Danilo Costa de Carvalho

Análise Comparativa de Parâmetros de um Servidor de Validação

Brasil

2017

Danilo Costa de Carvalho

Análise Comparativa de Parâmetros de um Servidor de Validação

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Maranhão – UFMA

Curso de Ciência da Computação

Orientador: Prof. Ms. Antônio de Abreu Batista Júnior

Brasil

2017

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

Carvalho, Danilo Costa de.
Análise Comparativa de Parâmetros de um Servidor de Validação / Danilo Costa de Carvalho. - 2017.
47 p.

Orientador(a): Antônio de Abreu Batista Júnior.
Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, Departamento de Informática, 2017.

1. Autoridade de Validação. 2. Certificado Digital. 3. Infraestrutura de chaves públicas. 4. Servidor de Validação. I. Abreu Batista Júnior, Antônio de. II. Título.

Danilo Costa de Carvalho

Análise Comparativa de Parâmetros de um Servidor de Validação

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Brasil, 26 de janeiro de 2017:

Antonio de Abreu Batista Júnior

Prof. Me. Antonio de Abreu Batista Júnior (Orientador)
Universidade Federal do Maranhão

Prof. Dr. Samyr Beliche Vale
Universidade Federal do Maranhão

Tiago Bonini Borchartt

Prof. Dr. Tiago Bonini Borchartt
Universidade Federal do Maranhão

Brasil
2017

A todos aqueles que de alguma forma estiveram e estão próximos de mim, fazendo esta vida valer cada vez mais a pena.

Agradecimentos

Agradeço a Deus e a minha família, ao meu pai Heraldo, a minha mãe Márcia e irmãs Deborah e Dandara pelo grande apoio, pela paciência e por sempre acreditarem em mim e no meu potencial. Parentes e amigos que me ajudaram dando palavras de incentivo e me reanimando em momentos difíceis também serão para sempre lembrados.

Sou muito grato também a todos os professores os quais tive aulas, sem eles não estaria elaborando esta monografia. Um agradecimento especial ao professor Antônio de Abreu, graças a ele este trabalho foi idealizado e concretizado através de sua orientação. Agradeço também aos professores Samyr Béliche e Tiago Bonini por aceitarem fazer parte da banca.

Grandes amigos, companheiros e colegas curso, como Thales Levi, Tiago Ramos, Wendell Luís, Jakelson Mendes, Johnatan Carvalho, Glécio Santos, Pedro Paulo e tantos outros que compartilharam comigo momentos de dificuldades e alegrias, para todos o meu grande agradecimento por propiciarem grandes momentos dentro do curso.

"Porque a sabedoria serve de defesa, como de defesa serve o dinheiro; mas a excelência do conhecimento é que a sabedoria dá vida ao seu possuidor."

(Bíblia Sagrada, Eclesiastes 9:10)

Resumo

Para garantir que um certificado digital não seja utilizado indevidamente, cada vez que um certificado é utilizado para autenticar um usuário, faz-se necessária uma verificação completa da sua validade. Uma Autoridade de Validação fornece a validação de certificados digitais de forma centralizada para todos os seus usuários, garantindo que certificados inválidos ou revogados não sejam utilizados. Este componente da Infraestrutura de chaves públicas é implementado por um servidor denominado Servidor de Validação. Este trabalho teve como objetivo a realização de uma análise comparativa de parâmetros de um servidor de Validação, consequentemente obtendo as circunstâncias em que a Autoridade de Validação terá um melhor desempenho. Para isso, foram considerados os seguintes fatores ou parâmetros: o sistema de coordenadas utilizado na implementação do algoritmo de assinatura digital, a curva elíptica escolhida e a carga de trabalho do sistema. Foram executadas todas as combinações possíveis de fatores e níveis de fatores. Ao fim de cada execução, foram realizadas aferições para obtenção dos valores médios das variáveis de saída. Ao final de todo o processo, verificou-se um melhor desempenho nos servidores que utilizam coordenadas do tipo Jacobianas.

Palavras-chave: Autoridade de Validação. Certificado Digital. Servidor de Validação. Infraestrutura de Chaves Públicas.

Abstract

To ensure that a digital certicate is not misused, each time a certicate is used to authenticate a user, a complete verication of its validity is required. A Validation Authority provides centralized digital certicate validation for all its users, ensuring that invalid or revoked certicates are not used.This component of Public Key Infrastructure is implemented by a server called Validation Server. To ensure that a digital certificate is not misused, each time a certificate is used to authenticate a user, a complete check of its validity is required. A Validation Authority provides centralized digital certificate validation for all its users, ensuring that invalid or revoked certificates are not used. This component of Public Key Infrastructure is implemented by a server called Validation Server. The purpose of this work was to perform a comparative analysis of parameters of a Validation server, thus obtaining the circumstances in which the Validation Authority will perform better. For this, the following factors or parameters were considered: the coordinate system used in the implementation of the digital signature algorithm, the chosen elliptic curve and the system workload. All possible combinations of factors and factor levels were performed. At the end of each run, measurements were taken to obtain the mean values of the output variables. At the end of the whole process, it was verified a better performance in the servers that use Jacobian type coordinates.

Keywords: Validation Authority. Digital Certificate. Validation Server. Public Key Infrastructure.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CA	Certificate Authority
RG	Registro Geral
CPF	Cadastro de Pessoa Física
DH	Diffie-Hellman
DSA	Digital Signature Algorithm
ECADD	Elliptic Curve Addition
ECC	Elliptic Curves Criptography
ECDBL	Elliptic Curve Doubling
ECDSA	Elliptic Curves Digital Signature Algorithm
Gen	Generate
IEEE	Institute of Electrical and Eletronic Engineers
ISO	International Standard and Technology
NIST	National Institute of Standard and Technology
PKC	Public Key Certificate
PKI	Public Key Infrastructure X.509
RA	Register Authority
RSA	Rivest-Shamir-Adleman
Sign	Signature
VA	Validation Authority
Vrfy	Verify

Sumário

1 Introdução

Os computadores e a Internet são amplamente utilizados para o processamento de dados, para a troca de mensagens e documentos entre pessoas, governos e empresas. O comércio eletrônico (e-commerce), o acesso a bancos online (Internet Banking) e os mensageiros online são exemplos de aplicações onde a segurança na troca de informações é de fundamental importância, pois é um requisito obrigatório devido à exigência de sigilo sobre as informações. Essas transações eletrônicas necessitam de mecanismos de segurança, que são garantidos por uma estrutura baseada no princípio da terceira parte confiável, denominada Infraestrutura de Chave Pública.

Hoje esse tipo de infraestrutura utiliza mecanismos de segurança baseados na criptografia de chaves públicas para promover a autenticação, a confidencialidade, a integridade e o não repúdio de informações. A infraestrutura de chave pública tem como objetivo a emissão de chaves públicas, intermediando uma relação de confiança e credibilidade entre entidades em transações. Um problema central com o uso da criptografia de chave pública é a confiança ou prova que uma chave pública específica é autêntica, ou seja, se ela é correta e pertence a uma determinada pessoa ou entidade, evitando a adulteração ou substituição da chave por um terceiro malicioso. Dentro dessa estrutura é necessário que também um terceiro certifique a propriedade das chaves públicas, denominado Autoridade Certificadora.

A Autoridade Certificadora faz o papel do terceiro confiável, o qual é o principal componente de uma Infraestrutura de Chaves Públicas e é responsável pelo fornecimento dos certificados digitais. A escolha de confiar em uma Autoridade Certificadora é parecida ao que ocorre em transações tradicionais, que não se utilizam do meio eletrônico, seja computadores ou dispositivos móveis. Por exemplo, uma empresa que vende parcelado aceita alguns documentos para identificar o cliente antes de efetivar a transação. Estes documentos habitualmente são emitidos pela Secretaria de Segurança de Pública e pela Secretaria da Receita Federal, como a Carteira de Identidade(RG) e o Cadastro de Pessoa Física(CPF). Nesta situação, há uma relação de confiança já estabelecida com esses órgãos.

