REVISTA DE COMPUTAÇÃO E SISTEMAS (RCS)
ISSN XXXX-XXXX
VOLUME 1 – NÚMERO 1 – ABRIL/2016
http://www.revistacomputacaoesistemas.net

Fortalecimento do Esquema de Criptografia por Curvas Elípticas via Rotações Tridimensionais

Daniel Lackeski Suigh Carlos, Luciano Silva
Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie
Caixa Postal 930 – 01.302-907 – São Paulo – SP – Brasil
danlscarlos@gmail.com,luciano.silva@mackenzie.br

Abstract. *The elliptic curve cryptography is a feasible alternative to the traditional RSA method. It uses the difficulty of solving the discrete logarithm problem in an algebraic field defined by an elliptic curve. Within this context, the development of techniques to strength this cryptography method has both academic and commercial interest. This work presents the elliptic curve cryptography, with its arithmetic and vulnerabilities. The objective is to investigate a way of using rotations about the x axis to improve this cryptographic scheme. A new cryptography method was elaborated, which takes advantage of 3D rotations to increase the cryptoanalysis difficulty. Tests were conducted to indicate that the proposed method is capable of encrypting and decrypting messages.*

Resumo. *A criptografia por curvas elítpicas representa uma alternativa muito interessante ao tradicional método RSA. Ela utiliza a dificuldade de resolver o problema do logaritmo discreto em um corpo algébrico definido por uma curva elíptica. Dentro deste contexto, o desenvolvimento de técnicas que fortaleçam este criptográfico possui não só interesse acadêmico quanto comercial. Este trabalho apresenta a criptografia com curvas elípticas, juntamente com sua aritmética e suas vulnerabilidades. A proposta é investigar uma forma de utilizar rotações em torno do eixo x para fortalecer esse esquema de criptografia. Para isso, foi elaborado um novo método de criptografia com curvas elípticas, que utiliza essas rotações para aumentar a dificuldade de se fazer a criptoanálise. No final, testes realizados indicam que o método proposto foi capaz de codificar e decodificar mensagens.*

1. Introdução

A criptografia com curvas elípticas foi descoberta em 1985 por Neal Koblitz e Victor Miller, e seu funcionamento é baseado em outros esquemas de criptografia com chave pública. Esses é um tipo de criptografia assimétrico, ou seja, o método

usado para codificar uma mensagem não pode ser invertido para decodificá-la, garantindo, dessa forma, que os detalhes do funcionamento de tal método possam ser tornados públicos sem que sua segurança seja comprometida. No caso da criptografia com curvas elípticas, a assimetria está na forma que as operações entre os pontos da curva são definidas. Quando um determinado ponto é somado a ele mesmo n vezes, é difícil encontrar n a partir do resultado, pois não existe uma operação de divisão para reverter esse procedimento.

Esse tipo de criptografia apresenta uma grande vantagem quando comparado aos outros esquemas criptográficos de chave pública, pois ele é baseado no problema do logaritmo discreto sobre curvas elípticas. Como não existe uma forma de ataque em tempo sub-exponencial para esse problema, é possível utilizar chaves menores sem aumentar a sua vulnerabilidade. Por exemplo, uma chave RSA de 1024 bits equivale a uma chave de uma curva elíptica de 160 bits em termos de segurança [Wong, 2009].

O objetivo desse trabalho é de fortalecer o esquema de criptografia com curvas elípticas. Para isso, rotações ao redor do eixo x serão aplicadas aos pontos da curva, para que, dessa forma, seja criada uma ambiguidade quanto à origem de cada ponto que precisar ser tornado público, e dificultando a operação dos métodos de criptoanálise.

Conforme avanços tecnológicos permitem que os computadores fiquem mais rápidos, as chaves dos métodos de criptografia ficam mais vulneráveis e, por isso, devem ter sua quantidade de bits aumentada. Porém, chaves maiores possuem uma série de desvantagens, como a necessidade de mais espaço e o fato de apresentarem uma dificuldade maior de serem calculadas. Por isso, se faz necessário que métodos mais eficientes e robustos de criptografia sejam pesquisados. Neste cenário, o presente trabalho pretende contribuir para que a criptografia com curvas elípticas, que já é mais eficiente que outros métodos, possa ser fortalecida sem a necessidade de utilizar chaves maiores. Dessa forma, o surgimento de computadores mais avançados terá um impacto reduzido na segurança desse tipo de criptografia.

2. Criptografia com Curvas Elípticas

2.1. Corpos Finitos F_p

Um corpo finito é um conjunto munido de duas operações binárias: adição e multiplicação. Os copos finitos primos F_p possuem os seguintes elementos: $\{0, 1, ..., p\text{-}1\}$.

A adição entre dois elementos a e b pertencentes a F_p é dada por $a + b = r \pmod{p}$. Para realizar a subtração de a por b, basta realizar a soma de a com o negativo de b, que é a solução da equação $b + x = 0 \pmod{p}$.

A multiplicação entre dois elementos a e b pertencentes a F_p é dada por $ab = s \pmod{p}$. Para realizar a divisão de a por b, basta multiplicar a pelo inverso de b, que é a solução da equação $bx = 1 \pmod{p}$.

2.2. Curvas elípticas sobre corpos finitos do tipo F_p

Uma curva elíptica E sobre o corpo F_p é definida pela equação: $y^2 = x^3 + ax + b$. O conjunto $E(F_p)$ é composto por todos os pontos $(x, y) \in F_p$ que satisfazem à essa equação junto com o ponto no infinito O.

Para que a adição em $E(F_p)$ possa ser definida, é necessária a definição de um novo símbolo: Θ.

Definição 1: *Sejam (x_1, y_1) e $(x_2, y_2) \in E(F_p)$, define-se Θ da seguinte forma:*

- *Se $(x_1, y_1) = (x_2, y_2)$, então $\Theta = (3x^2 + a) / 2y$*
- Se *$(x_1, y_1) \neq (x_2, y_2)$, então $\Theta = (y_2 - y_1) / (x_2 - x_1)$.*

Com base em Θ, define-se a adição de pontos em $E(F_p)$:

Definição 2: *A adição de dois pontos em $E(F_p)$ é dada da seguinte forma:*

$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$, onde:

- $x_3 = \Theta^2 - x_1 - x_2$
- $y_3 = \Theta(x_1 - x_3) - y_1$.

O conjunto $E(F_p)$ com a operação de adição forma um grupo e sua identidade é ponto O . O negativo do ponto (x, y) é $(x, -y)$, pois $(x, y) + (x, -y) = O$.

Cada ponto na curva possui uma ordem, isto é, o tamanho do ciclo gerado quando ele é somado a si próprio n vezes. De acordo com Wong (2009), a ordem de um ponto é definida formalmente da seguinte forma:

Definição3: *Dado o ponto $P \in E(F_p)$, a ordem n de P é o menor inteiro que resolve: $nP = O$, onde O é o ponto no infinito.*

2.3. Métodos para curvas elípticas

Agora, será definido um sistema criptográfico análogo ao sistema proposto por ElGamal, baseado no Problema do Logaritmo Discreto [ElGamal, 1985].

Antes de criptografar uma mensagem, é necessário decidir qual tipo de corpo será utilizado, os parâmetros da curva elíptica E, e um ponto base $P \in E$, que serão tornados públicos. Cada usuário desse sistema deve, então, escolher um número n, que será sua chave secreta, e calcular nP, que será sua chave pública.

Assumindo que Alice queira mandar a mensagem m para Bob, ela deverá primeiro representar m através de um ponto P_m na curva E. Alice conhece a chave pública de Bob n_bP, então para criptografar P_m ela escolhe um inteiro k e envia para Bob o seguinte par de pontos: $(C_1, C_2) = (kP, P_m + k(n_bP))$. Para decifrar o texto, Bob calcula $C_2 - n_bC_1 = (P_m + k(n_bP)) - n_b(kP) = P_m$.

Existe, também, uma outra forma de criptografar dados para curvas elípticas, o método Menezes-Vanstone (Menezes e Vanstone, 1993). Nesse método, ao invés de ser enviado um par de pontos como no método anterior, é enviado o

conjunto com um ponto e dois números $Y = (y_0, y_1, y_2)$. A seguir, seu funcionamento será detalhado.

Primeiramente, define-se o corpo finito que será utilizado e os parâmetros da curva elíptica E, bem como um número aleatório k pertencente ao corpo finito escolhido. A chave secreta é s, e a chave pública é o par de pontos (Q, P), onde $P \in E$ e $Q = sP$. Sendo $X = (x_1, x_2) \in E$ a mensagem a ser criptografada, calcula-se componentes de Y que serão transmitidos em da seguinte forma:

- $y_0 = kP$
- $y1 = c_1x_1$
- $y2 = c_2x_2$

onde $(c_1, c_2) = kQ$.

Ao receber $Y = (y_0, y_1, y_2)$, a mensagem original $X = (x_1, x_2)$ é recuperada da seguinte forma:

- $x_1 = y_1(c_1)^{-1}$
- $x_2 = y_2(c_2)^{-1}$

onde $(c_1, c_2) = s*y_0$.

3. Métodos de Criptoanálise em Curvas Elípticas

3.1. Método Pollard rho

O método Pollard rho [Pollard, 1978] é usado para resolver o problema do logaritmo discreto. O algoritmo abaixo mostra como ele pode ser adaptado para resolver o problema do logaritmo discreto sobre curvas elípticas, conforme Hankerson, Menezes e Vanestone (2004):

Entrada: Curva elíptica $E(F_p)$, ponto base P de ordem n, chave pública Q.
Saída: Chave privada k.

1. Selecionar a quantidade L de subgrupos para dividir os pontos de $E(F_p)$.
2. Criar uma função de partição H para dividir os pontos de $E(F_p)$ entre os L subgrupos.
3. Repetir para j de 1 até L
3.1. Selecionar aleatoriamente $a_j, b_j \in [0,n-1]$.
3.2. Computar $R_j = a_j*P + b_j*Q$.
4. Selecionar aleatoriamente $c', d' \in [0,n-1]$ e computar $X' = c'*P + d'*Q$.
5. Atribuir os valores: $X''=X'$, $c''=c'$, $d''=d'$.
6. Repetir
6.1. Computar $j = H(X')$.
Atribuir valores: $X'=X'+R_j$, $c'=c'+a_j \bmod n$, $d'=d'+b_j \bmod n$.
6.2. Para i de 1 até 2:
Computar $j = H(X'')$.
Atribuir valores: $X''=X''+R_j$, $c''=c''+a_j \bmod n$, $d''=d''+b_j \bmod n$.
Até que $X' = X''$.
7. Se $d' = d''$ retornar "falha".

3.2. Método Pohlig-Hellman

O método Pohlig-Hellman foi proposto como uma forma de resolver o problema do logaritmo discreto sobre F_p, com complexidade $O(\log^2 p)$ se p-1 possuir fatores

primos pequenos [Pohlig, Hellman, 1978]. O algoritmo exibido na próxima página, baseado em Barker (2008), mostra como ele pode ser usado para resolver o problema do logaritmo discreto sobre curvas elípticas.

Entrada: Curva elíptica $E(F_p)$, ponto base P de ordem n, chave pública Q.
Saída: Chave privada k.
1. Fatorizar n em seus r fatores: $l_1^{e1}, l_2^{e2}, l_3^{e3} \ldots l_r^{er}$.
2. Repetir para j de 1 até r
    2.1. Definir as constantes $P_0 = n/l_j * P$, $Q_0 = n/l_j * Q$ e $k_j = 0$.
    2.2. Criar uma lista $T = \{a*P_0 \mid 0 \leq a \leq lj-1\}$.
    2.3. Repetir para i de 0 até e_j-1
        2.3.1. Se i > 0, $Q_i = n/l_j^{i+1} * (Q_{i-1} - k_{i-1} * l^{i+1} * P)$.
        2.3.2. Encontrar na lista T o elemento a_i que satisfaça: $Q_i = a_i P_0$.
        2.3.3. Calcular $k_j = k_j + a_i * l^i$.
3. Resolver o sistema de congruências $k = k_1 \bmod l_1^{e1}$ (...) $k = k_r \bmod l_r^{er}$ para encontrar a

3.3. Método *Baby Step Giant Step*

Esse método, desenvolvido por Shanks (1971), resolve o problema do logaritmo discreto em $\sqrt{n}$ passos armazenando $\sqrt{n}$ valores. O algoritmo a seguir, baseado em Barker (2008), mostrada como aplicá-lo ao problema do logaritmo discreto sobre curvas elípticas:

Entrada: Curva elíptica $E(F_p)$, ponto base P de ordem n, chave pública Q.
Saída: Chave privada k.

1. Selecionar $m \in Z$ de forma que $m \geq \sqrt{n}$ e calcular $m*P$.
2. Cria uma lista $T = \{i*P \mid 0 \leq i < m\}$.
3. Para j de 0 até m - *1*
    3.1. Calcular $Q' = Q - j*m*P$.
    3.2. Se houver na lista T um elemento $i*P$ tal que $i*P = Q'$,
        retornar: $k = i + i*m \bmod n$.

4. Proposta de Fortalecimento do Esquema de Curvas Elípticas

4.1. Introdução

O objetivo aqui é propor uma forma de fortalecer a criptografia com curvas elípticas. Para isso, será introduzida uma maneira de utilizar rotações ao torno do eixo x, de forma que se possa mascarar os pontos da curva. Em seguida, será proposto um método de criptografia para trabalhar com essas rotações.

4.2. Rotação espacial de uma curva elíptica

Um ponto pode ser rotacionado ao redor dos eixos x, y ou z, através de uma matriz de rotação. Os pontos de uma curva elíptica serão rotacionados ao redor do eixo x, através da matriz de rotação definida a seguir.

Definição 3: *Dado um ponto P = (x, y, z), o ponto rotacionado ao redor do eixo x P' = (x, y', z'), com um ângulo θ, é dado por P' = R_x*P, onde R_x é a seguinte matriz de rotação:*

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\text{sen}\,\theta \\ 0 & \text{sen}\,\theta & \cos\theta \end{pmatrix}$$

Para reverter esta transformação, basta multiplicar P' pela matriz transposta, isto é, $P = P'*R_x^T$.

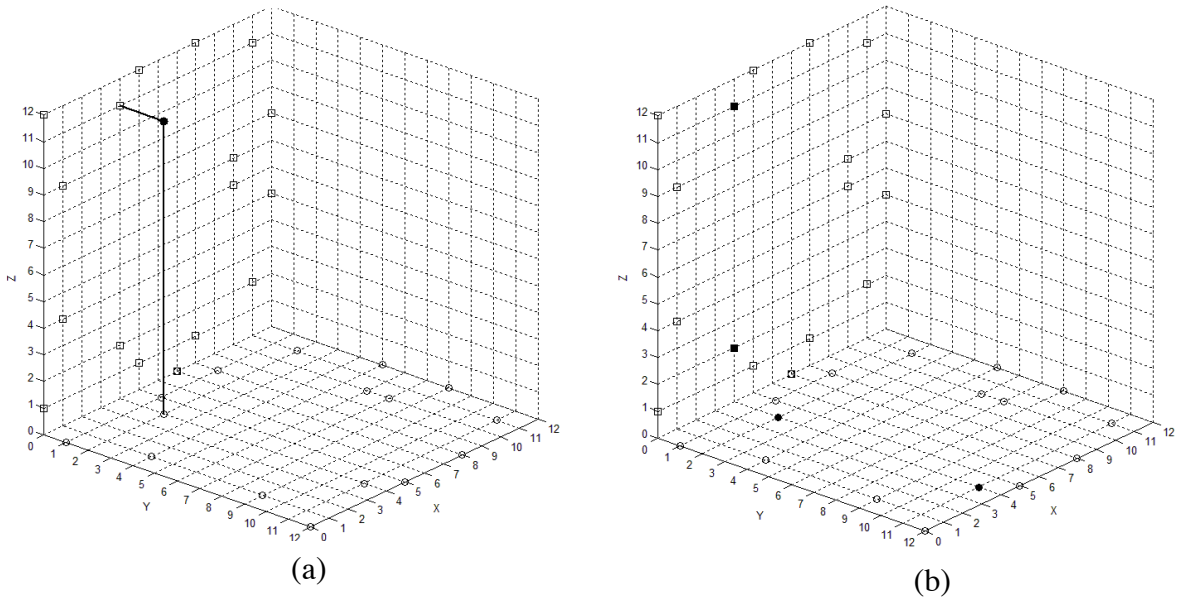
Para manter as coordenadas dos pontos da curva elíptica no conjunto dos números inteiros, serão usados quatro tipos de rotações: 0º, 90º, 180º e 270º. Dessa forma, cada ponto (x, y) da curva elíptica, depois de rotacionado, terá quatro possibilidades: (x, y, 0), (x, 0, y), (x, -y, 0) e (x, 0, -y).

4.3. Como usar a rotação para aumentar as possibilidades de criptografia

O objetivo aqui é, através das rotações, tentar mascarar os pontos da curva elíptica.

Um ponto (*x*, *y*) rotacionado 270º, por exemplo, é representado como (*x*, 0, -*y*). Ao analisar-se esse ponto sem saber a rotação que foi aplicada a ele, tem-se duas possibilidades ambíguas de sua origem: (*x*, 0, *y*) e (*x*, 0, -*y*).

Os pontos rotacionados serão representados da seguinte forma: se o ponto for do tipo (*x*, 0, *z*), ou seja, se ele estiver no plano *xz*, ele será representado como (*x*, -*z*, *z*). Caso ele seja do tipo (*x*, *y*, 0), ou seja, se ele estiver no plano *xy*, ele será representado como (*x*, *y*, -*y*). Para devolver o ponto ao seu plano original, basta igualar sua componente *z* a zero, caso a rotação tenha sido de 0º ou 180º, ou igualar sua componente *y* a zero, caso a rotação tenha sido de 90º ou 270 º. A figura 1 ilustra esse procedimento.



(a) (b)

Figura 1: (a) Ponto rotacionado e projetado para fora dos planos. (b) Os quatro pontos referentes às possíveis origens da rotação e projeção.

4.4. Método de Criptografia Proposto

Agora, será definido um novo método que utiliza rotações em torno do eixo x para aumentar a segurança do método de curvas elípticas.

Será necessário escolher o tipo de corpo que será utilizado, os parâmetros da curva elíptica E, um ponto base $P \in E$. As chaves públicas, porém, serão definidas de uma maneira diferente. Além de escolher uma chave privada s, cada usuário desse sistema deverá escolher uma rotação R' para aplicar ao ponto P, obtendo o ponto P'. Então, a chave pública Q' será o ponto sP'. Será necessário, também, um segundo par de chave pública e chave privada, portanto cada usuário deverá ainda escolher outro número para ser sua chave privada n, e multiplicar por P, obtendo a segunda chave pública Q.

Agora, será mostrado como a rotação que será utilizada no envio de mensagens é definida. Cada uma das possíveis rotações deve ser atribuída a um ponto na curva. Então, basta enviar o ponto referente à rotação R escolhida através do método de ElGamal, utilizando a segunda chave pública definida pelo outro usuário.

Após se ter definido a rotação R, o envio de mensagens será feito da seguinte forma: primeiro, deve-se aplicar a rotação R às duas possibilidades da chave pública do destinatário, obtendo Q'' e Q'''. Deve-se, também, escolher um inteiro k aleatório. Então, a mensagem $X = (x_1, x_2)$ será enviada através da 4-upla $Y = (y_0, y_1, y_2, y_3)$, que é calculado da seguinte forma:

- $y_0 = RkP$
- $y_1 = c_1 x_1$
- $y_2 = c_2 x_2$
- $y_3 = c_3 x_2$

onde $(c_1, c_2) = kQ''$, $(c_1, c_3) = kQ'''$ e y_0 tem sua coordenada $y = -z$ caso $y = 0$ ou $z = -y$, caso contrário. Para evitar que y_2 seja igual a y_3, quando a rotação R for de 90º, o y_3 será calculado como $y_3 = c_3(-x_2)$.

Ao receber $Y = (y_0, y_1, y_2, y_3)$, a mensagem $X = (x_1, x_2)$ é recuperada da seguinte forma: primeiro, y_0 deve ser retornado ao seu plano original igualando sua coordenada z a 0. Caso a rotação R seja de 0 ou 180 graus, ou, caso contrário, igualando sua coordenada y a 0. Em seguida, caso o resto da divisão do ângulo de R somado ao ângulo de R' do destinatário por 180 for 0:

- $x_1 = y_1(c_1)^{-1}$
- $x_2 = y_2(c_2)^{-1}$.

Caso contrário:

- $x_1 = y_1(c_1)^{-1}$
- $x_2 = y_3(c_3)^{-1}$

onde $(c_1, c_2, c_3) = s*R'*y_0$.

Caso a rotação R tenha sido de 90º, $x_2 = -x_2$.

5. Implementação

Agora, serão apresentados trechos de uma implementação em Java do método proposto, começando pela combinação da rotação entre os usuários. Para armazenar os quatro pontos referentes às possíveis rotações, foi criada a lista *lr*. Os pontos armazenados nas posições 0, 1, 2 e 3 correspondem, respectivamente, às rotações de 0º, 90º, 180º e 270º. Para representar a curva elíptica e sua aritmética, foi criada a classe CurvaEliptica. Uma instância dessa classe, chamada *e*, é utilizada para realizar os cálculos.

```
Ponto pm;
Ponto c1, c2;
pm = lr.get(rt);
Random rn = new Random(System.nanoTime());
int k = Math.abs(rn.nextInt()%e.order(p));
c1 = e.multiplica(k, p);
c2 = e.soma(pm, e.multiplica(k, x.getQ1()));
```

Nessa parte do código, são calculados os pontos C_1 e C_2 do método, recebendo como parâmetros o destinatário *x* e o inteiro *rt*, referente à rotação a ser combinada. O ponto *pm*, referente à mensagem a ser enviada, é retirado da posição *rt* da lista *lr*. O inteiro aleatório *k* é calculado usando a classe *Random*, e seu valor é limitado até a ordem do ponto base *p*, pois a partir daí os valores de *k*p* começam a se repetir. O método x.getQ1() retorna a primeira chave pública do usuário *x*.

```
Ponto pd= e.soma(c2, e.invPonto(e.multiplica(x.getN(), c1)));
int r1= lr.indexOf(pd);
```

Aqui, a mensagem é decodificada e armazenada no ponto *pd*, utilizando x.getN() para representar a chave privada de *x*. Então, o índice desse ponto na lista *lr* é armazenado na variável *r1*, que representa a rotação que acaba de ser combinada entre os usuários.

A seguir, será mostrada a implementação da troca de mensagens secretas.

```
Ponto3d q3, q4, c1, c2;
q3 = e.rotaciona(r1, new Ponto3d(x.getQ2().getX(), x.getQ2().getY(), 0));
q4 = e.rotaciona(r1, new Ponto3d(x.getQ2().getX(), 0, x.getQ2().getZ()));
c1 = e.multiplica3d(k, q3);
c2 = e.multiplica3d(k, q4);
int k = Math.abs(rn.nextInt()%e.order(p));
```

No trecho de código mostrado na próxima página, são calculados os valores que servirão de base para criar a quádrupla *Y*.

```
Ponto3d y0= e.rotaciona(r1, e.multiplica(k, p));
int y1, y2, y3;
if (r1 == 0 || r1 == 2) {
   y0.setZ(e.calcMod(-y0.getY()));
   y1 = e.calcMod(c1.getX() * px.getX());
   y2 = e.calcMod(c1.getY() * px.getY());
   y3 = e.calcMod(c2.getZ() * px.getY());
} else {
   y0.setY(e.calcMod(-y0.getZ()));
   y1 = e.calcMod(c1.getX() * px.getX());
   y2 = e.calcMod(c1.getZ() * px.getY());
   if(r1==1)
      y3 = e.calcMod(c2.getY() * e.calcMod(-px.getY()));
   else
      y3 = e.calcMod(c2.getY() * px.getY());
}
```

Aqui, os componentes de *Y* são calculados. O ponto *y0* é projetado para fora do plano *xy* ou *xz*, dependendo da rotação escolhida, e a mensagem *px* é embutida nos números *y1*, *y2* e *y3*, utilizando como base os pontos *c1* e *c2*.

```
Ponto3d c3;
Ponto pd;
if (r1 == 0 || r1 == 2) {
   y0.setZ(0);
} else {
   y0.setY(0);
}
c3 = e.multiplica3d(x.getS(), e.rotaciona(x.getRp(), y0));
if ((x.getRp() + r1) % 2 == 0) {
   x1 = e.calcMod(y1 * e.calcInvd(c3.getX()));
   x2 = e.calcMod(y2 * e.calcInvd(c3.getY()));
} else {
   x1 = e.calcMod(y1 * e.calcInvd(c3.getX()));
   x2 = e.calcMod(y3 * e.calcInvd(c3.getZ()));
   if(r1==1){
      x2 = e.calcMod(-x2);
   }
}
pd = new Ponto(x1,x2);
```

Aqui, a mensagem é decodificada e armazenada no ponto *pd*. Inicialmente, o ponto *y0* é projetado de volta para o plano xy ou xz, de acordo com a rotação combinada. Em seguida, é calculado o ponto *c3*, que é composto pelos números *c1*, *c2* e *c3* do método. Finalmente, os componentes de *pd*, *x1* e *x2*, são calculados de acordo com *r1*, a rotação combinada, e *rp*, a rotação privada do usuário que recebeu a mensagem.

6. Testes

Agora, serão realizados testes do método proposto com diferentes parâmetros, utilizando a implementação descrita na seção anterior.

Teste 1: o objetivo desse teste é verificar se o método realiza com sucesso o envio do ponto (629, 347). Para isso, será usada a curva elíptica E: $y^2 = x^3 + 33x + 33$ definida sobre o corpo finito F_{701}. O ponto P foi definido como (470, 82). O destinatário escolheu suas chaves privadas $n = 97$ e $s = 232$, sua rotação $rp = 180º$, e publicou suas chaves públicas $Q = (75, 147)$ e $Q' = (638, 344, 357)$.

A combinação da rotação de 90º é feita através do ponto (697, 347). Esse ponto é codificado nos pontos (C_1, C_2) = ((286, 137), (30, 191)). Ao recebê-los, o destinatário os decodifica e encontra o ponto (697, 347) = 90º.

O envio do ponto (629, 347) é feito através da 4-upla (y_0, y_1, y_2, y_3) = ((682, 87, 614), 483, 563, 138). Ao receber esses valores, efetua a decodificação e encontra o ponto (629, 309).

Resultado do teste: O ponto (629, 347) foi codificado, enviado e decodificado com sucesso.

Teste 2: o objetivo desse teste é verificar se o método envia com sucesso uma série de pontos, formando a mensagem "SOS". A curva elíptica a ser usada é a E: $y^2 = x^3 + 87x + 97$, definida sobre o corpo finito F_{1373}. O ponto P foi definido como (1107, 76). O destinatário escolheu suas chaves privadas $n = 399$ e $s = 578$, sua rotação $rp = 0º$, e publicou suas chaves públicas $Q = (1014, 417)$ e $Q' = (661, 1068, 305)$. A letra "S" foi mapeada no ponto (518, 479), e a letra "O" foi mapeada no ponto (518, 894).

A combinação da rotação de 180º é feita através do ponto (1187, 1152). Esse ponto é codificado nos pontos (C_1, C_2) = ((442, 1318), (677, 1150)). Ao recebê-los, o destinatário os decodifica e encontra o ponto (1187, 1152) = 180º.

A mensagem "SOS" e enviada através dos pontos (518, 479), (518, 894) e (518, 479). A codificação desses pontos é feitas nas três 4-uplas: (y_0, y_1, y_2, y_3) = ((1038, 941, 432), 766, 658, 715), ((160, 324, 1049), 492, 633, 740) e ((223, 437, 936), 830, 194, 1179). Ao receber esses valores, o destinatário os decodifica, encontrando os pontos (518, 479), (518, 894) e (518, 479), que correspondem, respectivamente, às letras "S", "O" e "S".

Resultado do teste: a sequência de pontos foi codificada, enviada e decodificada com sucesso.

7. Conclusões e trabalhos futuros

A criptografia com curvas elípticas baseia-se em uma aritmética que envolve os pontos da curva em um corpo finito, na qual a operação de multiplicação entre um inteiro e um ponto não é reversível. Os métodos de criptografia aproveitam esse fato para associar mensagens a pontos da curva, e combinar esses pontos a um outro ponto de forma que só alguém que possua uma

chave específica consiga desfazer essa operação. Com isso, surge o problema do logaritmo discreto sobre curvas elípticas, isto é, dados os pontos P e $Q \in E(F_p)$, encontrar um inteiro k tal que $kP = Q$. Evidências apontam ao fato de que esse problema seja intratável, apesar de não existirem provas matemáticas que comprovem isso.

A proposta apresentada nesse trabalho apresenta a vantagem de incluir um novo fator aleatório ao método, a rotação. Dessa forma, é criada uma ambiguidade na hora de se tentar quebrar a chave pública Q', pois não se sabe exatamente qual é a rotação aplicada ao o ponto P para tentar resolver $sP' = Q'$. As mensagens enviadas também se tornam mais difíceis de serem quebradas individualmente, pois ainda há uma outra rotação envolvida nesse processo, que deve ser combinada em segredo entre os usuários. O fato dessas rotações serem simples de se calcular também é uma vantagem, pois dessa forma o método não tem sua complexidade aumentada de forma significativa. Outra vantagem é o fato do método de envio de mensagens estar desacoplado do método de estabelecimento de rotações em comum, dando uma liberdade para que os usuários combinem novas rotações sem afetar o envio de mensagens.

Existem, porém, algumas limitações relacionadas ao método proposto. Uma delas é a necessidade de se estabelecer uma rotação em comum entre os usuários, e isso deve ocorrer em segredo. Esse procedimento é feito utilizando-se um método clássico de criptografia com curvas elípticas, tonando-o vulnerável aos métodos de criptoanálise. Outra limitação é o fato dessa rotação servir apenas para um par de usuários. Por exemplo, se Bob estiver recebendo mensagens de cem usuários diferentes e perder a rotação combinada de um deles, ele não poderá decifrar as mensagens recebidas desse usuário e deverá estabelecer uma nova rotação em comum. Existe, também, a limitação de rotações possíveis. Devido ao fato da aritmética que envolve os pontos das curvas elípticas servir apenas para números inteiros positivos, qualquer rotação que colocar as coordenadas dos pontos no conjunto dos números reais necessitará de um tratamento especial, pois qualquer adaptação nos cálculos estará sujeita a erros de precisão e arredondamento.

Um possível trabalho futuro seria explorar outras formas de transformação que podem ser aplicadas a um ponto e estudar as possíveis formas de utilizá-las para fortalecer a criptografia com curvas elípticas. Novas formas de representar os pontos da curva também podem ser exploradas para esse fim, mas é possível que seja necessário realizar adaptações na aritmética para que se possa trabalhar com essas alterações.

Outra possibilidade de trabalho futuro seria uma implementação do método proposto em hardware. A aritmética em corpos binários é mais fácil de ser implementada em hardware, tornando essa uma boa oportunidade para se realizar uma adaptação no método proposto para trabalhar com curvas elípticas definidas sobre os corpos binários, isto é, do tipo F_{2^m}.

Referências Bibliográficas

BARKER, Nathan. "Elliptic Curves, Factorization and Cryptography". University of Durham, 2008.

ELGAMAL, Taher. "A Public Key Cryptosystem And A Signature Scheme Based On Discrete Logarithms". California, 1985.

HANKERSON, Darrel; MENEZES, Alfred; VANSTONE, Scott. "Guide to Elliptic Curve Cryptography". New York:Springer-Verlag, 2004.

LAMB, Nicolas. "An investigation into Pollard's Rho method for attacking Elliptic Curve Cryptosystems". 2002.

POHLIG, S.; HELLMAN, M". An Improved Algorithm for Computing Logarithms Over GF(p) and its Cryptographyc Significance". *IEEE Transactions on Information Theory*. n. 24, p. 106-110, 1978.

POLLARD, J. M. "Monte Carlo methods for index computation mod p". *Mathematics of Computation*. v.32, n. 143, p. 918-924, 1978.

SHANKS, D. "Class number, a theory of factorization and genera". *Proc. Symp. Pure Math*. v. 20, p. 415–440, 1971.

WONG, David. "Elliptic Curves, Cryptography and Factorization". Department of Mathematics–Durham University, 2009.