
Curso de Ciência da Computação
Universidade Estadual de Mato Grosso do Sul

**PROTÓTIPO DE UM SISTEMA EXPLORADOR DE ARQUIVOS
BASEADO EM CRIPTOGRAFIA**

Douglas Correia Salazar e Jonas Forte Silva

Prof. Msc. Adriana Betânia de Paula Molgora (Orientadora)

Prof. Dr. Fabrício Sérgio de Paula (Co-Orientador)

DOURADOS-MS

2016

PROTÓTIPO DE UM SISTEMA EXPLORADOR DE ARQUIVOS BASEADO EM CRIPTOGRAFIA

Douglas Correia Salazar e Jonas Forte Silva

Este exemplar corresponde à redação final da monografia da disciplina Projeto Final de Curso devidamente corrigida e defendida por Douglas Correia Salazar e Jonas Forte Silva e aprovada pela Banca Examinadora, como parte dos requisitos para a obtenção do título de Bacharel em Ciência da Computação.

DOURADOS-MS 2016

Prof. Msc. Adriana Betânia de Paula Molgora
(Orientadora)

Prof. Dr. Fabrício Sérgio de Paula
(Co-Orientador)

**PROTÓTIPO DE UM SISTEMA EXPLORADOR DE ARQUIVOS
BASEADO EM CRIPTOGRAFIA**

Douglas Correia Salazar e Jonas Forte Silva

Novembro 2016

Banca Examinadora:

Prof. Msc. Adriana Betânia de Paula Molgora (Orientadora)

Ciência da Computação

Prof. Dr. Nilton César de Paula

Ciência da Computação

Prof. Dr. Odival Faccenda

Ciência da Computação

AGRADECIMENTO

Agradeço primeiramente a Deus, pois sem Ele jamais chegaria até aqui.

Aos meus pais, Maria Cristina e Osmar Francisco que me proporcionaram apoio durante minha graduação. Aos meus amigos que sempre me ajudaram em tudo durante a graduação e que me incentivaram a continuar.

Aos Professores, Adriana Betânia de Paula Molgora e Fabrício Sérgio de Paula pela orientação e incentivo dado para o término desse trabalho.

E a todos os demais Professores que contribuíram para o sucesso desse trabalho.

Jonas Forte Silva

Agradeço primeiramente a Deus, pois graças a Ele que tudo se tornou possível.

Aos meus pais, Regina Aparecida Correia e Candido Salazar que sempre me apoiaram, se esforçaram e não mediram esforços para que eu chegasse até a esta etapa de minha vida.

Aos meus amigos que sempre me ajudaram em tudo durante a graduação e que me incentivaram a continuar.

Aos Professores, Adriana Betânia de Paula Molgora e Fabrício Sérgio de Paula pela orientação e incentivo que possibilitou o término desse trabalho.

E a todos os demais Colegas e Professores que contribuíram para o sucesso desse trabalho.

Douglas Correia Salazar

RESUMO

A criptografia pode ser definida como a arte de codificar e decodificar uma mensagem. Há muito tempo, o homem sente a necessidade do sigilo de informações e, desde então, muitos métodos de criptografia foram desenvolvidos para auxiliar neste sigilo. Dentre os métodos criptográficos encontra-se a criptografia ECC (Elliptic Curve Cryptography) baseada nas curvas elípticas e o AES (Advanced Encryption Standard). Esses foram os métodos desenvolvidos e estudados neste trabalho.

O método ECC é uma técnica de criptografia de chave pública ou assimétrica, primeiramente proposta por Victor Miller e Neal Koblitz em 1985. Ela tornou-se uma atraente técnica criptográfica pelo fato de seus criadores assegurarem que esse método consegue obter o mesmo nível de segurança com chaves consideravelmente menores comparado a outros métodos.

Em uma necessidade de um novo método criptográfico, o NIST lançou um concurso, onde o método Rijndael se tornou a criptografia oficial do NIST, originando o AES. Esse é um método de criptografia simétrica amplamente usado em todo o mundo.

Este trabalho apresenta um protótipo de um sistema explorador de arquivos que utiliza ambos os métodos citados. O protótipo desenvolvido tem como objetivo oferecer um ambiente de proteção de arquivos pessoais com um uso prático e intuitivo para usuários de computador que desejam privacidade e segurança, usufruindo da criptografia como mecanismo de proteção.

Palavras-Chave: Criptografia. Curvas elípticas. AES, ECC.

ABSTRACT

Encryption can be defined as an art encode and decode a message. Long ago, the man feels a need for information confidentiality, and since then, many encryption methods were developed to assist this secrecy. Among cryptographic methods were developed, the based encryption in elliptic curves ECC (Elliptic Curve Cryptography) and AES (Advanced Encryption Standard).

The ECC method, is a public key cryptography technique or asymmetric technique. First proposed by Victor Miller and Neal Koblitz in 1985. Became an attractive cryptographic technique, because the fact it creators ensured the method can get the same level security with considerably minor keys.

In a need for a new cryptographic method NIST launched a league where the Rijndael method became an official NIST encryption, generating AES. Symmetric encryption method widely used throughout the world.

This paper presents a prototype of an explorer file system that uses both cited methods. The prototype aims to offer a personal file protection environment with a practical and intuitive use for computer users who want privacy and security, taking advantage of encryption as a protection mechanism.

Keywords: Cryptography. Elliptic curves. AES.

SUMÁRIO

1. INTRODUÇÃO	19
2. FUNDAMENTOS TEÓRICOS MATEMÁTICOS	21
2.1 TEORIA DOS NÚMEROS	21
2.1.1 DIVISIBILIDADE	21
2.1.2 MÁXIMO DIVISOR COMUM	21
2.1.3 NÚMEROS PRIMOS	22
2.1.4 CONGRUÊNCIA	22
2.1.5 INTEIROS MÓDULOS M.....	23
2.2 GRUPOS, ANÉIS E CORPOS.....	23
2.2.1 GRUPOS	23
2.2.2 ANÉIS.....	23
2.2.3 CORPOS	24
2.3 CURVAS ELÍPTICAS.....	24
2.3.2 OPERAÇÃO DE ADIÇÃO	25
2.3.3 PROPRIEDADE DA OPERAÇÃO DE ADIÇÃO	26
2.3.4 FÓRMULAS DE ADIÇÃO DE PONTOS EM CURVAS ELÍPTICAS.....	27
2.3.5 DUPLICAÇÃO DE PONTO	27
2.3.6 MULTIPLICAÇÃO DE PONTOS	27
2.3.7 CURVAS SOBRE CORPOS FINITOS.....	28
2.3.8 CURVAS ELÍPTICAS RECOMENDADAS	29
3. CRIPTOGRAFIA.....	31
3.1 MÉTODO CRIPTOGRÁFICO ECC.....	31
3.1.1 PROBLEMA DO LOGARITMO DISCRETO	31
3.1.2 DIFFIE-HELLMAN	31
3.1.2.1 DIFFIE-HELLMAN NAS CURVAS ELÍPTICAS	32
3.2 MÉTODO CRIPTOGRÁFICO AES.....	32
3.2.1 PARÂMETROS DO AES.....	32
3.2.2 ESTRUTURA GERAL DO AES	33
3.2.3 TRANSFORMAÇÃO SUBBYTES.....	33
3.2.4 TRANSFORMAÇÃO SHIFROWS	34
3.2.5 TRANSFORMAÇÃO MIXCOLUMNS.....	35
3.2.6 TRANSFORMAÇÃO ADDROUNDKEY	36
4. IMPLEMENTAÇÃO	37
4.1 CRIPTOGRAFIA.....	37

4.2	BANCO DE DADOS.....	39
4.3	INTERFACE GRÁFICA.....	40
4.3.1	TELA DE LOGIN.....	41
4.3.2	TELA PRINCIPAL.....	42
4.3.3	FUNCIONALIDADE.....	43
4.3.4	SELECIONAR ARQUIVOS.....	43
4.3.5	CRIPTOGRAFAR.....	45
4.3.6	DESCRIPTOGRAFAR.....	45
4.3.7	TROCA DE INFORMAÇÕES.....	46
4.3.8	COMPARTILHAR ARQUIVO.....	47
5.	CONSIDERAÇÕES FINAIS.....	49
6.	FUTURAS IMPLEMENTAÇÕES.....	50
	REFERÊNCIAS BIBLIOGRÁFICAS.....	51
	APÊNDICE A.....	53
	APÊNDICE B.....	54
	APÊNDICE C.....	55
	APÊNDICE D.....	56
	APÊNDICE E.....	57
	APÊNDICE F.....	58
	APÊNDICE G.....	59
	APÊNDICE H.....	60

LISTA DE FIGURAS

Figura 1 – Exemplos de Curvas elípticas	25
Figura 2 - Adição na curva.....	26
Figura 3 - Adição de pontos na curva, sendo o elemento neutro como o ponto no infinito.....	26
Figura 4 - Transformação direta SubBytes.....	34
Figura 5 - Transformação direta ShiftRow.....	34
Figura 6 - Operações de linha ShiftRows.....	35
Figura 7 - Transformação MixColumns.....	36
Figura 8 - Transformação AddRoundKey.....	36
Figura 9 - Divisão de arquivo sem resto.....	38
Figura 10 - Divisão do arquivo	38
Figura 11 - Tela Login	41
Figura 12 - Cadastro de Usuários	42
Figura 13 - Recuperação de Senha	42
Figura 14 - Tela Principal.....	43
Figura 15 - Selecionar Arquivo Criptografado.....	44
Figura 16 - Selecionar Arquivos Descriptografados	44
Figura 17 - Selecionar Arquivo	45
Figura 18 - Aguarde	45
Figura 19 - Aguarde	46
Figura 20 - Troca de Dados.....	46
Figura 21 - Trocar Senha.....	46
Figura 22 - Selecionar Arquivo	47
Figura 23 - Compartilhar.....	48
Figura 24 - Compartilhar.....	48

LISTA DE TABELAS

Tabela 1 - Parâmetros do AES	33
Tabela 2 - Tabela Usuários.....	39
Tabela 3 - Tabela Compartilhar	40

LISTA DE EQUAÇÕES

Equação 1 - Equação Cúbica.....	24
Equação 2 - Forma normal de Weierstrass.....	25
Equação 3 - Forma normal de Weierstrass.....	25
Equação 4 - Adição de pontos.....	27
Equação 5 - Duplicação de Pontos.....	27
Equação 6 – Multiplicação de pontos.	28
Equação 7 - Curva sobre um corpo finito.....	28
Equação 8 - Adição de Pontos.....	29
Equação 9 - Duplicação de Pontos.....	29
Equação 10 - Multiplicação de matrizes MixColumns.....	35
Equação 11 - Única coluna em MixColumns.....	35

LISTA DE ABREVIACOES E SIGLAS

ECC - Elliptic Curve Cryptography

AES - Advanced Encryption Standard

1. INTRODUÇÃO

Desde a antiguidade, o homem tem uma grande necessidade de estabelecer sigilo em troca de informações. E, com a evolução tecnológica e o surgimento dos computadores essa necessidade tem aumentado ainda mais. Uma grande quantidade de informações pessoais é armazenada e manipulada em computadores, porém seus usuários nem sempre encontram facilidade e praticidade em manter seus dados seguros e ilegíveis quando necessário, por meio de uma ferramenta simples e intuitiva.

A criptografia de dados consiste em métodos de codificação de informações, para garantir uma certa segurança e tornar assim informações ilegíveis a quem não deve ter acesso as mesmas. A criptografia é hoje uma ferramenta indispensável para a proteção de informações nos sistemas computacionais. Depois de muitos métodos e tipos de criptografia serem inventados, estudados e testados, dois desses métodos se mostram eficazes e competentes para a aplicação desenvolvida no decorrer deste trabalho: os métodos ECC (FLOSE, 2011) e AES ((NIST), 2001).