Os certificados digitais são assinados por uma Autoridade Certificadora por meio de sua assinatura digital. A técnica de assinatura digital é uma forma eficaz de garantir a autoria das mensagens em uma comunicação eletrônica, sendo tipicamente tratada como substituta à assinatura manuscrita. Segundo Katz e Lindell (2014), esta forma de assinatura consiste em o emissor encriptar o hash(resumo) de uma mensagem utilizando a sua chave privada e, além disso, também encriptar com a chave pública do receptor a mensagem pura. Depois de receber este par do emissor, o receptor descriptografa o texto pleno

usando sua chave privada e também faz o mesmo para obter o resumo da mensagem, mas utilizando a chave pública do emissor. No final do procedimento, o receptor aplica a função hash sobre o texto pleno e o compara ao resumo também recebido, se forem idênticos, então a assinatura confere. O processo de conferir a assinatura da autoridade certificadora que emitiu o certificado é feito por uma Autoridade de Validação. Este componente da Infraestrutura de chaves públicas é implementado por um servidor denominado Servidor de Validação. Um usuário da infraestrutura que deseja validar um certificado de um de seus clientes envia este certificado para este servidor que o valida ou não.

A motivação para a realização desse trabalho seria o aumento de demanda em servidores de validação ao longo tempo, ocasionado por um aumento no tamanho da infraestrutura. O desafio seria testar novas possibilidades com o intuito de melhorar o desempenho desses servidores que atuam como autoridades de validação, sem que haja o aumento de custo em relação ao hardware. O objetivo geral é realizar uma análise comparativa de parâmetros de um servidor de validação, ou seja, testar diversas combinações de parâmetros no algoritmo de verificação de assinaturas utilizado pelo servidor de validação. Os objetivos específicos seriam: estudar o método de assinatura digital ECDSA, definir as variáveis que possivelmente afetam o Servidor de Validação, desenvolver um estudo de caso; criar os experimentos e analisar os resultados.

A metodologia da análise comparativa segue as técnicas de avaliação de desempenho de sistemas computacionais propostas por Jain (1991). Segundo ele, os passos necessários para se realizar uma avaliação de desempenho sistemática são:

1. **Definir os objetivos e o sistema:** O primeiro passo é definir o objetivo do estudo, o sistema e suas fronteiras

2. **Listar serviços e saídas:** nem todas as saídas do sistema escolhido são relevantes ao estudo, é necessário definir o que será monitorado

3. **Selecionar métricas:** definir quais os critérios de comparação serão utilizados

4. **Definir parâmetros:** Listar quais parâmetros afetam a performance. Dentre estes deve-se definir quais serão manipulados no estudo e quais serão constantes.

5. **Selecionar a técnica de avaliação:** Jain define três técnicas de avaliação: modelagem analítica, simulação e aferição em sistemas reais. É necessário definir qual a técnica mais adequada para o resultado desejado.

6. **Selecionar a carga de trabalho:** A carga de trabalho consiste nas entradas que o sistema vai receber, que deve ser sempre o mais proximo do funcionamento real do sistema para maior precisão.

7. **Projeto de experimentos:** definir quais experimentos serão realizados para obter o maior número de informações desejadas no sistema.
8. **Analisar e interpretar dados:** Após os experimentos é necessário avaliar os resultados obtidos. Um conhecimento profundo do sistema analisado é importante nesta etapa pois pode ajudar a detectar discrepâncias nos resultados.
9. **Apresentar resultados:** A etapa final consiste em apresentar os resultados obtidos após a avaliação de maneira que seja facilmente compreendida.

Jain (1991) define alguns termos específicos utilizados nas etapas de criação e análise dos experimentos, estas são relevantes para este trabalho e são definidas a seguir:

1. **Variável de Resposta:** A saída ou saídas do sistema que serão utilizadas como métrica da avaliação de desempenho
2. **Fatores:** Variáveis que serão manipuladas durante os experimentos e que afetam a variável de resposta
3. **Níveis:** Os valores que serão atribuídos aos fatores durante os experimentos

Outro ponto relevante que Jain (1991) define são os 3 tipos de projeto mais utilizados:

1. **Fatorial simples:** Onde varia-se somente um fator por vez e analisa-se a influência dele nas variaveis de resposta
2. **Fatorial completo:** Onde todas as combinações possíveis de fatores e níveis são testadas, tem como vantagem apresentar um retrato mais amplo do sistema avaliado.
3. **Fatorial parcial:** Para experimentos muito grandes, trabalha-se somente com uma parte das possíveis combinações de níveis e fatores.

2 Referencial Teórico

Nas seções seguintes serão apresentados os conceitos de assinatura digital e criptografia de curvas elípticas, que são mecanismos que permitem o pleno funcionamento do sistema de segurança de um Servidor de Validação; como são obtidos e verificados os certificados digitais, como funciona e qual o papel da validação dentro do contexto da infraestrutura de chaves públicas, além da explicação do tipo de infraestrutura PKIX.

2.1 Assinaturas Digitais

Os esquemas de assinatura admitem que um assinante S, que determinou uma chave pública pk "assine" uma mensagem usando a chave privada associada sk, de tal forma que qualquer pessoa que conheça pk possa verificar que a mensagem é oriunda de S e que não foi modificada durante o percurso. Vale ressaltar que qualquer pessoa de posse de pk deve saber que esta chave pública foi de fato estabelecida por S.

Um bom exemplo, segundo Katz e Lindell (2014), é a atualização de diversos tipos de programas. Suponha que uma empresa de software deseja propagar atualizações de software de forma autenticada; ou seja, quando uma empresa lança uma atualização que possibilita para qualquer um dos seus clientes verificar se esta é autêntica, um terceiro malicioso jamais deve ser capaz de enganar um cliente para que este aceite uma atualização que não foi realmente lançada pela empresa. Para fazer isso, basta a empresa gerar uma chave pública pk em conjunto com uma chave privada sk e em seguida distribuir pk seguramente para seus clientes (pode ser através do empacotamento da chave com o software original adquirido). A empresa de software, ao liberar a atualização m, calcula uma assinatura digital σ sobre m usando a sua chave privada sk e envia (m, σ) a cada cliente, que pode logo em seguida verificar a autenticidade da atualização m relacionando a assinatura com a chave pública pk.

Um terceiro malicioso poderia tentar enviar uma atualização falsa (m', σ') para um cliente do software, mas quando o cliente tentasse verificar σ', acharia que esta é uma assinatura inválida em m' com respeito a chave pública pk e iria, portanto, rejeitar a assinatura, mesmo que m' seja apenas uma ligeira modificação de uma atualização genuína m. Este tipo de aplicação teórica de assinaturas digitais é usado hoje largamente na distribuição de atualizações de software.

2.1.1 Definição de Assinatura Digital

Segundo Katz e Lindell (2014), um esquema de assinatura digital *consiste em três algoritmos probabilísticos de tempo polinomial* (Gen, Sign, Vrfy) *tais que:*

1 - O algoritmo de geração de chaves Gen tem como entrada um parâmetro de segurança 1^n e emite um par de chaves (pk, sk). Estes são chamados de chave pública e a chave privada, respectivamente. Assumimos que pk e sk, cada um tem comprimento tem pelo menos n e que n pode ser determinada a partir de pk ou sk.

2 - O algoritmo de assinatura Sign tem como entrada uma chave privada sk e uma mensagem m. Ele produz uma assinatura σ, e escreve-se isso como $\sigma \leftarrow Sign_{sk}(m)$.

3 - O algoritmo de verificação deterministico Vrfy tem como entrada uma chave pública pk, uma mensagem m, e uma assinatura σ. Emite um bit b, com $b = 1$ que significa válido e $b = 0$ que significa inválido. Isto é escrito como b := $Vrfy_{pk}$ (m, σ).

2.2 Criptografia de Curvas Elípticas (ECC)

O uso de curvas elípticas foi inicialmente proposto por Miller (1986) e N.Koblitz (1987) no ano de 1985. Esta abordagem de criptografia tem aceitação ampla como alternativa aos criptossitemas tradicionais, como RSA, DSA e DH, pois como diz Katz e Lindell (2014), a criptografia de curvas elípticas oferece o mesmo nível de segurança utilizando tamanhos de chave muito menores, tornando-a muito mais atrativa. Por exemplo, uma chave de 160 bits na ECC é equivalente em nível de segurança à chave de 1024 bits no RSA, DSA e DH, porém há uma desvantagem na implementação ECC por esta ser bem mais complexa.

Figueiredo (2010) diz que a segurança de sistemas criptográficos de chave pública como Diffie-Hellman, ElGamal e DSA está na dificuldade do problema do logaritmo discreto, isto é, dados o grupo cíclico G e um gerador g deste grupo, como calcular x através de $h = g^x$. De forma semelhante, o problema do logaritmo discreto para curvas elípticas é o seguinte: dados os pontos P e $Q = k \cdot P$ em uma curva elíptica sobre um corpo finito, determinar o valor de k.

2.2.1 Corpos Finitos

Um corpo finito é um conjunto finito de elementos no qual pode-se somar, subtrair, multiplicar e dividir por não nulo. Neste corpo valem todas as propriedades usuais de tais operações, incluindo o elemento simétrico para a soma, o elemento inverso para a multiplicação; a comutatividade, associatividade e elemento neutro para ambas as operações (WASHINGTON, 2008). Além disso, precisa ser válida a propriedade da distributividade da multiplicação em relação à soma $x \cdot (y + z) = x \cdot y + x \cdot z$. O conjunto dos racionais

$\mathbb{Q}$, dos reais $\mathbb{R}$, dos complexos $\mathbb{C}$ e o conjunto dos inteiros vistos módulo p primo $\mathbb{Z}_p$ são exemplos de corpos.

Um corpo é munido de duas operações, adição e multiplicação, entretanto, a subtração de elementos do corpo é definida em termos da adição: para $a, b \in \mathbb{Z}_p$, $a - b = a + (-b)$ onde $-b$ é o único elemento em um corpo tal que $b + (-b) = 0$ ($-b$ é chamado de negativo de b). De forma parecida, a divisão de elementos do corpo é definida em termos da multiplicação: para $a, b \in \mathbb{Z}_p$ com $b \neq 0$, $a/b = a \cdot b^{-1}$ onde é o único elemento em um corpo tal que $b \cdot b^{-1} = 1$. Então b^{-1} é chamado de inverso de b.

Seja p um número primo. Os inteiros módulo p, constituídos dos inteiros $\{0, 1, 2, ..., p-1\}$ com adição e multiplicação executadas módulo p, é um corpo finito de ordem p. Denota-se este corpo por $\mathbb{Z}_p$ e chama-se p de módulo de $\mathbb{Z}_p$. Para qualquer inteiro a, a mod p denotará o único resto inteiro r, $0 \leq r \leq p - 1$, obtido pela divisão de a por p; essa operação é chamada redução módulo p.

Por exemplo, os elementos de $\mathbb{Z}_{29}$ são $\{0, 1, 2, ..., 28\}$. A seguir estão alguns exemplos de operações aritméticas em $\mathbb{Z}_{29}$:

(i) Adição: $14 + 19 = 4$, pois 33 mod 29 = 4. (ii) Subtração: $15 \breve{} 20 = 24$, pois -5 mod 29 = 24.

(iii) Multiplicação: $17 \cdot 20 = 21$, pois 340 mod 29 = 21.

(iv) Inversão: $17^{-1} = 12$, pois $71 \cdot 12$ mod 29 = 1.

2.2.2 Curvas Elípticas

Em (WASHINGTON, 2008), uma curva elíptica E, definida sobre um corpo finito $\mathbb{Z}_p$, é uma curva plana determinada por uma equação na forma:

$$y^2 \equiv x^3 + a \cdot x + b \pmod{p},$$

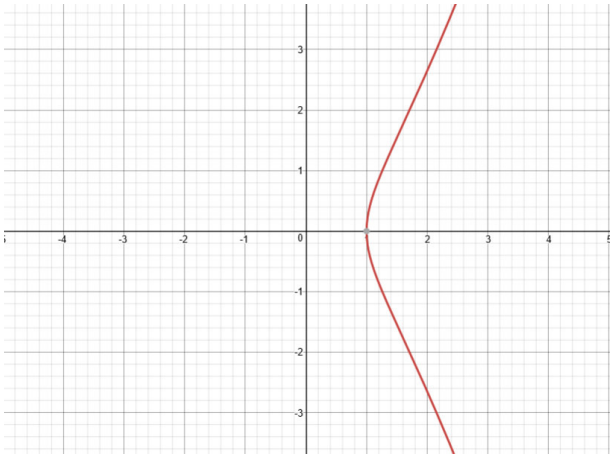
onde $a, b \in \mathbb{Z}_p$ são constantes as quais $4 \cdot a^3 + 27 \cdot b^2 \neq 0$ mod p. Seja $E(\mathbb{Z}_p)$ o conjunto de pares $(x, y) \in \mathbb{Z}_p$ x $\mathbb{Z}_p$, que satisfaçam a equação da curva plana juntamente com um valor especial O, então:

$$E(\mathbb{Z}_p) := \{(x, y) | x, y \in \mathbb{Z}_p \text{ e } y^2 \equiv x^3 + a \cdot x + b \pmod{p} \} \bigcup \{O\}.$$

Os elementos de $E(\mathbb{Z}_p)$ são denominados pontos na curva elíptica E e O é chamado de ponto no infinito. Uma curva elíptica é dita não singular, isto é, seu gráfico não possui auto-interseção e nem as chamadas cúspides, que são pontos no gráfico da curva onde não há suavidade, mas sim quinas ou ângulos salientes.

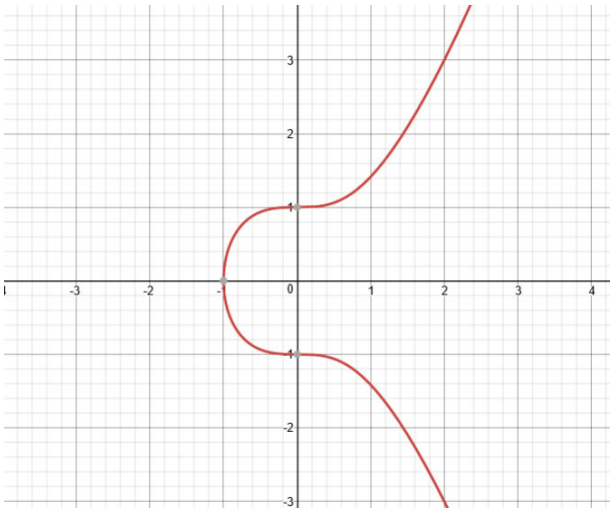
As figuras a seguir mostram os gráficos de algumas curvas elípticas. Note na Figura 3 que o gráfico de uma curva elíptica pode ter um ou dois "pedaços".

Figura 1 – Gráfico da curva $y^2 = x^3 - 1$



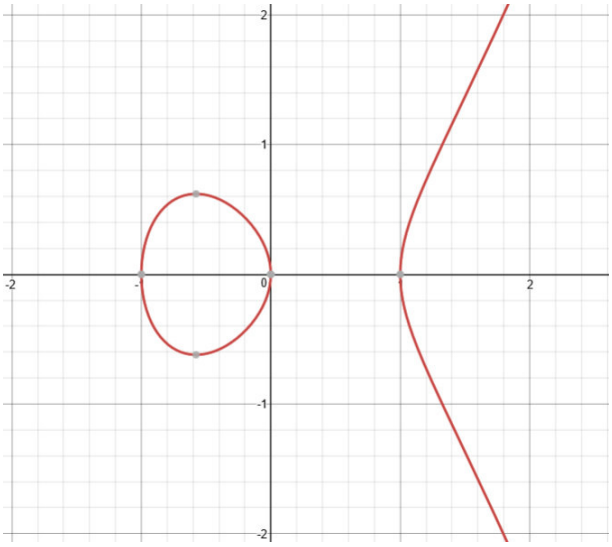
Fonte: Produzido pelo autor

Figura 2 – Gráfico da curva $y^2 = x^3 + 1$



Fonte: Produzido pelo autor

Figura 3 – Gráfico da curva $y^2 = x^3 - x$

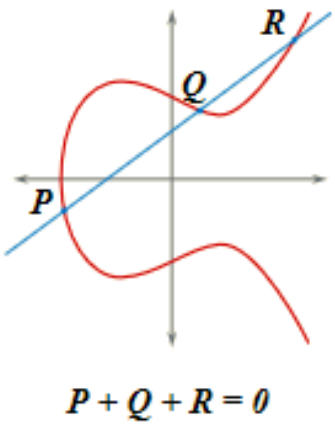


Fonte: Produzido pelo autor

Como já foi dito, o conjunto $E(\mathbb{Z}_p)$ consiste em todos os pontos $(x, y) \mid x, y \in \mathbb{Z}_p$ que satisfazem a equação da curva elíptica, juntamente com o ponto no infinito O. Contudo, existem operações feitas sobre esses pontos, que são as operações de soma de dois pontos e a multiplicação por escalar, onde um algoritmo computa um ponto resultante pertencente a $E(\mathbb{Z}_p)$. O conjunto de pontos $E(\mathbb{Z}_p)$ mais a operação de soma formam um grupo abeliano, onde o ponto no infinito O é o elemento neutro.

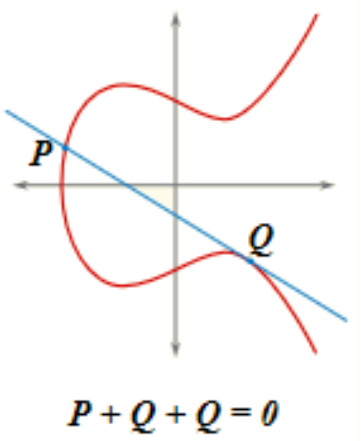
Graficamente, dados dois pontos P e Q em uma curva elíptica, é identificado de maneira única um ponto R, que é um terceiro ponto de interseção da reta que passar por P e Q na curva, como é mostrado na Figura 4. Contudo, como é ilustrado na Figura 5, caso a reta seja tangente à curva em algum dos pontos, este ponto na tangente será considerado o terceiro ponto de interseção R.

Figura 4 – R é o terceiro ponto de interseção na curva



Fonte: <https://pt.wikipedia.org/wiki/Curva_elíptica>

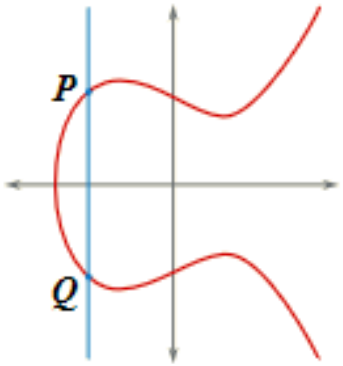
Figura 5 – O próprio Q é o terceiro ponto de interseção



Fonte: <https://pt.wikipedia.org/wiki/Curva_elíptica>

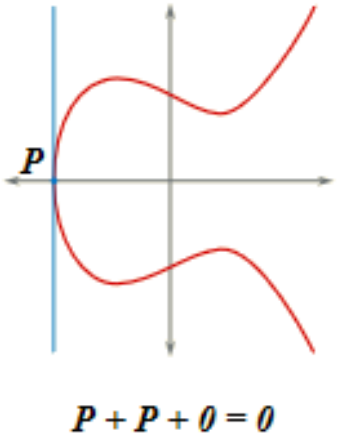
A Figura 6 mostra que, caso a reta seja vertical, então é definido o terceiro ponto de interseção como o ponto no infinito O. Assim, toda reta vertical paralela ao eixo y passa pelo ponto no infinito. A Figura 7 mostra o segundo exemplo em que o terceiro ponto de interseção é o ponto no infinito.

Figura 6 – Pontos P, Q e ponto no infinito O



Fonte: <https://pt.wikipedia.org/wiki/Curva_elíptica>

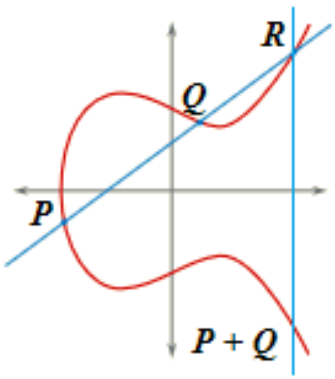
Figura 7 – Um ponto tangente P mais o ponto no infinito O



Fonte: <https://pt.wikipedia.org/wiki/Curva_elíptica>

No caso da Figura 4, por exemplo, é definido uma operação de soma nos pontos da curva da seguinte forma: $P + Q + R = O$, onde O é o elemento neutro da soma. O terceiro ponto de interseção na curva é o ponto $R = -(P + Q)$, depois é traçada uma reta vertical que passa por R. O ponto de encontro dessa reta vertical com a curva é a adição entre os pontos P e Q (Figura 8).

Figura 8 – Operação de adição $P + Q$



Fonte: Produzido pelo autor

2.2.3 Sistemas de Coordenadas

2.2.3.1 Sistema de Coodernadas Afim

Seja uma curva elíptica $y^2 = x^3 + a \cdot x + b$ sobre um corpo primo, um ponto dessa curva pode ser representado na forma (x, y). As duas operações aritméticas básicas em curvas elípticas são a adição em curvas elípticas (ECADD) e a duplicação em curvas elípticas (ECDBL), através destas duas operações é possível realizar a soma de dois pontos e a multiplicação de um ponto por um escalar. Seja $P = (x_1, y_1)$ e $Q = (x_2, y_2)$, a adição de pontos será feita enquanto $P \neq Q$ e se $P = Q$, então a operação de duplicação de um

ponto será realizada. A Tabela 1 mostra as computações necessárias para a realização da ECADD e da ECDBL.

Tabela 1 – Operações de soma e duplicação de pontos afins em curvas elípticas

Operação	Fórmula (Sistema de Coordenada Afim)
ECADD	$x_3 = \lambda^2 - x_1 - x_2 \bmod p$, $y_3 = \lambda \cdot (x_1 - x_3) - y_1 \bmod p$ e $\lambda \equiv \frac{(y_2 - y_1)}{(x_2 - x_1)} \bmod p$
ECDBL	$x_3 = \lambda^2 - 2 \cdot x_1 \bmod p$, $y_3 = \lambda \cdot (x_1 - x_3) - y_1 \bmod p$ e $\lambda \equiv \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} \bmod p$

Fonte: Produzido pelo autor

Através das operações ECADD e ECDBL é possível a realização da operação de multiplicação por escalar, que é a computação da forma $Q = k \cdot P$, onde P e Q são pontos da curva elíptica e k é um inteiro. É necessário que o inteiro k esteja representado na forma binária: o custo dessa multiplicação depende do tamanho de k, além da quantidade de 1s nesta representação. A Tabela 2 ilustra o algoritmo de multiplicação de um ponto por um escalar através do método binário, versão "esquerda para a direita".

Tabela 2 – Algoritmo de multiplicação de um ponto por um escalar

Multiplicação por escalar: Método Binário

Entrada: representação binária de k e o ponto P
Saída: $Q = k \cdot P$
$Q = P$
for $i = n - 1$ to 0 do
  $Q = 2 \cdot Q$ (Duplicação)
  if $k_i = 1$ then
  $Q = Q + P$ (Adição)
Return Q

Fonte: Produzido pelo autor

2.2.3.2 Sistema de Coodernadas Jacobianas

Coordenadas Jacobianas são usadas para representar pontos de curvas elípticas em curvas $y^2 = x^3 + a \cdot x + b$ sobre corpos primos. Segundo (WASHINGTON, 2008), nas coordenadas jacobianas o triplo (X, Y, Z) representa o ponto afim $(X/Z^2, Y/Z^3)$.

Seja (X, Y, Z) um ponto (diferente do ponto no infinito) representado em coordenadas jacobianas. Então a duplicação (X', Y', Z') pode ser calculada conforme o algoritmo mostrado na Tabela 3.

Tabela 3 – Algoritmo de duplicação de um ponto no sistema de coordenadas Jacobianas

Duplicação: Coordenada Jacobiana
if $(Y == 0)$
return $PontoNoInfinito$
$S = 4 \cdot X \cdot Y^2$
$M = 3 \cdot X^2 + a \cdot Z^4$
$X' = M^2 - 2 \cdot S$
$Y' = M \cdot (S - X') - 8 \cdot Y^4$
$Z' = 2 \cdot Y \cdot Z$
return (X', Y', Z')

Fonte: Produzido pelo autor

Seja (X_1, Y_1, