*INTERMATHS, VOL. 5, NO. 2 (**2024**)*, **69–79**
https://doi.org/10.22481/intermaths.v5i2.15134
Artigos Gerais
licença creative commons

Sistemas Criptográficos usando Curvas Elípticas

Cryptographic systems using Elliptic Curves

Jaime Edmundo Apaza Rodriguez [a,*]

[a] Universidade Estadual Paulista Júlio de Mesquita Filho: Sao Paulo, SP, BR

* Autor Correspondente: jaime.rodriguez@unesp.br

Resumo: Neste trabalho iremos discutir e comparar alguns Criptosistemas baseados em Curvas Elípticas. E por quê usar as Curvas Elípticas em Criptografia? O motivo principal é que elas fornecem segurança equivalente aos sistemas clássicos usando menos bits. Por exemplo, em [1] foi estimado que o tamanho de uma chave de 4096 bits para o sistema RSA fornece o mesmo nível de segurança do que 313 bits num sistema usando Curvas Elípticas. Isto significa que a implementação para sistemas com Curvas Elípticas requer chips de menor tamanho, menor consumo de energia, entre outros fatores. Em [4], os autores fizeram um experimento num pequeno dispositivo portátil (3Com's Palm Pilot) maior que um cartão inteligente mas menor que um laptop. Eles verificaram que gerar uma chave de 512-bit RSA toma 3,4 minutos, enquanto gerar uma chave de 163-bit no sistema ECC-DSA toma 0,597 segundos. Embora certos procedimentos, como verificação de assinaturas, tenham sido um pouco mais rápidos para o RSA, os métodos com Curvas Elípticas, como ECC-DSA, oferecem maior velocidade em diversas situações.

Palavras-chave: Curvas Elípticas; Criptografia; Sistemas Criptográficos.

Abstract: In this work we will discuss and compare some Cryptosystems based on Elliptic Curves. And why use Elliptic Curves in Cryptography? The main reason is that they provide equivalent security to classical systems using fewer bits. For example, in [1] it was estimated that a key size of 4096 bits for the RSA system provides the same level of security as 313 bits in a system using Elliptic Curves. This means that the implementation for systems with Elliptic Curves requires smaller chips, lower power consumption, among other factors. In [4], the authors did an experiment on a small portable device (3Com's Palm Pilot) larger than a smart card but smaller than a laptop. They found that generating a 512-bit RSA key takes 3.4 minutes, while generating a 163-bit key in the ECC-DSA system takes 0.597 seconds. Although certain procedures, such as signature verification, were slightly faster for RSA, Elliptic Curve methods, such as ECC-DSA, offer greater speed in several situations

keywords: Elliptic Curves; Cryptography; Criptography Systems.

1 Introdução

A situação clássica no envio de mensagens entre duas pessoas pode ser colocada no seguinte contexto. Alice deseja enviar uma mensagem para Bob. Para evitar que Eva, a intrusa, leia a mensagem, Alice criptografa a mensagem para obter o texto cifrado. Quando Bob recebe o

Submetido em: 30 de Julho de 2024 Aprovado em: 25 de Dezembro de 2024 Publicado em: 31 de Dezembro de 2024
ISSN 2675-8318 ©2024 INTERMATHS. Publicado por Edições UESB. Este é um artigo de acesso aberto sob a licença CC BY 4.0.

texto cifrado, ele descriptografa e lê a mensagem. Para criptografar a mensagem, Alice usa uma Chave de encriptação. Bob usa uma chave de descriptografia para descriptografar o texto cifrado. Obviamente a chave de descriptografia deve ser mantida em segredo.

Existem dois tipos básicos de Criptografia. Na Criptografia Simétrica ou de Chave Privada, onde a chave de criptografia e a chave de descriptografia são iguais ou uma pode ser facilmente deduzida da outra. Os métodos populares de Criptografia Simétrica incluem o Data Padrão de Criptografia (DES) e o Padrão de Criptografia Avançado (AES). Nesse caso, Alice e Bob precisam encontrar alguma forma de estabelecer uma chave. Por exemplo, Bob poderia enviar um mensageiro para Alice com vários dias de antecedência. Então, quando chegar a hora de enviar a mensagem, ambos terão a chave. É evidente que isto é impraticável em muitas situações.

O outro tipo de Criptografia é a Criptografia de Chave Pública ou Criptografia Assimétrica. Neste caso, Alice e Bob não precisam ter contato prévio. Bob publica uma chave de Criptografia Pública, que Alice usa. Ele também tem uma chave de descriptografia privada, que lhe permite descriptografar os textos cifrados. Já que todo mundo sabe a Chave de Criptografia, deveria ser inviável deduzir a chave de descriptografia. O sistema de Chave Pública mais famoso é conhecido como RSA e está baseado na dificuldade de fatorar números inteiros em primos. Outro sistema bem conhecido é devido ao ElGamal e está baseado no Problema do Logaritmo Discreto.

Geralmente, os Sistemas Assimétricos são mais lentos que os bons Sistemas Simétricos. Portanto, é comum usar um Sistema Assimétrico para estabelecer uma chave que será usada num Sistema Simétrico. A melhoria na velocidade é importante quando grandes quantidades de dados estão sendo transmitidas.

2 Troca de chaves Diffie-Hellman

Alice e Bob querem chegar a um acordo sobre uma chave comum que possam usar para troca de dados por meio de um esquema de Criptografia Simétrica, como o DES ou o AES. Por exemplo, Alice e Bob poderiam ser bancos que desejam transmitir dados financeiros. É impraticável e demorado usar um mensageiro para entregar a chave. Além disso, suponha que Alice e Bob não tiveram contato prévio e, portanto, os únicos canais de comunicação entre eles são públicos. Uma maneira de estabelecer uma chave secreta é dada pelo seguinte método, devido a Diffie e Hellman (na verdade, eles usam grupos multiplicativos de corpos finitos). Tal método consiste dos seguintes passos:

1. Alice e Bob escolhem uma curva elíptica E sobre o corpo finito $\mathbf{F}_q$ de modo que o Problema do Logaritmo Discreto seja difícil de resolver em $E(\mathbf{F}_q)$. Eles também escolhem um ponto $P \in E(\mathbf{F}_q)$ de modo que o subgrupo gerado por P tenha ordem grande (geralmente, a curva e o ponto são escolhidos de modo que a ordem desse subgrupo seja um primo grande).
2. Alice escolhe um inteiro secreto a, calcula $P_a = aP$ e envia P_a para Bob.
3. Bob escolhe um inteiro secreto b, calcula $P_b = bP$ e envia P_b para Alice.
4. Alice calcula $aP_b = abP$.
5. Bob calcula $bP_a = baP$.
6. Alice e Bob usam algum método acordado publicamente para extrair uma chave de abP.

Por exemplo, eles poderiam usar os últimos 256 bits da coordenada x de abP como chave ou poderiam avaliar uma função hash na coordenada x.

As únicas informações que a intrusa Eva tem é a curva E, o corpo finito $\mathbf{F}_q$ e os pontos P, aP *e* bP. Para ela descifrar a mensagem, precisa resolver o seguinte problema:

2.1 Problema de Diffie-Hellman

Dados $P,\ aP\ e\ bP$ em $E(\mathbf{F}_q)$, calcular abP

Se Eva puder resolver logaritmos discretos em $\mathbf{F}_q$, então ela poderá usar $P\ e\ aP$ para encontrar a. Então ela pode calcular $a(bP)$ para obter abP. Contudo, não se sabe se existe alguma maneira de calcular abP sem primeiro resolver o problema de logaritmo discreto.

Uma questão relacionada é a seguinte:

2.2 O problema de Decisão de Diffie-Hellman

Dados $P,\ aP\ e\ bP$ em $E(\mathbf{F}_q)$, e dado um ponto $Q \in E(\mathbf{F}_q)$ determine se $Q = abP$

Em outras palavras, se Eva receber uma denúncia anônima informando seu abP, ela poderá verificar se as informações estão corretas?

Os Problemas Diffie-Hellman e de Decisão Diffie-Hellman podem ser apresentados para grupos arbitrários. Originalmente, eles apareceram no contexto de grupos multiplicativos $\mathbf{F}_q^*$ de corpos finitos.

Para Curvas Elípticas, o emparelhamento de Weil pode ser usado para resolver o Problema de Decisão de Diffie-Hellman em alguns casos.

Exemplo: Seja a curva $E : y^2 = x^3 + 1$ definida sobre $\mathbf{F}_q$, onde $q \equiv 2\ (mod\ 3)$. Esta curva é supersingular ($E[p] = \{\infty\}$, com $q = p^r$, ou seja, a curva não possui pontos de ordem p). Seja $\omega \in \mathbf{F}_{q^2}$ uma raiz cúbica primitiva da unidade. Observar que $\omega \notin \mathbf{F}_q$ pois a ordem de $\mathbf{F}_q^*$, é $q-1$, o qual não é múltiplo de 3. Defina-se a aplicação

$$\beta : E(\overline{\mathbf{F}_q}) \longrightarrow E(\overline{\mathbf{F}_q}),\quad (x,y) \longmapsto (\omega x, y),\ \ \beta(\infty) = \infty.$$

Verifica-se que β é um isomorfismo (usando as leis de adição). Se $P \in E(\overline{\mathbf{F}_q})$ tem ordem n, então $\beta(P)$ também tem ordem n.

Agora defina-se o emparelhamento modificado de Weil

$$\tilde{e}_n(P_1, P_2) = e_n(P_1, \beta(P_2)),$$

onde e_n é é o emparelhamento usual de Weil e $P_1, P_2 \in E[n]$.

Lema: Suponha que $mdc(3, n) = 1$ e $P \in E(\mathbf{F}_q)$ de ordem n. Então $\tilde{e}_n(P, P)$ é uma n-ésima raiz primitiva da unidade.

Demonstração: Suponha que $aP = b\beta(P)$, para alguns inteiros a, b. Então

$$\beta(bP) = b\beta(P) = aP \in E(\mathbf{F}_q).$$

Se $bP = \infty$, então $aP = \infty$ e assim $a \equiv 0(mod\ n)$. Se $bP \neq \infty$, escrevemos $bP = (x, y)$, com $x, y \in \mathbf{F}_q$. Neste caso temos

$$(\omega x, y) = \beta(bP) \in E(\mathbf{F}_q).$$

Dado que $\omega \notin \mathbf{F}_q$, deve se ter $x = 0$. Por tanto $bP = (0, \pm 1)$, e tem ordem 3. Mas isto é impossível pois temos assumido que $mdc(3, n) = 1$. Disto segue se que a única relação da forma $aP = b\beta(P)$ deve ter $a, b \equiv 0(mod\ n)$, de modo que P e $\beta(P)$ formam uma base para o espaço $E[n]$, o que garante que $\tilde{e}_n(P, P) = e_n(P, \beta(P))$ seja uma raiz primitiva n-ésima da

unidade. Assim concluímos a prova.

Suponha agora que conhecemos $P,\ aP,\ bP\ e\ Q$ e queremos decidir se $Q = abP$. Primeiro, usamos o emparelhamento usual de Weil para decidir se Q é um múltiplo de P. Sabemos que Q é um múltiplo de P se e somente se $e_n(P, Q) = 1$. Suponha que este seja o caso, ou seja, $Q = tP$, para algum t. Assim temos

$$\tilde{e}_n(aP, bP) = \tilde{e}_n(P, P)^{ab} = \tilde{e}_n(P, abP)$$

e

$$\tilde{e}_n(Q, P) = \tilde{e}_n(P, P)^t.$$

Supondo que $mdc(3, n) = 1$, temos que $\tilde{e}_n(P, P)$ é uma raíz primitiva n-ésima da unidade e assim

$$Q = abP \iff t = ab\ (mod\ n) \iff \tilde{e}_n(aP, bP) = \tilde{e}_n(Q, P).$$

Isto resolve o Problema de Decisão de Diffie-Hellman neste caso. Observar que não foi necessário calcular nenhum logaritmo discreto, mesmo em corpos finitos. Apenas foi necessário calcular o emparelhamento de Weil.

Joux ([8]) mostrou uma outra aplicação do emparelhamento modificado de Weil que é conhecido como troca tripartida de chaves Diffie-Hellman. Este método funciona assim. Suponha que Alice, Bob e Chris queiram estabelecer uma chave comum. O procedimento padrão Diffie-Hellman requer duas rodadas de interação. O modificado permite que isso seja reduzido para uma rodada. Como acima, seja E a curva $y^2 = x^3 + 1$ sobre $\mathbf{F}_q$, onde $q \equiv 2(mod\ 3)$. Seja P um ponto de ordem n. Normalmente, n deve ser escolhido como sendo um número primo grande. Então Alice, Bob e Chris fazem o seguinte:

1. Alice, Bob e Chris escolhem os inteiros secretos $a,\ b\ e\ c,\ mod\ n$, respectivamente.
2. Alice transmite aP, Bob transmite bP e Chris transmite cP.
3. Alice calcula $\tilde{e}_n(bP, cP)$a, Bob calcula $\tilde{e}_n(aP, cP)b$ e Chris calcula $\tilde{e}_n(aP, bP)c$.
4. Como cada um dos três usuários calculou o mesmo número, eles usam este número para produzir uma chave, usando algum método pre-combinado publicamente.

Dado que E é supersingular, o problema de logaritmo discreto em E pode ser reduzido a um problema de logaritmo discreto para $\mathbf{F}^*_{q^2}$. Portanto, q deve ser escolhido o suficiente grande para que esse problema do logaritmo discreto seja difícil.

3 Criptografia Massey-Omura

Alice deseja enviar uma mensagem para Bob através de canais públicos. Eles ainda não estabeleceram uma chave privada. Uma maneira de fazer isso é a seguinte. Alice coloca sua mensagem em uma caixa, coloca um cadeado nela e envia para Bob. Bob coloca seu cadeado nela e a envia de volta para Alice. Alice então tira o cadeado e envia a caixa de volta para Bob. Bob então remove o cadeado, abre a caixa e lê a mensagem.

Este procedimento pode ser implementado matematicamente como segue.

1. Alice e Bob tomam uma curva elíptica E sobre um corpo finito $\mathbf{F}_q$, tal que o problema do logaritmo discreto em $E(\mathbf{F}_q)$ seja difícil. Seja $N = card\ E(\mathbf{F}_q)$.
2. Alice representa sua mensagem como um ponto $M \in E(\mathbf{F}_q)$.
3. Alice escolhe um número inteiro secreto m_A tal que $mdc(m_A, N) = 1$, calcula $M_1 = m_A M$ e o envia a Bob.
4. Bob escolhe um número inteiro secreto m_B tal que $mdc(m_B, N) = 1$, calcula $M_2 = m_B M$

e o envia a Alice.

5. Alice calcula $m_A^{-1} \in \mathbf{Z}_N$, logo calcula $M_3 = m_A^{-1} M_2$ e o envia a Bob.
6. Bob calcula $m_B^{-1} \in \mathbf{Z}_n$, logo calcula $M_4 = m_B^{-1} M_3$. Então $M_4 = M$ é a mensagem.

Vamos verificar que M_4 é a mensagem original M. Formalmente temos

$$M_4 = m_B^{-1} m_A^{-1} m_B m_A M = M.$$

De fato, temos $m_A^{-1} m_A \equiv 1 (mod\ N)$, de modo que $m_A^{-1} m_A = 1 + kN$, para algum k. Dado que o grupo $E(\mathbf{F}_q)$ tem ordem N, pelo Teorema de Lagrange temos que $NR = \infty$, para todo $R \in E(\mathbf{F}_q)$. Portanto

$$m_A^{-1} m_A R = (1 + kN)R = R + k\infty = R.$$

Aplicando isto para $R = m_B M$ temos que

$$M_3 = m_A^{-1} m_B m_A M = m_B M,$$

ou seja, m_A e m_A^{-1} cancelam. De modo análogo, m_B e m_B^{-1} cancelam e assim

$$M_4 = m_B^{-1} M_3 = m_B^{-1} m_B M = M.$$

A intrusa Eva conhece então $E(\mathbf{F}_q)$, e os pontos $m_A M, m_B m_A M$ *e* $m_B M$. Sejam $a = m_A^{-1}$, $b = m_B^{-1}$, $P = m_A m_B M$. Então temos que Eva conhece $P, bPaP$ e deseja encontrar abP. Este é o Problema de Diffie-Hellman.

O método acima é aplicável em qualquer grupo finito, mas raramente usado na prática.

Agora, como representar a mensagem como sendo um ponto de uma curva elíptica? Usaremos o método proposto por Koblitz ([5]). Seja a curva $E : y^2 = x^3 + Ax + B$ definida sobre $\mathbf{F}_p$. O caso de um corpo finito arbitrário $\mathbf{F}_q$ é similar. Seja m a mensagem, expressa como um número $0 \leq m < p/100$. Seja $x_j = 100m + j$, para $0 \leq j < 100$. Para $j = 0, 1, 2, \cdots, 99$, calcular $s_j = x_j^3 + Ax_j + B$.

Se $s_j^{(p-1)/2} \equiv 1\ (mod\ p)$, então s_j é um quadrado módulo p, em cujo caso não é necessário tentar mais valores de j. Se $p \equiv 3\ (mod\ 4)$, então uma raíz quadrada de s_j é dada por $y_j = s_j^{(p+1)/4}\ (mod\ 4)$. Se $p \equiv 1\ (mod\ 4)$, também é possível calcular uma raíz quadrada de s_j. Assim é obtido um ponto (x_j, y_j) na curva E. Logo, para recuperar m a partir de (x_j, y_j), basta calcular $[x_j/100]$. Dado que x_j é basicamente um elemento aleatório de $\mathbf{F}_p^*$, que é cíclico de ordem par, a probabilidade de que s_j seja um quadrado é aproximadamente $1/2$. Assim, a probabilidade de não ser possível encontrar um ponto para m, logo após de tentar 100 valores, está perto de 2^{-100}.

4 Criptografia de chave pública ElGamal

Alice deseja enviar uma mensagem para Bob. Primeiro, Bob estabelece sua chave pública como segue. Escolhe uma curva elíptica E sobre um corpo finito $\mathbf{F}_q$ de modo que o problema do logaritmo discreto seja difícil de resolver no grupo $E(\mathbf{F}_q)$. Ele também escolhe um ponto P em E (em geral é assumido que a ordem de P seja um número primo grande). Ele escolhe um inteiro secreto s e calcula $B = sP$. A curva elíptica E, o corpo finito $\mathbf{F}_q$ e os pontos P e B são a chave pública de Bob e são feitos públicos. A chave secreta de Bob é o inteiro s.

Para enviar uma mensagem a Bob, Alice segue o seguinte roteiro:

1. Baixa a chave pública de Bob.
2. Expressa sua mensagem como um ponto $M \in E(\mathbf{F}_a)$.

3. Escolhe um número inteiro secreto aleatório k e calcula $M_1 = kP$.
4. Calcula $M_2 = M + kB$.
5. Envia M_1 *e* M_2 para Bob.

Bob descriptografa a mensagem calculando $M = M_2 - sM_1$. Esta descriptografia funciona pois

$$M_2 - sM_1 = (M + kB) - s(kP) = M + k(sP) - skP = M.$$

A intrusa Eva conheçe a informação pública de Bob e os pontos M_1 *e* M_2. Se ela consegue calcular o logaritmo discreto, então pode usar P e B para encontrar s e usar esta informação para descriptografar a mensagem como $M_2 - sM_1$. Também, usando P e M_1, pode encontrar k. Assim pode calcular $M = M_2 - kB$. Se ela não puder calcular o logaritmo discreto, então não haverá maneira de encontrar M.

É importante que Alice use um k aleatório diferente cada vez que envia uma mensagem para Bob. Suponha que Alice use o mesmo k para M *e* M'. Então Eva reconhece isso pois $M_1 = M_1'$. Então calcula $M_2' - M_2 = M' - M$. Suponha que M seja um anúncio de vendas divulgado um dia depois. Então Eva descobre M e calcula $M_1 = M - M_2 + M_2'$. Portanto, neste caso, o conhecimento de um texto simples M permite que Eva deduza outro texto simples M'.

O sistema de chave pública ElGamal, em contraste com a assinatura ElGamal, esquema da próxima seção, não parece ser amplamente utilizado.

5 Assinaturas Digitais ElGamal

Alice quer assinar um documento. A maneira clássica é escrever sua assinatura em um pedaço de papel contendo o documento. Suponha, no entanto, que o documento é eletrônico, por exemplo, um arquivo de computador. A solução ingênua seria digitalizar a assinatura de Alice e anexá-la ao arquivo que contém o documento. Neste caso, a intrusa Eva pode copiar a assinatura e anexá-la ao outro documento. Portanto, devem ser tomadas medidas para vincular a assinatura ao documento de tal forma que não possa ser utilizado novamente. No entanto, deve ser possível verificar se a assinatura é válida, assim como mostrar que Alice deve ter sido a pessoa que assinou o documento. Uma solução para o problema depende da dificuldade do logaritmo discreto. Classicamente, o algoritmo foi desenvolvido para o grupo multiplicativo de um corpo finito. Na verdade, aplica-se a qualquer grupo finito. Vamos apresenta-lo para curvas elípticas.

Alice primeiro deve estabelecer uma chave pública. Ela escolhe uma curva elíptica E sobre um corpo finito $\mathbf{F}_q$ tal que o problema de logaritmo discreto seja difícil para $E(\mathbf{F}_q)$. Ela também escolhe um ponto $A \in E(\mathbf{F}_q)$. Geralmente as escolhas são feitas de modo que a ordem N de A seja um número primo grande. Alice também escolhe um inteiro secreto a e calcula $B = aA$. Finalmente, ela escolhe uma função

$$f : E(\mathbf{F}_q) \longrightarrow \mathbf{Z}.$$

Por exemplo, se $\mathbf{F}_q = \mathbf{F}_p$, então ela poderia usar $f(x, y) = x$, onde x é um número inteiro, com $0 \leq x < p$. A função f não precisa ter de propriedades especiais, exceto que sua imagem deve ser grande e apenas um pequeno número de entradas devem produzir qualquer saída (por exemplo, para $f(x, y) = x$, no máximo dois pontos (x, y) produz uma determinada saída x).

As informações públicas de Alice são E, $\mathbf{F}_q$, f, A *e* B. Ela mantém a em privado. O inteiro N não precisa ser tornado público. Seu sigilo não afeta a análise da segurança do sistema. Para assinar um documento, Alice segue os seguintes passos:

1. Representa o documento como sendo um número inteiro m (se $m > N$, escolhe uma curva maior ou usa uma função hash.
2. Escolhe um número inteiro aleatório k tal que $mdc(k, N) = 1$ e calcula $R = kA$.
3. Calcula $s = k^{-1}(m - af(R))(mod\ N)$.

A mensagem assinada é (m, R, s). Observar que m *e* s são números inteiros, enquanto R é um ponto da curva E. Observar também que Alice não está tentando manter o documento m em segredo. Se ela quiser fazer isso, então precisaria usar alguma forma de criptografia. Agora Bob verifica a assinatura da seguinte forma:

1. Baixa as informações públicas de Alice.
2. Calcula $V_1 = f(R)B + sR$ e $V_2 = mA$.
3. Se $V_1 = V_2$, ele declara a assinatura válida.

Se a assinatura é válida, então $V_1 = V_2$ pois

$$V_1 = f(R)B + sR = f(R)aA + skA = f(R)aA + (m - af(R))A = mA = V_2.$$

Aqui temos usado o fato de que $sk \equiv m - af(R)$, e portanto $sk = m - af(R) + zN$ para algum inteiro z. Assim

$$skA = (m - af(R))A + zNA = (m - af(R))A + \infty = (m - af(R))A.$$

Isto é válido pois a congruência que define s foi considerada $mod\ N$.

Se Eva puder calcular logaritmos discretos, ela poderá usar A *e* B para encontrar a. Neste caso, ela pode colocar a assinatura de Alice em qualquer mensagem. Alternativamente, Eva pode usar A *e* R para encontrar k. Como ela conheçe s, $f(R)$ *e* m, então pode usar $ks \equiv\ m - af(R)\ (mod\ N)$ para encontrar a. Se $d = mdc(f(R), N) \neq 1$, então $af(R) \equiv m - ks\ (mod\ N)$ tem d soluções para a. Desde que d seja pequeno, Eva pode tentar cada possibilidade até obter $B = aA$. Então ela pode usar a, como antes, para falsificar a assinatura de Alice em mensagens arbitrárias.

Como acabamos de ver, Alice deve manter a *e* k em segredo. Além disso, ela deve usar um k aleatório diferente para cada assinatura. Suponha que ela assine m *e* m' usando o mesmo k para obter mensagens assinadas (m, R, s) *e* (m', R, s'). Eva imediatamente reconheçe que k foi usado duas vezes, pois R é mesmo para ambas as assinaturas. As equações para s *e* s' produzem:

$$ks \equiv m - af(R)\ (mod\ N)$$
$$ks' \equiv m' - af(R)\ (mod\ N).$$

Subtraindo estas equações temos $k(s-s') \equiv m-m'(mod\ N)$. Seja $d = mdc(s-s', N)$. Existem d valores possíveis para k. Então a Eva tenta cada um até que $R = kA$ seja satisfeita. Uma vez que determinou k, então pode encontrar a, como antes.

Talvez não seja necessário que Eva resolva o problema do logaritmo discreto para falsificar a assinatura de Alice em outra mensagem m. Tudo o que Eva precisa fazer é produzir R *e* s de modo que seja satifeita a equação $V_1 = V_2$. Isso significa que ela precisa encontrar $R = (x, y)$ *e* s tal que

$$f(R)B + sR = mA.$$

Se ela escolher algum ponto R (não há necessidade de escolher um inteiro k), então precisa resolver o problema do logaritmo discreto $sR = mA - f(R)B$ para o inteiro s. Se, em vez disso, ela escolher s, então deverá resolver a equação para $R = (x, y)$. Essa equação parece ser

menos complexa do que o problema do logaritmo discreto, embora não tenha sido analisada minuciosamente. Além disso, ninguém foi capaz de descartar a possiblidade de usar algum procedimento que encontre R *e* s simultaneamente. Existem maneiras de usar uma mensagem assinada válida para produzir outra mensagem assinada válida. No entanto, é pouco provável que as mensagens produzidas sejam significativas.

Existe a ideia geral de que a segurança do sistema ElGamal está muito próxima da segurança dos logaritmos discretos para o grupo $E(\mathbf{F}_q)$.

Uma desvantagem do sistema ElGamal é que a mensagem assinada (m, R, s) é aproximadamente três vezes maior que a mensagem original (não é necessário armazenar a coordenada y de R, uma vez que existem apenas duas opções ela para um dado x). Um método mais eficiente é escolher uma função hash pública H e sinal $H(m)$. Uma função hash criptográfica é uma função que toma entradas de comprimento arbitrário, às vezes uma mensagem de bilhões de bits, e saídas de comprimento fixo, por exemplo, 160 bits. Uma função hash H deve ter as seguintes propriedades:

1. Dada uma mensagem m, o valor $H(m)$ pode ser calculado rapidamente.

2. Dado y, é computacionalmente inviável encontrar m satisfazendo $H(m) = y$ (isto significa que H tem pré-imagem resistente).

3. É computacionalmente inviável encontrar mensagens distintas m_1 *e* m_2, com $H(m_1) = H(m_2)$ (isto significa que H é fortemente livre de colisões).

A razão para (2) e (3) é evitar que Eva produza mensagens com um valor de hash desejado ou duas mensagens com o mesmo valor de hash. Isso ajuda evitar falsificações. Existem várias funções hash populares disponíveis, por exemplo, MD5 (devido ao Rivest; produz uma saída de 128 bits) e o Secure Hash Algoritmo (do NIST; produz uma saída de 160 bits). Para obter mais detalhes, consulte [6]. Trabalho recente de Wang, Yin e Yu [7] descobriu fraquezas neles, então o assunto está ainda em um estado de fluxo.

Se Alice usar uma função hash, a mensagem assinada será então

$$(m, R_H, s_H),$$

onde $(H(m), R_H, s_H)$ é a assinatura válida.

Para verificar que a assinatura (m, R_H, s_H) seja válida, Bob realiza o seguinte:

1. Abaixa a informação pública de Alice.

2. Calcula $V_1 = f(R_H)B + s_H R_H$ *e* $V_2 = H(m)A$.

3. Se $V_1 = V_2$ declara que a assinatura é válida.

A vantagem é que uma mensagem muito longa contendo bilhões de bits tem uma assinatura que requer apenas alguns milhares de bits extras. Enquanto o problema do logaritmo discreto for difícil para $E(\mathbf{F}_q)$, Eva não conseguirá colocar a assinatura de Alice numa outra mensagem. O uso de uma função hash tambám protege contra outras falsificações.

Uma variante recente do esquema de assinatura ElGamal (devido a Van Duin) é muito eficiente em certos aspectos. Por exemplo, evita o cálculo de k_1, e seu procedimento de verificação requer apenas dois cálculos de um número inteiro vezes um ponto. Como antes, Alice tem um documento que deseja assinar. Para configurar o sistema, ela escolhe uma curva elíptica E sobre um corpo finito $\mathbf{F}_q$ e um ponto $A \in E(\mathbf{F}_q)$ de ordem primo grande N. Ela também escolhe uma função hash criptografica H. Ela escolhe um inteiro secreto a e calcula $B = aA$. A informação pública é (E, q, N, H, A, B). A informação secreta é a. Para assinar m, Alice faz o seguinte:

1. Escolhe um número inteiro aleatório k módulo N e calcula $R = kA$.

2. Calcula $t = H(R, m)k + a(mod\ N)$.

O documento assinado é (m, R, t). Para verificar a assinatura, Bob abaixa a informação pública de Alice e verifica se $tA = H(R, m)R + B$ é válida. Se for assim, a assinatura é declarada válida. De outra forma é inválida.

6 O Algoritmo de assinatura digital

O Padrão de Assinatura Digital está baseado no Algoritmo de Assinatura Digital (DSA). A versão original usa grupos multiplicativos de corpos finitos. Uma versão mais recente, o ECDSA, usa curvas elípticas. O algoritmo é uma variante do esquema de assinatura ElGamal, com algumas modificações. Esboçamos o algoritmo aqui

Alice quer assinar um documento m, que é um número inteiro (na verdade, ela normalmente assina o hash do documento, como antes). Alice escolhe uma curva elíptica sobre um corpo finito $\mathbf{F}_q$ tal que $card(E(\mathbf{F}_q)) = fr$, onde r é um primo grande e f é um número inteiro pequeno, geralmente 1,2 ou 4 (f deve ser pequeno para manter o algoritmo eficiente). Ela escolhe um ponto base G em $E(\mathbf{F}_q)$ de ordem r. Finalmente, Alice escolhe um inteiro secreto a e calcula $Q = aG$. Alice faz públicas as seguintes informações:

$$\mathbf{F}_q,\ E,\ r,\ G,\ Q.$$

(não há necessidade de manter f em secreto; este pode ser deduzido de q e r usando o Teorema de Hasse).

Para assinar a mensagem m Alice segue os passos a seguir:

1. Escolhe um número inteiro aleatório k, com $1 \leq k \leq r$ e calcula $R = kG = (x, y)$.
2. Calcula $s = k^{-1}(m + ax)(mod\ r)$.

O documento assinado é (m, R, s).

Para verificar a assinatura Bob segue os passos:

1. Calcula $u_1 = s^{-1}m(mod\ r)$ e $u_2 = s^{-1}x(mod\ r)$.
2. Calcula $V = u_1G + u_2Q$.
3. Declara a assinatura válida se $V = R$.

Se a mensagem for assinada corretamente, a equação de verificação é válida:

$$V = u_1G + u_2Q = s^{-1}mG + s^{-1}xQ = s^{-1}(mG + xaG) = kG = R.$$

A principal diferença entre o sistema ECDSA e o sistema ElGamal é o procedimento de verificação. No sistema ElGamal, a equação de verificação $f(R)B + sR = mA$ requer três cálculos de um número inteiro vezes um ponto. Estas são as partes mais caras do algoritmo. Na ECDSA, apenas são necessários dois cálculos de um número inteiro vezes um ponto. Se muitas verificações serão feitas, então a maior eficiência da ECDSA é valiosa. Este é o mesmo tipo de melhoria do sistema Van Duin mencionado no final da seção anterior.

7 O esquema de Criptografia integrada de Curva Elíptica (ICIES)

O ECIES foi inventado por Bellare e Rogaway [3] e trata-se de um esquema de criptografia de chave pública.

Alice deseja enviar uma mensagem m para Bob. Primeiro Bob estabelece sua chave pública, escolhe uma curva elíptica E sobre um corpo finito $\mathbf{F}_q$ de modo que o problema do logaritmo discreto seja difícil em $E(\mathbf{F}_q)$. Logo escolhe um ponto A sobre E de ordem primo N. A seguir

escolhe um número inteiro secreto s e calcula $B = sA$. A chave pública é (q, E, NA, B). A chave privada é s.

O algoritmo também precisa de duas funções hash criptográficas, H_1 *e* H_2, e uma função de criptografia simétrica E_k (a qual depende de uma chave k) que são publicamente acordados.

Para criptografar e enviar a mensagem, Alice segue os seguintes passos:

1. Baixa a chave pública de Bob.
2. Escolhe um número inteiro aleatório k com $1 \leq k \leq N - 1$.
3. Calcula $R = kA$ e $Z = kB$.
4. Escreve a saída de $H_1(R, Z)$ como $k_1 \parallel k2$ (isto é, k_1 seguido de k_2).
5. Calcula $C = E_{k_1}(m)$ e $t = H_2(C, k_2)$.
6. Envia (R, C, t) para Bob.

Para decriptografar, Bob segue os passos:

1. Calcula $Z = sR$, usando seu conhecimento da chave secreta s.
2. Calcula $H_1(R, Z)$ e escreve a saída como $k_1 \parallel k2$.
3. Calcula $H_2(C, k_2)$. Se não for igual a t, Bob para e rejeita o texto cifrado.
4. Calcula $m = D_{k_1}(C)$, onde D_{k_1} é a função de decriptografia para E_{k_1}.

Um fato importante é o processo de autenticação dado no passo 3. ao decriptografar. Em muitos sistemas criptográficos, um invasor pode escolher vários textos cifrados e forçar Bob para descriptografá-los. Essas descriptografias são usadas para atacar o sistema. No sistema presente, o invasor pode gerar textos cifrados escolhendo C e k_2' e tomando então $t' = H_2(C, k_2')$. Mas o invasor não conheçe Z, então ele não pode usar o mesmo valor k_2 que Bob obtém de $H_1(R, Z)$. Portanto, é muito improvável que $t' = H_2(C, k_2')$ seja igual a $t = H_2(C, k_2)$. Com probabilidade muito alta, Bob simplesmente rejeita o texto cifrado e não retorna uma descriptografia.

Na descrição do processo foram usadas funções hash para a autenticação. Existem outros métodos de autenticação que poderiam ser usados.

Uma das vantagens do ICIES sobre os métodos de chave pública Masey-Omura e ElGamal é que a mensagem não é representada por um ponto da curva. Além disso, desde que um método simétrico de chaveado é usado para enviar a mensagem, não precisamos fazer um novo cálculo de curva elíptica para cada bloco da mensagem.

Orcid

Jaime Edmundo Apaza Rodriguez https://orcid.org/0000-0002-1359-9898

Referências

1. I. F. Blake, G. Seroussi and N. P. Smart, "Elliptic Curves Cryptography", volume 265 of *London Mathematical Society Lecture Notes Series.* Cambridge University Press, Cambridge, 2000. Reprint of the 1999 original.
2. L. C. Washington, "Elliptic Curves Number Theory and Cryptography", CRC Press A Chapman and Hall, Book 2008.
3. M. Abdalla, M. Bellare and P. Rogaway, "The Oracle Diffie-Hellman assumption and an analysis of DHIES", *Topics in Cryptology - CT RSA 0*, volume 2020 of *Lectures Notes in Computer Science*, Springer, Berlin, 2001, pages 143-158.

4. D. Boneh, “The decision Diffie-Hellman problem”, In *Algorithmic number theory (Portland, OR, 1998)*, volume 1423 of *Lecture Notes in Comput. Sci*, pages 48-63. Springer-Verlag, Berlin,1998.
5. N. Koblitz, “Introduction to elliptic curves and modular forms”, volume 114 of graduate texts in Mathematics. Springer-Verlag, New York, second edition, 1994.
6. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, “Handbook of applied cryptography”, CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997. With a foreword by R. L. Rivest.
7. X. Wang, Y. Yin, Yiqun, and H. Yu, “Finding collisions in the full SHA-1”, *Advances in cryptology - CRYPTO 2005*, volume 3621 of Lecture Notes in Comput. Sci. pages 17-36, Springer, Berlin, 2005.
8. A. Joux, “A one round protocol for tripartite Diffie-Hellman”, In *Algorithmic Number Theory (Leiden, The Netherlands, herefore maps2000), volume 1838 of Lecture Notes in Comput. Sci.*, pages 385-394. Springer-Verlag, Berlin, 2000.

© INTERMATHS
CC BY 4.0