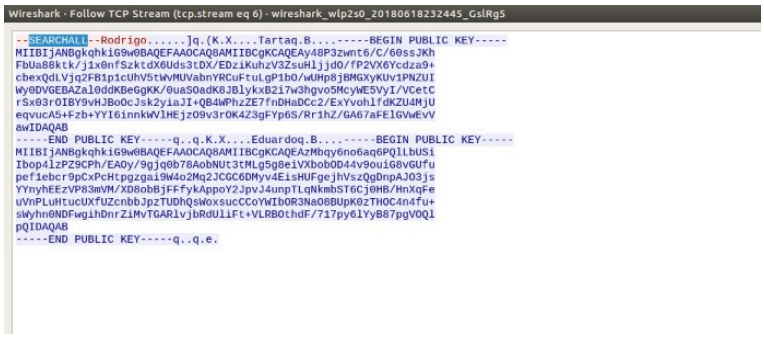

Figura 5. Teste da requisição de amigos do Servidor.

Já na Figura 6 é possível analisar que as mensagens que são enviadas de um usuário ao outro de forma sigilosa, utilizando criptografia simétrica para manter as trocas de mensagem segura para todos os usuários que utilizam o sistema.

Figura 6. Teste no envio da Mensagem.

Na Figura 7 temos o teste na requisição das mensagens que foram enviadas, e neste teste é possível ver também que as mensagem são retornadas do

servidor de forma sigilosa e segura ao usuário. Desta forma todo sistema esta protegido por criptografia, garantindo ao usuário sigilo na troca de mensagens.

Figura 7. Teste da requisição de amigos do Servidor.

Com esses testes feitos no sistema, estamos solucionando o problema que foi abordado neste projeto, onde as trocas de mensagens em sistemas de comunicação não são confiáveis. Obtemos então êxitos nos resultados esperados com o desenvolvimento do projeto.

V. Conclusão

Ao fim deste projeto, alguns pontos devem ser destacados. Em primeiro lugar, é que o projeto foi realizado com êxito segundo o que foi proposto inicialmente. Já em segundo lugar, tivemos um retorno esperado com o projeto, podendo portanto apresenta-los a partir deste relatório com qualidade. E por fim, adquirimos uma aprendizagem maior a respeito dos métodos de criptografia, tendo como destaque as criptografias simétricas e assimétricas, que foram utilizadas para o desenvolvimento deste projeto, e aprendemos como incorpora-las no nosso sistema de forma correta.

Referências

[1] Desconhecido. Advanced encryption standard. [Online]. Available: https://en.wikipedia.org/wiki/AdvancedEncryptionStandard.

[2] Desconhecido. RSA (Rivest-Shamir-Adleman)[Online]. Available: https://pt.wikipedia.org/wiki/RSA.