O objetivo geral desse trabalho é desenvolver um protótipo de uma ferramenta que possibilite ao usuário do sistema operacional Windows, manter seus arquivos em segurança. Especificamente, busca-se aplicar de maneira eficaz a criptografia como sistema de segurança principal da ferramenta, onde o usuário poderá utilizar as funcionalidades de cifragem e decifragem de seus arquivos pessoais, tendo como resultado uma aplicação que seja de uso simples e eficiente para o usuário.

Nesse sentido, o Capítulo 2 apresenta alguns fundamentos matemáticos necessários para o entendimento dos métodos apresentados. No Capítulo 3 é realizada uma descrição dos métodos criptográficos ECC e AES. O Capítulo 4, trata da implementação e teste da aplicação proposta nesse trabalho e o Capítulo 5 apresenta algumas considerações finais e sugestões de trabalhos futuros.

2. FUNDAMENTOS TEÓRICOS MATEMÁTICOS

Neste capítulo são disponibilizados os fundamentos teóricos matemáticos básicos para a compreensão dos métodos criptográficos ECC e AES. Esses fundamentos foram baseados em (MILIES, 2006), (SANTOS, 2007) e (STALLINGS, Tradução: VIEIRA, Revisão Técnica: BRESSAN, BARBOSA, & SUCCI, 2008).

2.1 TEORIA DOS NÚMEROS

2.1.1 DIVISIBILIDADE

Definição 2.1: Se a e b são inteiros, dizemos que a divide b , denotado por $a|b$, se existir um inteiro c tal que $b = ac$.

Proposição 2.1: Se a, b e c são inteiros, $a|b$ e $b|c$ então $a|c$.

Teorema 2.1: A divisão tem as seguintes propriedades:

- i. $n|n$.
- ii. $d|n \Rightarrow ad|an$.
- iii. $ad|an$ e $a \neq 0 \Rightarrow d|n$.
- iv. $1|n$.
- v. $n|0$.
- vi. $d|n$ e $n \neq 0 \Rightarrow |d| \leq |n|$.
- vii. $d|n$ e $n|d \Rightarrow |d| = |n|$
- viii. $d|n$ e $d \neq 0 \Rightarrow \left(\frac{n}{d}\right) |n$.

2.1.2 MÁXIMO DIVISOR COMUM

Definição 2.2: O máximo divisor comum de dois inteiros a e b (a e b diferentes de zero), denotado por (a, b) , é o maior inteiro que divide a e b

Teorema 2.2: Seja d o máximo divisor comum de a e b , então existem inteiros n_0 e m_0 tais que $d = n_0 * a + m_0 * b$.

Teorema 2.3: O máximo divisor comum d de a e b é o divisor positivo de a e b que é divisível por todo divisor comum.

Teorema 2.4: Se $a|bc$ e $\text{MDC}(a, b) = 1$, então $a|c$.

2.1.3 NÚMEROS PRIMOS

Definição 2.3: Um número inteiro n , ($n > 1$), possuindo somente dois divisores positivos n e 1 é chamado de primo.

Se não é primo dizemos que n é composto.

Proposição 2.2: Se $p|ab$, p primo, então $p|b$.

Teorema 2.5: (Teorema Fundamental da Aritmética). Todo inteiro maior do que 1 pode ser representado de maneira única (a menos da ordem) como um produto de fatores de primos.

2.1.4 CONGRUÊNCIA

Definição 2.4: Seja $m \neq 0$ um inteiro fixo. Dois inteiros a e b dizem-se congruentes módulos m se m divide a diferença $a - b$.

Proposição 2.3: Seja m um inteiro fixo. Dois inteiros a e b são congruentes módulo m se, e somente se, eles têm como resto o mesmo inteiro quando dividimos por m .

Proposição 2.4: Sejam $m > 0$ um inteiro fixo, e a, b, c, d inteiros arbitrários. Então, valem as seguintes propriedades:

- i. $a \equiv a \pmod{m}$.
- ii. Se $a \equiv b \pmod{m}$, então $b \equiv a \pmod{m}$.
- iii. Se $a \equiv b \pmod{m}$ e $b \equiv c \pmod{m}$, então $a \equiv c \pmod{m}$.
- iv. Se $a \equiv b \pmod{m}$ e $c \equiv d \pmod{m}$, então $a + c \equiv b + d \pmod{m}$.
- v. Se $a \equiv b \pmod{m}$, então $a + c \equiv b + c \pmod{m}$.
- vi. Se $a \equiv b \pmod{m}$ e $c \equiv d \pmod{m}$, então $ac \equiv bd \pmod{m}$.

- vii. Se $a \equiv b \pmod{m}$, então $a^n \equiv b^n \pmod{m}$, para todo inteiro positivo n .
- viii. Se $a + c \equiv b + c \pmod{m}$, então $a \equiv b \pmod{m}$.

Proposição 2.5: Seja m um inteiro fixo e sejam a, b e c inteiros arbitrários. Se $\text{mdc}(c, m) = 1$, então $ac \equiv bc \pmod{m}$ implica $a \equiv b \pmod{m}$.

2.1.5 INTEIROS MÓDULOS M

Definição 2.5: Seja a um inteiro. Chama-se classe de congruência de a módulo m o conjunto formado por todos os inteiros que são congruentes a a módulo m . Denotaremos esse conjunto por \bar{a} . Temos então, $\bar{a} = \{x \in \mathbb{Z} \mid x \equiv a \pmod{m}\}$. Como $x \equiv a \pmod{m}$ se, e somente se, x é da forma $x = a + tm$, para algum $t \in \mathbb{Z}$, também podemos escrever $\bar{a} = \{a + tm \mid t \in \mathbb{Z}\}$.

Proposição 2.6: Sejam a e b inteiros. Então $a \equiv b \pmod{m}$ se, e somente se, $\bar{a} = \bar{b}$.

2.2 GRUPOS, ANÉIS E CORPOS

2.2.1 GRUPOS

Definição 2.6: Um **grupo** G , às vezes indicado por $\{G, \cdot\}$, é um conjunto de elementos com uma operação binária, indicada por " \cdot ", que associa a cada par ordenado (a, b) de elementos em G um elemento $(a \cdot b)$ em G , de modo que os seguintes axiomas são obedecidos.

Proposição 2.7: São propriedades dos grupos:

- i. **Fechamento:** Se a e b pertencem a G , então $a \cdot b$ também está em G .
- ii. **Associativo:** $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ para todo a, b, c em G .
- iii. **Elemento identidade:** Existe um elemento e em G , de modo que $a \cdot e = e \cdot a$ para todo a em G .
- iv. **Elemento inverso:** Para cada a em G existe um elemento a' em G , de modo que $a \cdot a' = a' \cdot a = e$.
- v. **Comutativo:** $a \cdot b = b \cdot a$ para todo a, b em G .

2.2.2 ANÉIS

Definição 2.7: Um **anel** R , às vezes indicado por $\{R, +, \cdot\}$, é um conjunto de elementos com duas operações binárias, chamadas adição e multiplicação, de forma que para todo a, b, c em R , os seguintes axiomas são obedecidos:

- i. R é um grupo abeliano com relação à adição, ou seja, R satisfaz os axiomas de (i-v, proposição 2.7). Para o caso de um grupo aditivo, indicamos o elemento de identidade como zero e o inverso de a como $-a$.
- ii. **Fechamento sobre multiplicação:** Se a e b pertencem a R , então ab também está em R .
- iii. **Associatividade da multiplicação:** $a(bc) = (ab)c$ para todo a, b, c em R .
- iv. **Leis distributivas:** $a(b + c) = ab + ac$, para todo a, b, c em R .
 $(a + b)c = ac + ab$, para todo a, b, c em R .
- v. **Comutatividade da multiplicação:** $ab = ba$ para todo a, b em R .
- vi. **Identidade multiplicativa:** Existe um elemento 1 em R de modo que $a1 = 1a = a$ para todo a em R .
- vii. **Sem divisores zero:** Se a, b em R e $ab = 0$, $a = 0$ ou $b = 0$.

2.2.3 CORPOS

Definição 2.8: Um corpo F , às vezes indicado por $\{F, +, X\}$, é um conjunto de elementos com duas operações binárias chamadas de adição e multiplicação, de modo que para todo a, b, c em F , os seguintes axiomas são obedecidos:

- i. F é um domínio de integridade; ou seja, F satisfaz os axiomas (i-v, proposição 2.7; e ii-vi na definição 2.7).
- ii. **Inverso multiplicativo:** Para cada a em F , exceto 0 , existe um elemento a^{-1} em F tal que $aa^{-1} = (a^{-1})a = 1$.

2.3 CURVAS ELÍPTICAS

Todos os conceitos de curvas elípticas foram retirados de (MOLGORA, 2006).

Definição 2.9: As curvas elípticas podem ser definidas sobre um conjunto K como, por exemplo, o corpo dos números complexos. Elas são descritas por uma equação cúbica do tipo

$$ax^3 + bx^2y + cxy^2 + dy^3 + ex^2 + fxy + gy^2 + hx + iy + j = 0$$

Equação 1 - Equação Cúbica

onde a, b, \dots, j são elementos de um corpo K . Através de uma mudança apropriada de variáveis, uma curva elíptica geral sobre um corpo de característica diferente da equação 2 e 3, pode ser escrita na forma normal de Weierstrass:

$$y^2 = f(x) = x^3 + ax + b$$

Equação 2 - Forma normal de Weierstrass

Onde

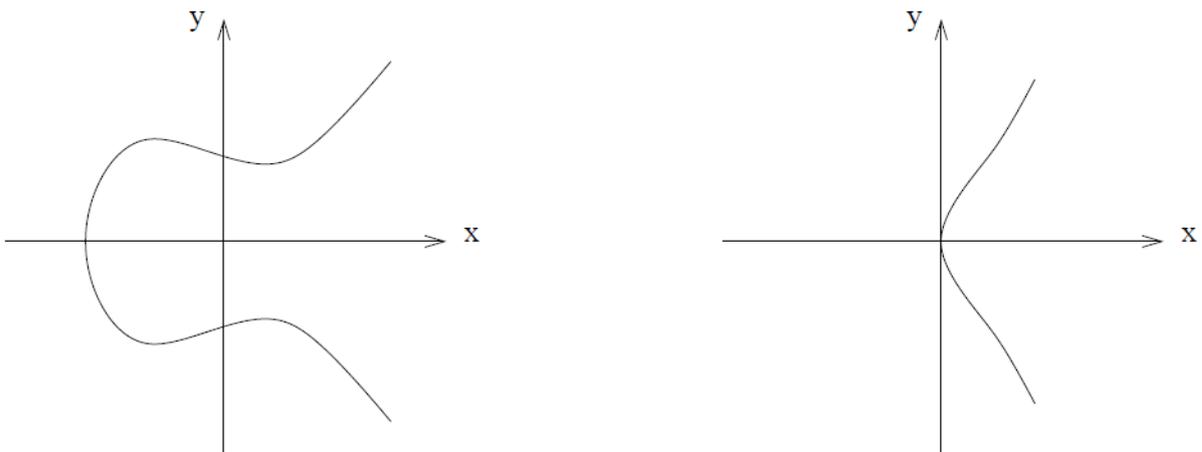
$$4a^3 + 27b^2 \neq 0$$

Equação 3 - Forma normal de Weierstrass

O que garante que f não tem raízes múltiplas.

Os gráficos de curvas elípticas variam de acordo com os parâmetros utilizados. A Figura 1 apresenta alguns exemplos de gráficos de curvas elípticas.

Figura 1 – Exemplos de Curvas elípticas



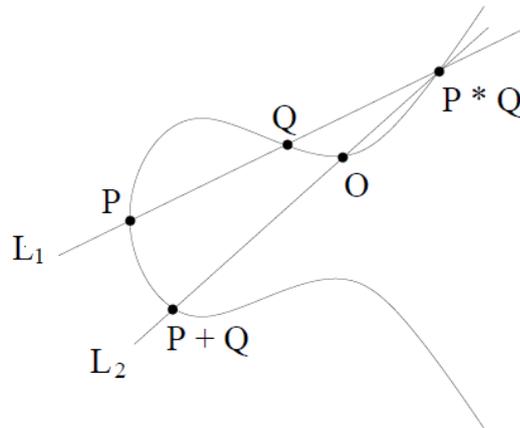
A teoria das curvas elípticas tem alguns fatos importantes e, um deles, é que se pode definir uma estrutura de grupo abeliano sobre essas curvas. Dado dois pontos P e Q em uma curva elíptica, é possível obter um terceiro ponto que será a adição de P e Q . O processo de adição de pontos em curvas elípticas é apresentado na próxima seção.

2.3.2 OPERAÇÃO DE ADIÇÃO

Seja O um ponto qualquer em uma curva elíptica C . Considere a operação $+$ que cada par (P, Q) de pontos de C associa o ponto $P + Q$ sobre C . Para isso, determina-se um terceiro ponto $P * Q$ traçando uma reta L_1 que passa por P e Q ; logo após uma reta L_2 é traçada passando

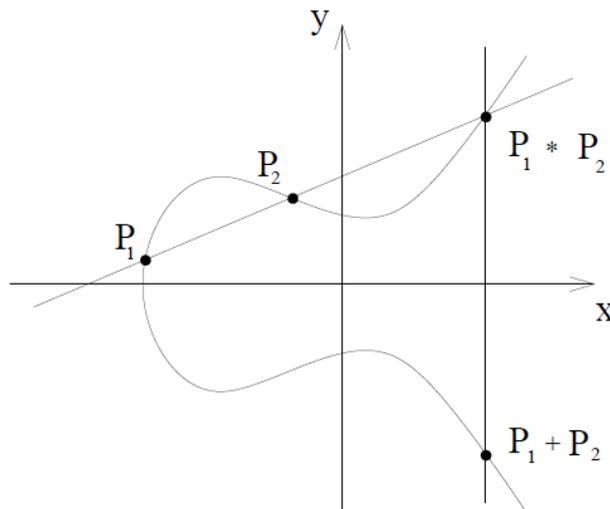
por $P * Q$ e por O , determinando o ponto que é $P + Q$. Esse processo pode ser visualizado na Figura 2.

Figura 2 - Adição na curva.



Supondo que o ponto O seja um ponto no infinito (o ponto de interseção entre as retas verticais), traça-se uma reta que passa por dois pontos P_1 e P_2 , encontrando o ponto $P_1 * P_2$. Em seguida, é traçada uma reta que passa por O e por $P_1 * P_2$, que é a reta vertical que passa por $P_1 * P_2$. O ponto simétrico de $P_1 * P_2$ será $P_1 + P_2$. Ver Figura 3.

Figura 3 - Adição de pontos na curva, sendo o elemento neutro como o ponto no infinito.



2.3.3 PROPRIEDADE DA OPERAÇÃO DE ADIÇÃO

- **Associativa:** Quaisquer que sejam os pontos P, Q e R em C tem-se que

$$(P + Q) + R = P + (Q + R).$$

- **Existência de elemento neutro:** O ponto $O \in C$ é tal que $P + O = O + P = P, \forall P \in C$, logo O é o elemento neutro da adição.
- **Existência de elemento simétrico:** Para cada ponto $P \in C$, existe o ponto $-P \in C$ tal que $P + (-P) = (-P) + P = O$.
- **Comutatividade:** Quaisquer que sejam os pontos $P, Q \in C$, tem-se $P + Q = Q + P$.

2.3.4 FÓRMULAS DE ADIÇÃO DE PONTOS EM CURVAS ELÍPTICAS

Adição de pontos P_1 e P_2 com $P_1 \neq P_2$ e $x_1 \neq x_2$: Dados dois pontos $P_1 = (x_1, y_1)$ e $P_2 = (x_2, y_2)$ pertencentes à curva $C(K)$ (definida sobre um corpo K) de equação $y^2 = x^3 + ax + b$, obtêm-se a soma $P_1 + P_2 = (x_3, y_3)$ através das seguintes equações:

$$\begin{cases} \lambda = \frac{y_2 - y_1}{x_2 - x_1} \\ x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

Equação 4 - Adição de pontos

De forma análoga pode-se reduzir as fórmulas para $P_1 + P_2$ no caso em que $P_1 = P_2$.

2.3.5 DUPLICAÇÃO DE PONTO

Seja $P_1 = (x_1, y_1)$. O resultado de $2P_1 = P_1 + P_1 = (x_3, y_3)$, é obtido através das fórmulas:

$$\begin{cases} \lambda = \frac{3x_1^2 - a}{2y_1} \\ x_3 = \lambda^2 - 2x_1 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

Equação 5 - Duplicação de Pontos

2.3.6 MULTIPLICAÇÃO DE PONTOS

A multiplicação de um ponto P de uma curva C por um inteiro $K > 2$ é considerada uma **operação escalar de um ponto**, podendo ser realizada utilizando as fórmulas de adição e duplicação de pontos:

$$kP = \underbrace{P + P + P \dots + P}_{k \text{ Vezes}}$$

Equação 6 – Multiplicação de pontos.

2.3.7 CURVAS SOBRE CORPOS FINITOS

Curvas elípticas tem diversas aplicações reais e, uma delas é a criptografia. Porém, na criptografia utilizam-se curvas sobre corpos finitos. Essa limitação é necessária para resolver problemas de arredondamento de valores e limites. Geralmente a curvas são limitas em corpos finitos \mathbb{Z}_p .

Definição 2.9: Seja p um número primo. O corpo finito \mathbb{Z}_p consiste do conjunto $\mathbb{Z}_p = \{0, 1, 2 \dots, p - 1\}$, onde as operações de adição e multiplicação sobre esse conjunto são definidas da seguinte maneira:

- **Adição:** Se $a, b \in \mathbb{Z}_p$, então $a + b = r$, onde $r, 0 \leq r \leq p - 1$, é o resto da divisão de $a + b$ (adição em \mathbb{Z}) por p . Esta operação é denominada **adição módulo p** .
- **Multiplicação:** Se $a, b \in \mathbb{Z}_p$, então $a * b = s$, onde $s, 0 \leq s \leq p - 1$, é o resto da divisão de $a * b$ por p . Esta operação é denominada **multiplicação módulo p** .

Uma curva sobre um corpo finito \mathbb{Z}_p indicada por $C(\mathbb{Z}_p)$ é definida pela equação:

$$y^2 = x^3 + ax + b \pmod{p}$$

Equação 7 - Curva sobre um corpo finito

onde, $a, b \in \mathbb{Z}_p$ e $4a^3 + 27b^2 \neq 0 \pmod{p}$, do mesmo modo com o ponto no infinito O . Essa curva consiste em todos os pontos (x, y) $x, y \in \mathbb{Z}_p$ do conjunto $C(\mathbb{Z}_p)$ satisfazendo a Equação 7.

Assim, tem-se $P_1 + P_2 = (x_3, y_3)$ sobre a curva limitada por \mathbb{Z}_p :

$$\begin{cases} \lambda = \frac{y_2 - y_1}{x_2 - x_1} \pmod{p} \\ x_3 = \lambda^2 - x_1 - x_2 \pmod{p} \\ y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p} \end{cases}$$

Equação 8 - Adição de Pontos.

Duplicação do ponto $P_1 = (x_1, y_1)$, onde $2P_1 = P_1 + P_1 = (x_3, y_3)$ sobre a curva limitada por \mathbb{Z}_p :

$$\begin{cases} \lambda = \frac{3x_1^2 - a}{2y_1} \pmod{p} \\ x_3 = \lambda^2 - 2x_1 \pmod{p} \\ y_3 = \lambda(x_1 - x_3) - y_1 \pmod{p} \end{cases}$$

Equação 9 - Duplicação de Pontos.

2.3.8 CURVAS ELÍPTICAS RECOMENDADAS

Na criptografia ECC, a curva elíptica é um elemento crucial para a segurança e eficiência do método. Com isso (BERNSTEIN & LANGE, 2014) efetuaram um estudo sobre curvas recomendadas para serem usadas em criptografia. Essas curvas seguem alguns padrões. Dentre eles destacam-se: ANSII X9.62 (1999), IEEE P1363 (2000) SEC2(2000). Essas curvas são classificadas no padrão SafeCurve (True e False). Com isso, temos alguns exemplos de curvas abaixo:

True:

- M-221.
- E-222.
- Curve1174.

False:

- Anomalous
- BN(2,254)
- NIST P-256

Neste trabalho optou-se por usar a curva M-221 , dada por $y^2 = x^3 + 117050x^2 + x \pmod p$ onde $p = 2^{221}-3$, que atende o padrão SafeCurve (ARANHA, BARRETO, PEREIRA, & RICARDINI, 2013).

3. CRIPTOGRAFIA

Utilizada a muito tempo a criptografia é um mecanismo de proteção de informações que hoje, após muitos estudos e desenvolvimentos, é um recurso amplamente empregado na computação. Existe diversos métodos criptográficos, cada um com seu sistema próprio de proteção.

3.1 MÉTODO CRIPTOGRÁFICO ECC

Segundo citação de (CASTELLANOS, 2004):

“Uma estrutura criptográfica baseada em curvas elípticas mapeia o puro texto (neologismo para plaintext) em pontos de uma curva elíptica pré-estabelecida. O algoritmo segue com a parametrização desses pontos com a chave do usuário obtendo assim um segundo conjunto de pontos representativo do cifrotexto (neologismo para ciphertext). O processo de decifração é dado pela conversão desses pontos (o cifrotexto) nos pontos originais com o auxílio das chaves, aplicando-se assim o mapeamento inverso”.

3.1.1 PROBLEMA DO LOGARITMO DISCRETO

Definição 3.1: Seja um grupo G e $y, \alpha \in G$ tal que y é potência de α . Dizemos que o logaritmo discreto de y na base α é o menor inteiro não negativo x tal que $\alpha^x = y$, denotado por $\log_{\alpha} y = x$.

Segundo (FLOSE, 2011), no problema do logaritmo discreto não é possível encontrar tal x em um tempo computacionalmente considerável, garantindo assim a segurança do método.

3.1.2 DIFFIE-HELLMAN

Já apresentados os conceitos básicos para criptografia, será apresentado o sistema criptográfico de chave pública. Esse método de criptografia será abordado com base em um algoritmo de troca de chaves, o algoritmo de Diffie-Hellman, citado por (FLOSE, 2011) da seguinte maneira:

“ Imagine que Alice deseja enviar uma mensagem pessoal e altamente secreta para Bob. Ela coloca sua carta secreta em uma caixa de ferro com um cadeado e envia para Bob. Este coloca um outro cadeado e envia para Alice novamente. Ao receber, Alice retira seu cadeado colocado inicialmente, e reenvia para Bob, que agora pode abri-la e ler a carta, pois a caixa está trancada apenas pelo seu cadeado. Observamos aqui que é possível realizar a troca de chaves através de um canal inseguro já que todos podem saber o transporte da caixa de ferro, porém ninguém, além de Alice e Bob, pode descobrir o que estava escrito na carta.”

3.1.2.1 DIFFIE-HELLMAN NAS CURVAS ELÍPTICAS

(FLOSE, 2011) Explica, em uma série de passos, como “Alice” envia uma mensagem cifrada e “Bob” a recebe, e lê. Imagina-se que B é conhecido publicamente e seja um ponto fixo em C , do qual a ordem é muito grande (N ou um divisor grande de N). Aplicam-se os seguintes passos para se obter a chave P :

- i. Alice e Bob primeiro escolhem publicamente um ponto $B \in C$.
- ii. Alice escolhe um inteiro a de uma ordem de grandeza q , aleatoriamente, que é aproximadamente a mesma ordem de N , que é secreta. Ela realiza o cálculo de $aB \in C$.
- iii. Bob seleciona um número aleatoriamente a qual ele o torna público $bB \in C$.
- iv. Alice tem o conhecimento de bB (que é público) e de seu segredo a , com isso ele pode calcular $P = abB \in C$. Bob conhece aB , e consegue calcular $P = abB \in C$.

De qualquer maneira, se uma terceira pessoa tem o conhecimento de apenas aB e bB , sem resolver o problema de logaritmo discreto, não há como calcular abB .

3.2 MÉTODO CRIPTOGRÁFICO AES

Em 1999, o NIST lançou uma nova versão de seu padrão criptográfico DES, esclarecendo que seu uso já deveria ser substituído pelo 3DES, exceto em sistemas legados. Porém, anteriormente, já haviam se diagnosticados alguns problemas com lentidão em software no padrão 3DES. Em 1997, o NIST pediu propostas para um novo padrão Advanced Encryption Standard que deveria ter segurança igual ou superior ao 3DES e eficiência melhorada. Partindo de dois criptógrafos belgas: o Dr. Joan Daemen e Dr. Vincent Rijmen, a proposta do padrão Rijndael foi aceita pelo NIST, e assim o AES se destinou a substituir o 3DES (STALLINGS, Tradução: VIEIRA, Revisão Técnica: BRESSAN, BARBOSA, & SUCCI, 2008).

3.2.1 PARÂMETROS DO AES

Na proposta do Rijndael (AES), foram definidos os possíveis tamanhos de chaves e dos blocos, independentemente, como 126, 192 e 256 bits. A seguir, apresenta-se uma tabela com os parâmetros do AES, baseados em (STALLINGS, Tradução: VIEIRA, Revisão Técnica: BRESSAN, BARBOSA, & SUCCI, 2008), ((NIST), 2001), (Joan Daemen, 2002) e (ROSA).

Tamanho da chave (word/bytes/bits)	4/16/128	6/24/192	8/32/256
Tamanho do bloco de texto claro (word/bytes/bits)	4/16/128	4/16/128	4/16/128
Número de rodadas	10	12	14
Tamanho da chave da rodada (word/bytes/bits)	4/16/128	4/16/128	4/16/128
Tamanho da chave expandida (word/bytes/bits)	44/176	52/208	60/240

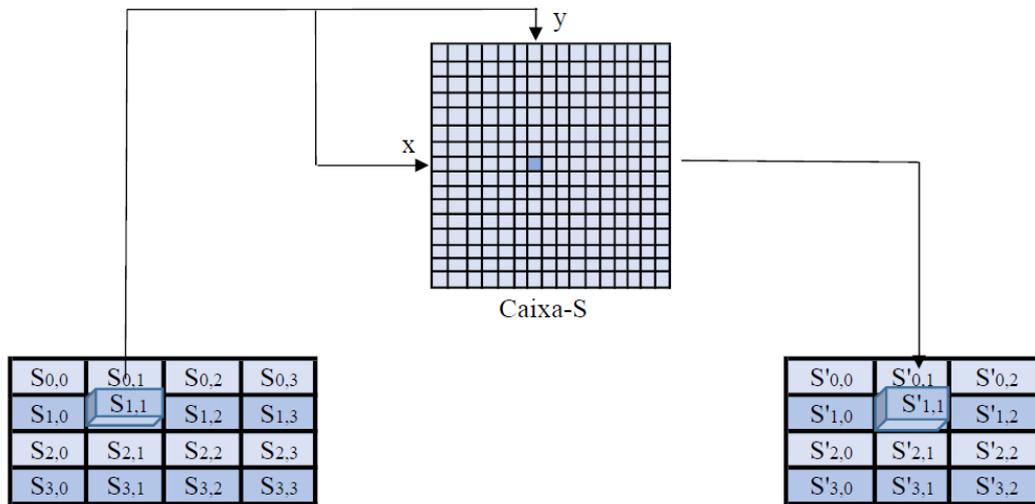
Tabela 1 - Parâmetros do AES

3.2.2 ESTRUTURA GERAL DO AES

Esse método criptográfico estabelece quatro estágios em seu procedimento. No AES, um estágio é responsável pela permutação, chamado de “ShiftRows”. Outros três estágios são responsáveis pelo processo de substituição, onde: “SubBytes” é o processo que utiliza uma caixa-S para realizar uma substituição byte a byte do bloco; “MixColumns” trata-se de uma combinação linear que utiliza aritmética sobre $GF(2^8)$; por fim, tem-se o processo “AddRoundKey”, que aplica um XOR bit a bit simples do bloco atual com uma parte da chave expandida.

3.2.3 TRANSFORMAÇÃO SUBBYTES

A substituição direta de bytes, consiste basicamente em uma tabela de busca. Para o AES é definida uma matriz de 16×16 de valores de bytes. Essa matriz é chamada de caixa-S, a qual contém uma permutação de 256 valores possíveis de 8 bits. Cada byte é mapeado para um novo byte de maneira que os 4 bits mais à esquerda formam um valor referente ao índice de linha ‘x’. De maneira análoga, os outros 4 bits formam um valor que é o índice de coluna ‘y’. Esses índices servem para determinar a posição na caixa-S, para então selecionar um valor de saída de 8 bits.

Figura 4 - Transformação direta SubBytes

3.2.4 TRANSFORMAÇÃO SHIFTRAWS

A primeira linha do **State** (Cópia do bloco de 128 bits de entrada) não é modificada. Já, a segunda linha, sofre um deslocamento circular de 1 byte à esquerda. Para a terceira linha, o procedimento é o mesmo, porém com um deslocamento de 2 bytes. A quarta linha, por sua vez, tem um deslocamento de 3 bytes à esquerda. Esse processo pode ser visualizado nas figuras 5 e 6.

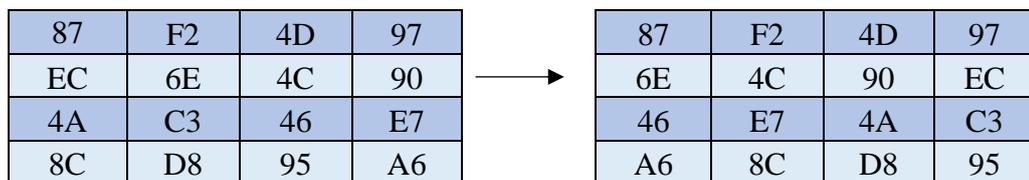
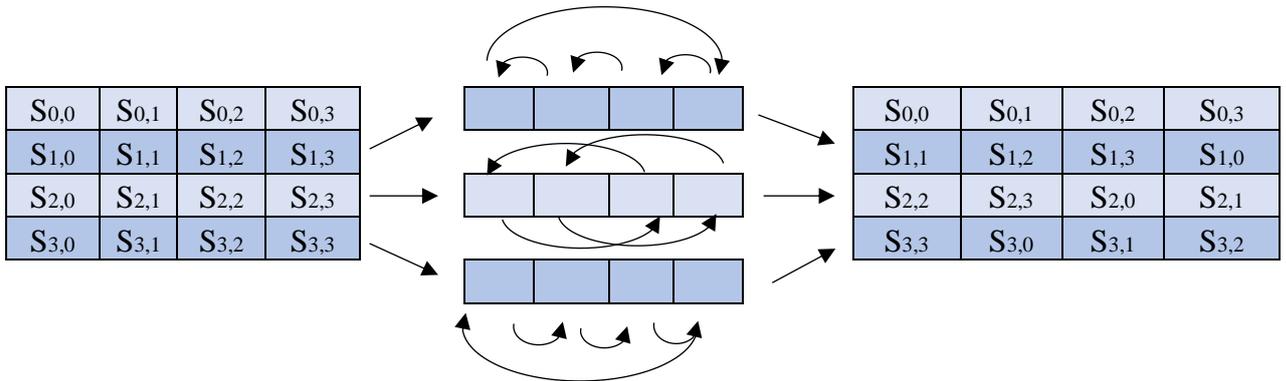
Figura 5 - Transformação direta ShiftRow

Figura 6 - Operações de linha ShiftRows



3.2.5 TRANSFORMAÇÃO MIXCOLUMNS

Essa transformação pode ser descrita como embaralhamento de colunas. O MixColumns opera sobre cada coluna de maneira individual, com o seguinte processo: cada byte de uma coluna é mapeado para um novo valor, o qual é uma combinação linear de todos os quatro bytes dessa coluna. A seguir, está definida uma transformação a partir da multiplicação matricial de **State**:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$

Equação 10 - Multiplicação de matrizes MixColumns

Esta transformação sobre uma única coluna j ($0 \leq j \leq 3$) de **State** pode ser expressa como:

$$S'_{0,j} = (2 \cdot S_{0,j}) \oplus (3 \cdot S_{1,j}) \oplus S_{2,j} \oplus S_{3,j}$$

$$S'_{1,j} = S_{0,j} \oplus (2 \cdot S_{1,j}) \oplus (3 \cdot S_{2,j}) \oplus S_{3,j}$$

$$S'_{2,j} = S_{0,j} \oplus S_{1,j} \oplus (2 \cdot S_{2,j}) \oplus (3 \cdot S_{3,j})$$

$$S'_{3,j} = (3 \cdot S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2 \cdot S_{3,j})$$

Equação 11 - Única coluna em MixColumns

A seguir, na Figura 7 um exemplo para a transformação MixColumns sobre o corpo $GF(2^8)$:

Figura 7 - Transformação MixColumns

87	F2	4D	97	→	47	40	A3	4C
6E	4C	90	EC		37	D4	70	9F
46	E7	4 ^a	C3		94	E4	3A	42
A6	8C	D8	95		ED	A5	A6	BC

3.2.6 TRANSFORMAÇÃO ADDROUNDKEY

A transformação AddRoundKey, consiste em aplicar um *XOR* bit a bit, entre os 128 bits de **State** e os 128 bits da chave da rodada. Essa transformação pode ser vista como uma operação de coluna entre os 4 bytes de uma coluna do **State** e uma word da chave da rodada. No exemplo apresentado na Figura 8, a primeira matriz é o **State**, e a segunda a chave.

Figura 8 - Transformação AddRoundKey

47	40	A3	4C	⊕	AC	19	28	57	=	EB	59	8B	1B
37	D4	70	9F		77	FA	D1	5C		40	2E	A1	C3
94	E4	3A	42		66	DC	29	00		F2	38	13	42
ED	A5	A6	BC		F3	21	41	6A		1E	84	E7	D2

4. IMPLEMENTAÇÃO

4.1 CRIPTOGRAFIA

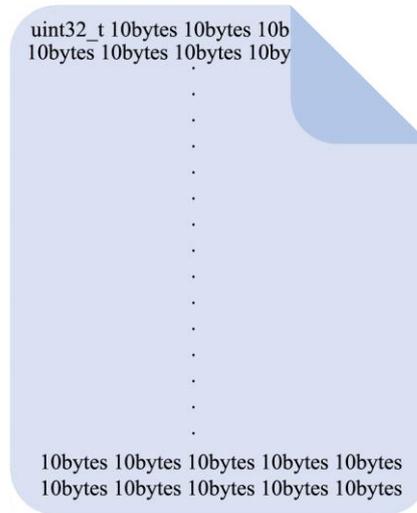
Para realizar a criptografia nos arquivos foi utilizado o método criptográfico AES, pelo fato de se tratar de uma criptografia simétrica onde o usuário é protagonista do cenário criptográfico, e pode então utilizar do método com sua chave única para a funcionalidade de criptografia local. Diferentemente do método ECC que se trata de uma criptografia assimétrica a qual auxilia no processo de troca de mensagens entre receptores de maneira segura, já esse método é indicado e cumpre com perfeição a função de compartilhar arquivos entre usuários.. Após o estudo do mesmo, esse método foi implementado em meio computacional utilizando a linguagem de programação C++, com auxílio da biblioteca Crypto++ (Crypto++® Library 5.6.3, 2016). Após o término da implementação, foi criada uma DLL contendo todos os códigos necessários para criptografar e descriptografar arquivos.

Diversos desafios foram enfrentados para obter o êxito em criptografar e descriptografar arquivos sem perder a legitimidade dos dados. Isso ocorreu, pois, dependendo da forma em que os dados eram coletados do arquivo, não era possível retornar ao estado normal. Para contornar essa dificuldade foi adotada uma divisão do arquivo em partes de 10 bytes, ou seja, eram coletados dados de 10 em 10 bytes do arquivo a ser criptografado.

Após a coleta dos dados do arquivo a ser cifrado e o conhecimento da chave e divisão de blocos que são necessários para realizar a criptografia, foi aplicada a criptografia AES nos blocos e iniciou-se a gravação do arquivo de saída.

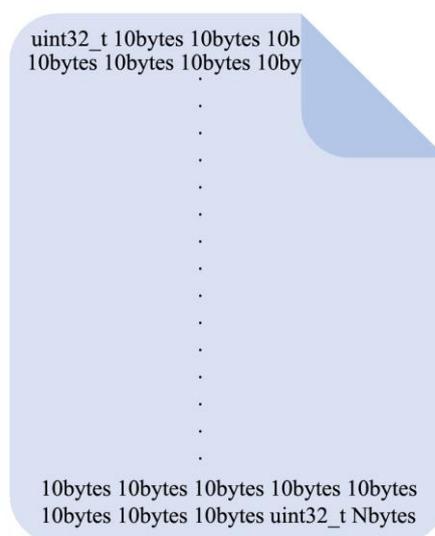
O arquivo de saída, que no caso é o arquivo criptografado, é composto por uma variável `uint32_t` em seu início, indicando a quantidade de blocos criptografados. Esses blocos de 10 bytes foram separados por espaço para não ocorrer a mistura de informações, como mostra a Figura 9.

Figura 9 - Divisão de arquivo sem resto.



Em alguns casos a divisão do arquivo não se torna exata, ou seja, ocorre de ter dados menores que 10 bytes tendo assim que serem tratados de forma diferente. Para resolver esse empasse, foi definida uma estrutura parecida com o arquivo anterior, porém é definido um `uint32_t` no final do arquivo indicando o tamanho dos últimos N bytes. Ver Figura 10.

Figura 10 - Divisão do arquivo



Para retornar o arquivo ao estado original, ou seja, descriptografar, são coletados os dados do arquivo criptografado usando como referência para a localização, as variáveis

uint32_t. Após os dados coletados e o conhecimento da chave e divisão de bloco, foi então aplicada a função para descriptografar o arquivo.

Foram implementados também os recursos necessários para utilizar a criptografia ECC, garantindo, assim, um canal seguro para troca de informações. Além disso, foi criada uma DLL contendo todos os códigos necessários do método criptográfico ECC.

4.2 BANCO DE DADOS

Para o armazenamento de informações dos usuários foi utilizado o banco de dados Firebird 3.0 (Firebird, 2016). Esse banco de dados possui duas tabelas, uma chamada **Usuários** que contém todos os dados dos usuários e uma chamada **Compartilhar** que auxilia no compartilhamento de arquivos entre usuários.

A tabela Usuários é composta pelos campos ID_USER (um contador de usuários), NOME (o nome completo dos usuários), DATA_NASC (data de nascimento), RECP (Palavra para recuperação), EMAIL (e-mail do usuário), SENHA (senha definida pelo usuário e criptografada utilizando a função bcrypt), CHAVE (chave de criptografia, gerada a partir da DLL e criptografada), DIV_BLOCO (divisão do blocos, gerada a partir da DLL e criptografada), P_CRIP (diretório de armazenamento de arquivos criptografados) e P_DESC(diretório de armazenamento de arquivos descriptografados).

TABELA USUARIOS	
ID_USER	INTEGER Not Null
NOME	VARCHAR(50) Not Null
DATA_NASC	VARCHAR(10) Not Null
RECP	VARCHAR(20) Not Null
EMAIL	VARCHAR(50) Not Null
SENHA	VARCHAR(200) Not Null
CHAVE	VARCHAR(200) Not Null
DIV_BLOCO	VARCHAR(200) Not Null
P_CRIP	VARCHAR(200) Not Null
P_DESC	VARCHAR(200) Not Null

Tabela 2 - Tabela Usuários

A tabela Compartilhar é composta pelo campo ID (contador de ações), ID_ENVIO (quem está compartilhando o arquivo), ID_RECB (quem está enviando o arquivo),

NOME_ARQ (nome do arquivo enviado) e ATIVO (que indica se a ação já foi ou não executada).

TABELA COMPARTILHAR	
ID	INTEGER Not Null
ID_ENVIO	INTEGER Not Null
ID_RECB	INTEGER Not Null
NOME_ARQ	VARCHAR(50) Not Null
ATIVO	INTEGER Not Null

Tabela 3 - Tabela Compartilhar

Toda manipulação do banco de dados foi realizada através da linguagem de programação Python e scripts SQL.

4.3 INTERFACE GRÁFICA

No protótipo desenvolvido buscou-se um sistema prático, onde o uso deste se dá de maneira intuitiva. Com o intuito de atingir usuários comuns de computador, ou seja, com pouco conhecimento sobre recursos computacionais, o protótipo criado apresenta uma interface simples e leve, onde os recursos disponíveis são apenas aqueles necessários para a cifragem e decifragem de arquivos pessoais.

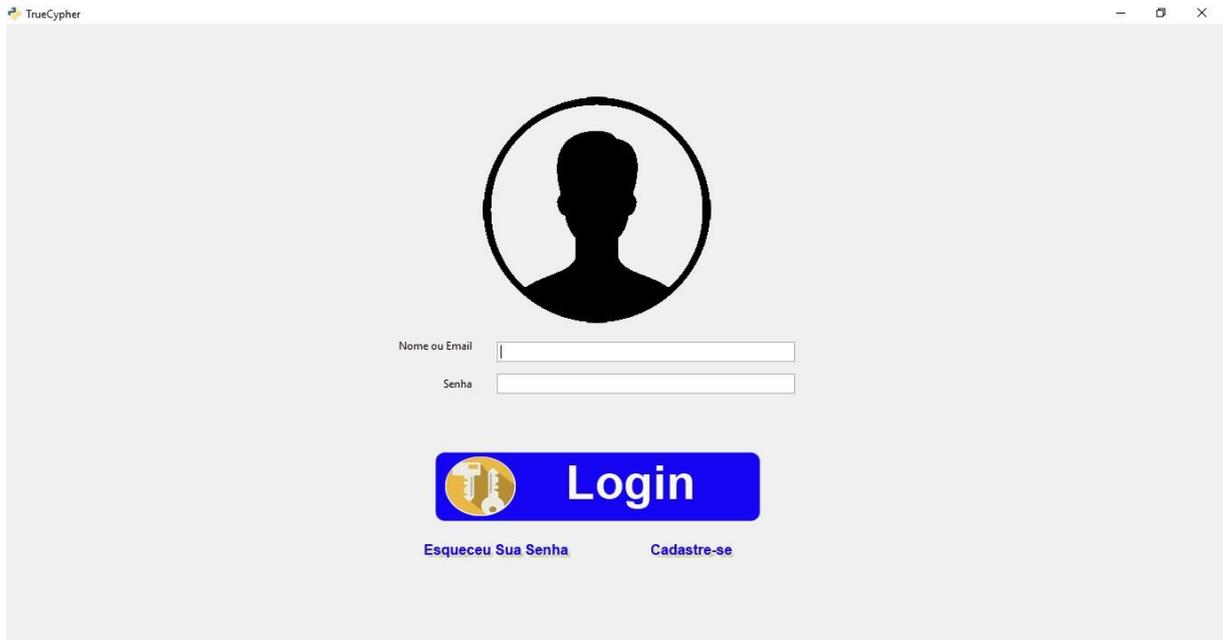
O protótipo dessa aplicação se encontra com a seguinte disposição da interface gráfica: o usuário inicia sua navegação no sistema com a tela de login, onde os casos de uso: cadastro de um novo usuário e recuperação de senha de um usuário existente estão presentes. Ao realizar o caso de uso de login no sistema, o usuário tem acesso a tela principal do sistema, onde uma janela acoplada, mostra a disposição de arquivos criptografados no sistema, e outra, mostra os arquivos descriptografados. As informações do usuário ativo são mostradas no canto superior direito, e as informações do arquivo corrente no sistema encontra-se no canto superior esquerdo.

O desenvolvimento da interface gráfica foi feito com a linguagem de programação Python 2.7 com o auxílio da biblioteca PyGTK (PyGTK: GTK+ for Python, 2016). Esta biblioteca é baseada na biblioteca GTK+, exclusiva para o ambiente Desktop GNOME e permite a criação de interfaces gráficas multiplataforma.

4.3.1 TELA DE LOGIN

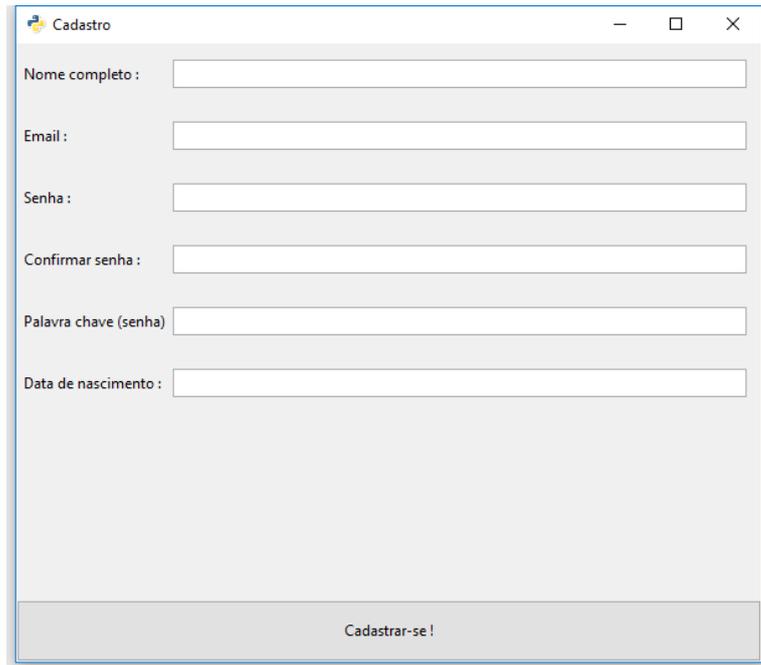
A primeira tela disponível no sistema é a tela de login (Figura 11).

Figura 11 - Tela Login



Essa tela é formada por alguns componentes, que são: três botões, “Login”, “Esqueceu Sua Senha” e “Cadastre-se”, com seus empregos e responsabilidades intuitivos a partir de seus nomes. Duas entradas de texto ficam responsáveis pela inserção de dados para o login, sendo “Nome ou Email” e “Senha”.

Ao clicar em “Cadastre-se” uma tela para o cadastro de um novo usuário (Figura 10) é ativada, com entradas de texto referentes as informações de registro e um botão para a conclusão do cadastro.

Figura 12 - Cadastro de Usuários

Nome completo :

Email :

Senha :

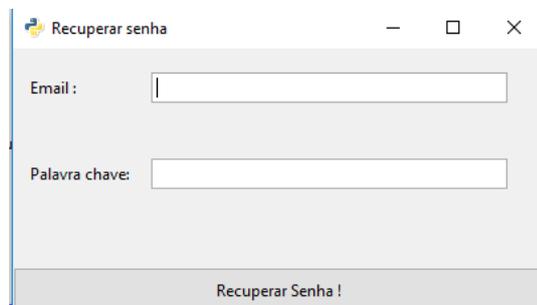
Confirmar senha :

Palavra chave (senha)

Data de nascimento :

Cadastrar-se !

Ao clicar em “Esqueceu Sua Senha” outra janela se abre (Figura 13), para a recuperação de senha.

Figura 13 - Recuperação de Senha

Recuperar senha

Email :

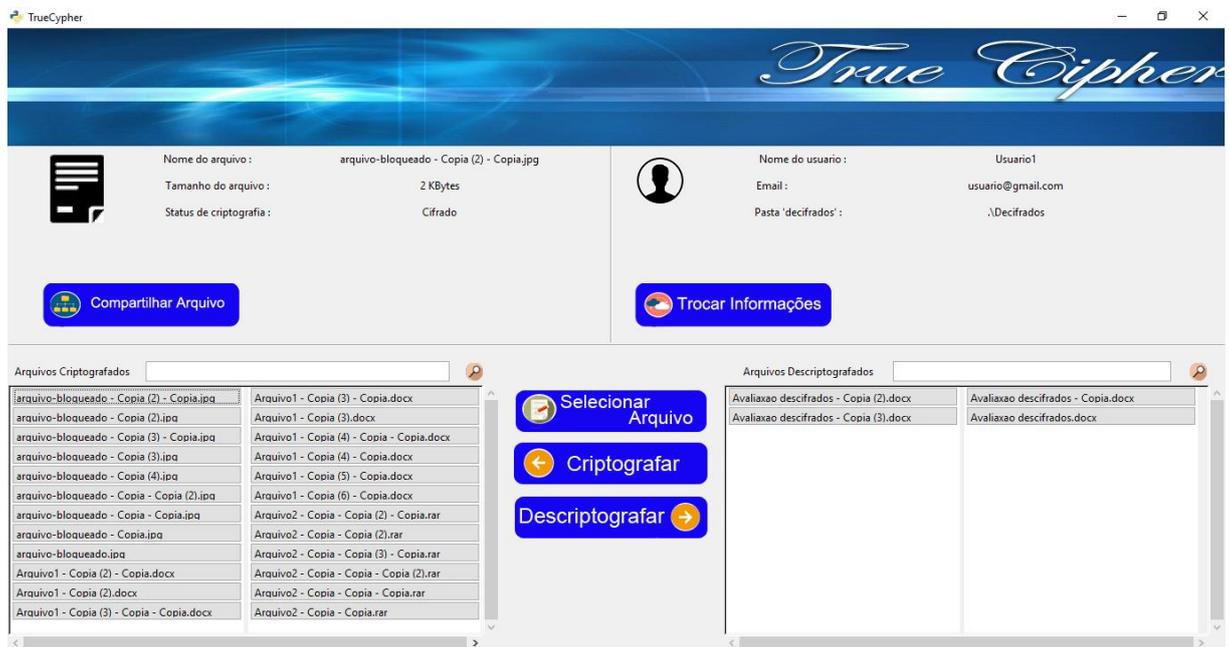
Palavra chave:

Recuperar Senha !

4.3.2 TELA PRINCIPAL

Ao iniciar a tela principal, caso tenha algum arquivo que foi compartilhado para ele, o mesmo aparece na tela identificando o arquivo compartilhado, quem compartilhou e se deseja excluir ou manter o arquivo. Na tela principal, o usuário do sistema tem acesso a todos os recursos oferecidos pelo mesmo. Entre as opções de uso do sistema temos os casos de uso: “Selecionar Arquivo”, “Criptografar”, “Descriptografar”, “Compartilhar Arquivo” e “Trocar informações”, que serão detalhados na Seção 4.3.3.

Figura 14 - Tela Principal



4.3.3 FUNCIONALIDADE

Como mostrado na Seção 4.3 a interface gráfica é composta por várias páginas e componentes. Como o objetivo desse trabalho era desenvolver um protótipo de uso simples, isso foi possível com poucos casos de uso na tela principal do sistema, onde “Criptografar” e “Descifrar” são o centro da aplicação.

No sistema, dois diretórios são protagonistas: o diretório de arquivos cifrados e o de arquivos decifrados. O primeiro é um diretório fixo, encontrado na raiz do protótipo onde são armazenados os arquivos que já passaram pelo processo de criptografia, tendo em vista que o arquivo original não se encontra mais no computador, o mesmo é apagado. O outro é um diretório de escolha pessoal, onde o usuário é responsável por selecionar.

4.3.4 SELECIONAR ARQUIVOS

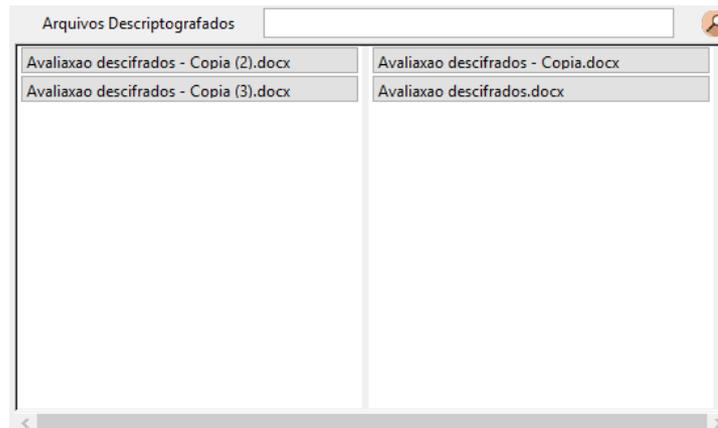
O protótipo desse sistema oferece 3 maneiras de seleção de arquivos, os diretórios padrão, e a seleção a partir da raiz do computador onde o sistema está instalado. Na Figura 15 é apresentado o diretório de arquivos já criptografados, que é acompanhado por um campo de busca para facilitar a seleção de arquivos.

Figura 15 - Selecionar Arquivo Criptografado

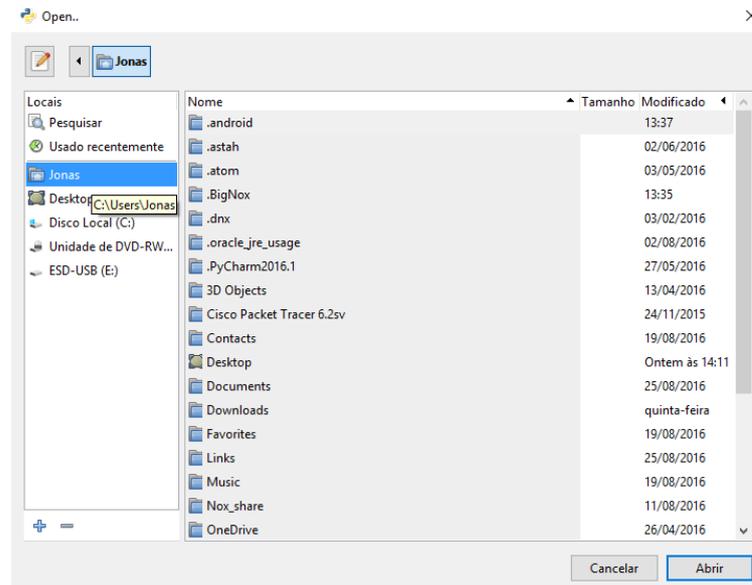


De maneira análoga à anterior, o diretório de arquivos descriptografados permite a segunda forma de seleção no sistema.

Figura 16 - Selecionar Arquivos Descriptografados



Por fim, o usuário pode fazer uma seleção de um arquivo a partir de outros diretórios do computador. Para isso ele deve clicar no botão “Selecionar Arquivo” e, a partir de um explorador, será possível realizar a seleção.

Figura 17 - Selecionar Arquivo

4.3.5 CRIPTOGRAFAR

Esse caso de uso é responsável por proteger um arquivo pertencente ao usuário por meio da criptografia. O usuário deseja cifrar um arquivo, então ele seleciona um arquivo por meio do explorador de arquivos no botão “Selecionar Arquivo” ou na pasta de arquivos decifrados. Ao selecionar um arquivo, as informações deste aparecem nas informações do arquivo. Para completar o caso de uso “Criptografar” o usuário deve então clicar no botão “Criptografar” e, então o sistema inicia a cifragem do arquivo selecionado e expõe uma janela para que o usuário aguarde o fim.

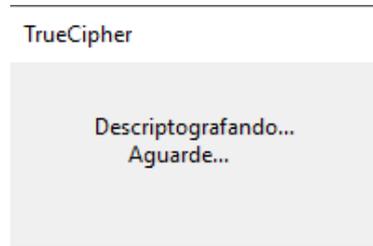
Figura 18 - Aguarde

4.3.6 DESCRIPTOGRAFAR

Quando o usuário pretende decifrar um arquivo, o procedimento de selecionar o arquivo desejado é o mesmo, porém a pasta local agora é a pasta de arquivos cifrados. Ao clicar em

“Descriptografar” o sistema inicia o processo para decifrar o arquivo. Uma janela é apresentada ao usuário.

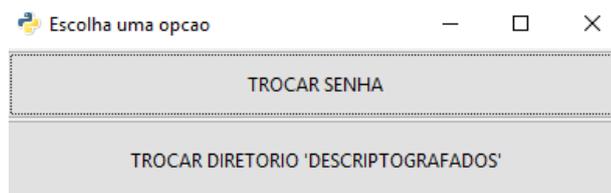
Figura 19 - Aguarde



4.3.7 TROCA DE INFORMAÇÕES

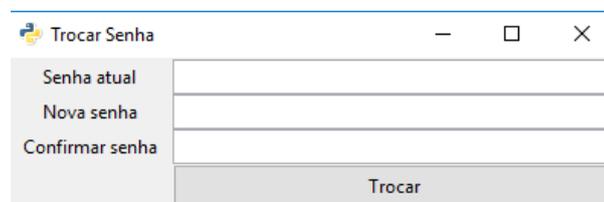
Na tela principal do protótipo o usuário pode ter acesso a pequenas configurações de sua conta, por meio do botão “Trocar informações”. O usuário receberá duas opções: trocar a senha de sua conta, ou trocar de diretório de arquivos descriptografados.

Figura 20 - Troca de Dados

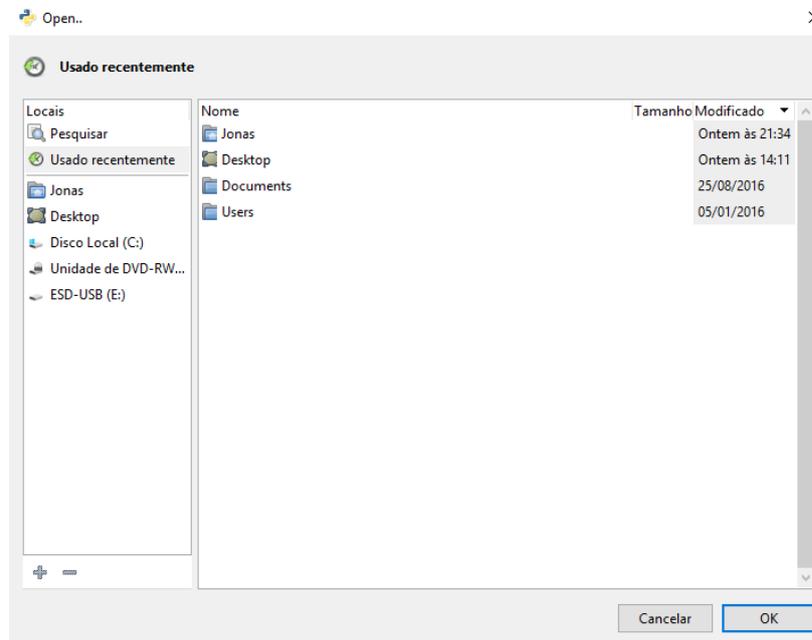


Ao selecionar “TROCAR SENHA” outra janela é criada para que os campos de alteração de senha sejam preenchidos.

Figura 21 - Trocar Senha

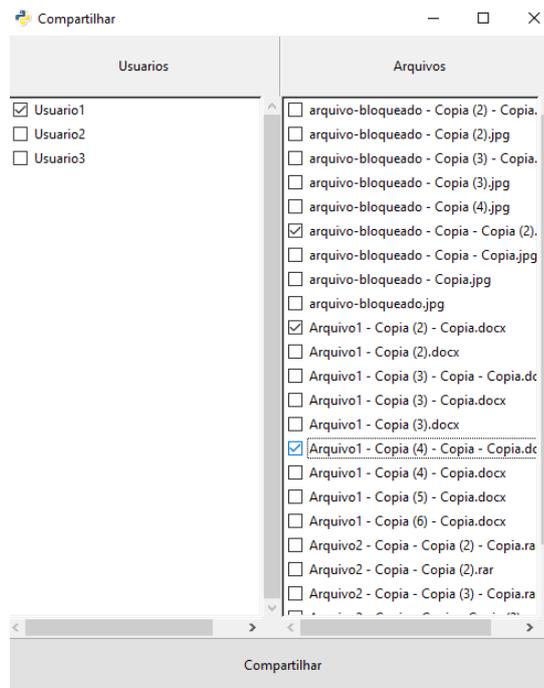


Escolhendo a segunda opção, um explorador de diretório semelhante ao de arquivos mostrado anteriormente é aberto, desta vez tendo a seleção somente de diretórios.

Figura 22 - Selecionar Arquivo

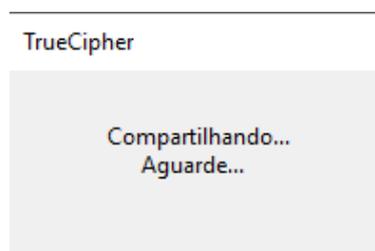
4.3.8 COMPARTILHAR ARQUIVO

Por fim, existe também a possibilidade de um usuário compartilhar arquivos com outro usuário local de maneira segura. Para acessar essa funcionalidade o usuário deve clicar no botão “Compartilhar Arquivo”, então uma janela com as opções de usuários disponíveis e arquivos possíveis para o envio aparece.

Figura 23 - Compartilhar

Nessa tela, o usuário pode fazer a seleção de usuários e arquivos já cifrados, como mostrado na Figura 23.

Quando o botão compartilhar é clicado, abre uma janela assim como mostra a Figura 24 informando que o compartilhamento iniciou, após o termino do compartilhamento o mesmo é fechado

Figura 24 - Compartilhar.

5. CONSIDERAÇÕES FINAIS

Sabemos que a quantidade de informações digitais cresce a cada momento e, armazenar e/ou manipular informações tem sido um trabalho árduo para todos que usufruem da tecnologia de informação. Com os sistemas se amplificando, a privacidade e segurança de arquivos pessoais ou confidenciais pode diminuir, sendo necessário o estudo de mecanismos para codificar as informações de forma eficiente.

A criptografia utiliza diversos recursos, como a matemática e lógica comumente, para tornar uma informação ilegível. Buscando essa segurança, muitos métodos criptográficos já foram desenvolvidos visando maximizar a proteção e integridade da informação. Mesmo assim, invasões de privacidade e a falta de segurança no meio computacional afetam diariamente grandes companhias e usuários convencionais de sistemas de informação. No entanto, seja para evitar perdas ou encorajar a permeação da tecnologia da informação em nossa sociedade, a criptografia ainda é pouco utilizada quando comparado ao uso de outras funcionalidades na tecnologia.

É de fundamental importância o crescimento do acervo de ferramentas que possibilitem a proteção e privacidade do usuário de tecnologia. Nesse sentido, esse trabalho possibilitou o desenvolvimento de um protótipo de uma ferramenta que permite ao usuário do sistema operacional Windows, manter seus arquivos em segurança. Para isso, foram realizados estudos sobre os métodos criptográficos AES e ECC, os quais foram aplicados no desenvolvimento dessa ferramenta permitindo sua segurança. Além disso, foi desenvolvida uma interface gráfica que possibilitou a obtenção de uma aplicação de uso simples e eficiente para o usuário.

6. FUTURAS IMPLEMENTAÇÕES

Desde o início desse trabalho, buscou-se desenvolver uma ferramenta que usasse da criptografia para oferecer ao usuário um ambiente seguro, estável e prático, que permitisse que seus arquivos pessoais se tornassem ilegíveis por pessoas não autorizadas. O protótipo desenvolvido oferece funcionalidades básicas, que podem ser aprimoradas.

Sugerem-se como futuras modificações e implementações neste estudo, agregar fatores como confiança, segurança, comodidade e disponibilidade de recursos avançados. Mantendo os ideais do projeto, também pode-se aplicar uma programação paralela, visando otimização dos recursos oferecidos, com um desempenho maior do processador. Além disso, e de grande importância a otimização da interface gráfica, para uma melhor usabilidade, mantendo-a sobre responsabilidade de um núcleo do processador exclusivo, destinando a capacidade da máquina para os processos de cifragem e decifragem de dados.

No protótipo desenvolvido até o momento, existe uma funcionalidade da ferramenta nomeada de “compartilhar arquivos”, que permite o compartilhamento de arquivos entre usuários locais no sistema. Em futuras implementações a mesma funcionalidade poderia ter a possibilidade de expandir e efetuar o compartilhamento entre máquinas fisicamente distantes, enfatizando o uso de um método criptográfico de chave pública, podendo ser o método ECC. Para tal implementação necessita-se que o gerenciamento dos usuários do sistema seja feito através de um servidor central.

REFERÊNCIAS BIBLIOGRÁFICAS

- (NIST), N. I. (2001). *ADVANCED ENCRYPTION STANDARD (AES)*. Federal Information Processing Standards Publication 197 . Fonte:
<http://www.cisco.com/c/dam/en/us/products/collateral/security/anyconnect-secure-mobility-client/fips.pdf>
- ARANHA, D. F., BARRETO, P. S., PEREIRA, G. C., & RICARDINI, J. (4 de Novembro de 2013). *A note on high-security general-purpose elliptic curves*. Fonte: Cryptology ePrint archive: <http://eprint.iacr.org/2013/647.pdf>
- BERNSTEIN, D. J., & LANGE, T. (18 de Janeiro de 2014). *SafeCurves: choosing safe curves for elliptic-curve cryptography*. Acesso em 13 de Junho de 2016, disponível em <http://safecurves.cr.yt.to>
- CASTELLANOS, A. S. (2004). *Criptografia usando curvas hiperelípticas*. Dissertação de Mestrado - Instituto de Matemática, Estatística e Computação Científica. Universidade Estadual de Campinas .
- Crypto++® Library 5.6.3*. (Acessado 10 de Setembro de 2016). Fonte: Crypto++® Library 5.6.3: <https://www.cryptopp.com/>
- Firebird*. (Acessado 10 de Setembro de 2016). Fonte: Firebird: <http://firebirdsql.org/>
- FLOSE, V. B. (2011). *Criptografia e curvas elípticas*. Dissertação de Mestrado - Universidade Estadual Paulista, Instituto de Geociências e Ciências Exatas.
- gmplib.org*. (2016). *The GNU Multiple Precision Arithmetic Library*. Fonte: <https://gmplib.org/>
- Joan Daemen, S. B. (2002). *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag.
- MILIES, F. C. (2006). *Números: Uma Introdução a matemática*. São Paulo: Editora da Universidade de São Paulo.
- MOLGORA, A. B. (2006). *Uma implementação do Método das Curvas Elípticas para Fatoração de números Inteiros*. Dissertação de Mestrado, Departamento de Computação e Estatística, Universidade Federal de Mato Grosso do Sul.
- PyGTK: GTK+ for Python*. (Acessado 10 de Setembro de 2016). Fonte: PyGTK: GTK+ for Python: <http://www.pygtk.org/>
- RIVEST, R., SHAMIR, A., & ADLEMAN, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, Vol. 21 (2).
- ROSA, R. A. (s.d.). *Análise do Algoritmo vencedor do AES: O Rijndael*. Instituto Tecnológico de Aeronáutica(ITA).
- SANGALLI, L. A. (2012). *Criptossistemas baseados em curvas elípticas e seus desafios*. Fonte: Departamento de Engenharia de Computação e Automação Industrial. Universidade Estadual de Campinas (UNICAMP):
<http://www.dca.fee.unicamp.br/portugues/pesquisa/seminarios/2012/artigos/217.pdf>

SANTOS, J. P. (2007). *Introdução a teoria dos números*. Rio de Janeiro : IMPA.

STALLINGS, W., Tradução: VIEIRA, D., Revisão Técnica: BRESSAN, G., BARBOSA, Á., & SUCCI, M. (2008). *Criptografia e Segurança de Redes*. São Paulo: Pearson Prentice Hall.

APÊNDICE A

CRIPTOGRAFA AES

```
int criptografa_AES(char * Entrada, char * Saida, char * iv_aux, char *chave_aux){
    byte * iv = (byte*)iv_aux;
    byte * chave = (byte*)chave_aux;
    char c='\0';
    char buffer[10];
    uint8_t buffer2[20];
    uint32_t tamanho_arq=0,i=0, divisao = 0 ;
    FILE *mensagem; //Arquivo Descriptografado
    FILE *crypto_mensagem; //Arquivo Criptografado

    if ((mensagem=fopen(Entrada, "rb"))==NULL){
        printf("ERRO NA ABERTURA DO ARQUIVO MENSAGEM\n");
        return -1;
    }else{
        if ((crypto_mensagem=fopen(Saida, "w+b"))==NULL){
            printf("ERRO NA ABERTURA DO ARQUIVO CRYPTOTOMSG\n");
            return -2;
        }else{
            fseek(mensagem, 0, SEEK_END);
            tamanho_arq = ftell(mensagem);
            fseek(mensagem, 0, SEEK_SET);
            divisao=tamanho_arq/10;
            if((tamanho_arq%10)!=0)
                divisao--;
            fwrite(&divisao, sizeof(uint32_t), 1, crypto_mensagem);
            c=' ';
            for(i=0;i<divisao;i++){
                memset(&buffer,0,sizeof(buffer));
                if(fread(&buffer,sizeof(char),10,mensagem)>0){

                    CFB_Mode<AES>::Encryption cfbEncryption(chave, AES::MAX_KEYLENGTH, iv);
                    cfbEncryption.ProcessData((byte*)buffer, (byte*)buffer, 10);

                    fwrite(&c,sizeof(char),1,crypto_mensagem);
                    fwrite(buffer,sizeof(char),10,crypto_mensagem);
                }
            }
            if((tamanho_arq%10)!=0){
                divisao=divisao*10;
                divisao=tamanho_arq-divisao;

                fwrite(&c,sizeof(char),1,crypto_mensagem);
                fwrite(&divisao,sizeof(uint32_t),1,crypto_mensagem);

                memset(&buffer2,0,sizeof(buffer2));
                if(fread(&buffer2,sizeof(char),divisao,mensagem)>0){

                    CFB_Mode<AES>::Encryption cfbEncryption(chave, AES::MAX_KEYLENGTH, iv);
                    cfbEncryption.ProcessData((byte*)buffer2, (byte*)buffer2, divisao);

                    fwrite(&c,sizeof(char),1,crypto_mensagem);
                    fwrite(buffer2,sizeof(char),divisao,crypto_mensagem);
                }
            }
            fclose(crypto_mensagem);
            fclose(mensagem);
        }
    }
    return 0;
}
```

APÊNDICE B

DESCRIPTOGRAFA AES

```

int descriptografa_AES(char* Entrada, char * Saida, char * iv_aux, char *chave_aux){
    byte * iv = (byte*)iv_aux;
    byte * chave = (byte*)chave_aux;
    char c='\0';
    char buffer[10];
    char buffer2[20];
    uint32_t divisao=0,i;

    FILE *descript_mensagem; //Arquivo Descriptografado
    FILE *crypto_mensagem; //Arquivo Criptografado

    if ((crypto_mensagem=fopen(Entrada, "rb"))==NULL){
        printf("ERRO NA ABERTURA DO ARQUIVO MENSAGEM\n");
        return -1;
    }else{
        if ((descript_mensagem=fopen(Saida, "w+b"))==NULL){
            printf("ERRO NA ABERTURA DO ARQUIVO CRYPTOTOMSG\n");
            return -2;
        }else{

            fread(&divisao,sizeof(uint32_t),1,crypto_mensagem);

            for(i=0;i<divisao;i++){
                fread(&c,sizeof(char),1,crypto_mensagem);
                if(!feof(crypto_mensagem)){
                    memset(&buffer,0,sizeof(buffer));
                    if(fread(&buffer,sizeof(char),10,crypto_mensagem)>0){

                        CFB_Mode<AES>::Decryption cfbDecryption(chave, AES::MAX_KEYLENGTH, iv);
                        cfbDecryption.ProcessData((byte*)buffer, (byte*)buffer, 10);

                        fwrite(buffer,sizeof(char),10,descript_mensagem);
                    }
                }
            }

            if(!feof(crypto_mensagem)){
                fread(&c,sizeof(char),1,crypto_mensagem);
                fread(&divisao,sizeof(uint32_t),divisao,crypto_mensagem);
                fread(&c,sizeof(char),1,crypto_mensagem);

                memset(&buffer2,0,sizeof(buffer2));
                if(fread(&buffer2,sizeof(char),divisao,crypto_mensagem)>0){

                    CFB_Mode<AES>::Decryption cfbDecryption(chave, AES::MAX_KEYLENGTH, iv);
                    cfbDecryption.ProcessData((byte*)buffer2, (byte*)buffer2, divisao);

                    fwrite(buffer2,sizeof(char),divisao,descript_mensagem);
                }
            }

            fclose(crypto_mensagem);
            fclose(descript_mensagem);
        }
    }
    return 0;
}

```

APÊNDICE C

COMPARTILHAR

```

int Compartilhar(char* Entrada, char * Saida, char * iv_aux, char *chave_aux, char *
iv_aux_saida, char *chave_aux_saida){

    byte * iv = (byte *) iv_aux;
    byte * chave = (byte *) chave_aux;
    byte * iv_saida = (byte *) iv_aux_saida;
    byte * chave_saida = (byte *) chave_aux_saida;
    char c='\0';
    char buffer[10];
    char buffer2[20];
    uint32_t divisao=0,i;

    FILE *descript_mensagem; //Arquivo Descriptografado
    FILE *crypto_mensagem; //Arquivo Criptografado

    if ((crypto_mensagem=fopen(Entrada, "rb"))==NULL){
        return -1;
    }else{
        if ((descript_mensagem=fopen(Saida, "w+b"))==NULL){
            return -2;
        }else{

            fread(&divisao,sizeof(uint32_t),1,crypto_mensagem);
            fwrite(&divisao,sizeof(uint32_t),1,descript_mensagem);
            for(i=0;i<divisao;i++){
                fread(&c,sizeof(char),1,crypto_mensagem);
                fwrite(&c,sizeof(char),1,descript_mensagem);
                if(!feof(crypto_mensagem)){
                    memset(&buffer,0,sizeof(buffer));
                    if(fread(&buffer,sizeof(char),10,crypto_mensagem)>0){
                        CFB_Mode<AES>::Decryption cfbDecryption(chave, AES::MAX_KEYLENGTH, iv);
                        cfbDecryption.ProcessData((byte*)buffer, (byte*)buffer, 10);

                        CFB_Mode<AES>::Encryption cfbEncryption(chave_saida, AES::MAX_KEYLENGTH, iv_saida);
                        cfbEncryption.ProcessData((byte*)buffer, (byte*)buffer, 10);

                        fwrite(buffer,sizeof(char),10,descript_mensagem);
                    }
                }
            }

            if(!feof(crypto_mensagem)){
                fread(&c,sizeof(char),1,crypto_mensagem);
                fwrite(&c,sizeof(char),1,descript_mensagem);
                fread(&divisao,sizeof(uint32_t),divisao,crypto_mensagem);
                fwrite(&divisao,sizeof(uint32_t),divisao,descript_mensagem);
                fread(&c,sizeof(char),1,crypto_mensagem);
                fwrite(&c,sizeof(char),1,descript_mensagem);

                memset(&buffer2,0,sizeof(buffer2));
                if(fread(&buffer2,sizeof(char),divisao,crypto_mensagem)>0){

                    CFB_Mode<AES>::Decryption cfbDecryption(chave, AES::MAX_KEYLENGTH, iv);
                    cfbDecryption.ProcessData((byte*)buffer2, (byte*)buffer2, divisao);

                    CFB_Mode<AES>::Encryption cfbEncryption(chave_saida, AES::MAX_KEYLENGTH, iv_saida);
                    cfbEncryption.ProcessData((byte*)buffer2, (byte*)buffer2, 10);

                    fwrite(buffer2,sizeof(char),divisao,descript_mensagem);
                }
            }

            fclose(crypto_mensagem);
            fclose(descript_mensagem);
        }
    }
    return 0;
}

```

APÊNDICE D

TELA – LOGIN

```

*** Exemplo de criação de janela ***
self.window2 = gtk.Window(gtk.WINDOW_TOPLEVEL)
self.window2.set_size_request(1000, 500)
self.window2.maximize()
self.window2.set_title('TrueCypher')
self.window2.connect("delete_event", self.delete_event)
self.window2.connect("destroy", self.destroy)
self.window2.set_border_width(1)

*** Exemplo de criação de container TABELA ***
fixed = gtk.Table(120, 120, True)
self.window2.add(fixed)
fixed.show()

*** Exemplo de criação entrada de texto ***
entry = gtk.Entry(max=0)
fixed.attach(entry, 50, 80, 62, 67)
entry.show()
entry2 = gtk.Entry(max=0)
fixed.attach(entry2, 50, 80, 68, 73)
entry2.set_visibility(False)
entry2.set_invisible_char("***")
entry2.show()

*** Exemplo de criação de botão ***
button = gtk.Button()
button.props.relief = gtk.RELIEF_NONE
button.connect("clicked", self.controle_log, self.window2, entry, entry2)
button.set_usize(380, 90)
image = gtk.image_new_from_file('Imagens\Login.jpg')
button.add(image)
image.show()
fixed.attach(button, 40, 80, 80, 100)
button.show()
button = gtk.Button()
self.image = gtk.image_new_from_file('Imagens\E esqueceu.jpg')
button.add(self.image)
button.props.relief = gtk.RELIEF_NONE
button.connect("clicked", add1.controle_esquecisenha)
button.set_usize(100, 20)
fixed.attach(button, 40, 60, 100, 104)
button.show()

button = gtk.Button()
self.image = gtk.image_new_from_file('Imagens\Cadastre-se.jpg')
button.add(self.image)
button.props.relief = gtk.RELIEF_NONE
button.connect("clicked", add1.controle_cadastro)
button.set_usize(100, 20)
fixed.attach(button, 60, 80, 100, 104)
button.show()

*** criação de label ***
label = gtk.Label("Nome ou Email")
label.set_alignment(0,0)
fixed.attach(label, 40, 50, 62, 67)
label.set_alignment(0,0)
label = gtk.Label("Senha")
fixed.attach(label, 42, 50, 68, 73)

*** Indexação de imagem ***
image = gtk.image_new_from_file('Imagens\peril.png')
image.show()
fixed.attach(image, 40, 80, 15, 60)
self.window2.show_all()

```

APÊNDICE E

TELA – SCROLL

```
*** Criação da tela ***

self.boxH = gtk.HBox(False, 5)
self.scrolled_window2.add_with_viewport(self.boxH)
self.list = gtk.List()
self.boxH.pack_start(self.list, False, False, 0)

*** Indexação da lista ***
for i in range(len(onlyfiles)):
    if (i == len(onlyfiles)/2):
        self.list = gtk.List()
        self.boxH.pack_start(self.list, False, False, 0)

        nome = onlyfiles[i]
        button = gtk.Button(nome)
        button.set_size_request(256, 20)
        button.set_alignment(0, 0)
        list_item = gtk.ListItem()
        list_item.add(button)
        self.list.add(list_item)
        string = button.get_label()
        button.set_tooltip_text(string)
        button.connect("clicked", self.radio_response_aux, string)

self.boxH.show_all()
```

APÊNDICE F

TELA – FILECHOOSER/REPO. DIALOG

```
dialog = gtk.FileChooserDialog("Open..",  
                               None,  
                               gtk.FILE_CHOOSER_ACTION_OPEN,  
                               (gtk.STOCK_CANCEL, gtk.RESPONSE_CANCEL,  
                                gtk.STOCK_OPEN, gtk.RESPONSE_OK))  
  
dialog.set_default_response(gtk.RESPONSE_OK)  
response = dialog.run()
```

APÊNDICE G

CRIPTOGRAFANDO ECC

```
void Criptografando(Chave Alice,char c1,char c2, int *x1, int *x2, Curva Equa_Cryp){
    mpz_t alice_x_aux2, alice_y_aux2;
    mpz_init(alice_x_aux2);
    mpz_init(alice_y_aux2);

    *x1= (int)c1;
    //Calculo Criptografia por x1
    mpz_mul_ui(alice_x_aux2,Alice.x,*x1);
    mpz_mod(alice_x_aux2,alice_x_aux2,Equa_Cryp.field);
    *x1=mpz_get_ui(alice_x_aux2);

    *x2= (int)c2;
    //Calculo Criptografia por x2
    mpz_mul_ui(alice_y_aux2,Alice.y,*x2);
    mpz_mod(alice_y_aux2,alice_y_aux2,Equa_Cryp.field);
    *x2=mpz_get_ui(alice_y_aux2);
}
```

APÊNDICE H

DESCRIPTOGRAFA ECC

```
void Descriptografando(Chave Alice,char *c1,char *c2, int x1, int x2, Curva Equa_Cryp){
    mpz_t alice_x_aux, alice_y_aux;
    mpz_t alice_x_aux2, alice_y_aux2;
    mpz_init(alice_x_aux);
    mpz_init(alice_y_aux);
    mpz_init(alice_x_aux2);
    mpz_init(alice_y_aux2);

    mpz_invert(alice_x_aux,Alice.x,Equa_Cryp.field);
    mpz_invert(alice_y_aux,Alice.y, Equa_Cryp.field);

    //Calculo Descriptografando por x1
    mpz_mul_ui(alice_x_aux2,alice_x_aux,x1);
    mpz_mod(alice_x_aux2,alice_x_aux2,Equa_Cryp.field);
    x1=mpz_get_ui(alice_x_aux2);
    *c1=(char)x1;

    //Calculo Descriptografando por x2
    mpz_mul_ui(alice_y_aux2,alice_y_aux,x2);
    mpz_mod(alice_y_aux2,alice_y_aux2,Equa_Cryp.field);
    x2=mpz_get_ui(alice_y_aux2);
    *c2=(char)x2;
}
```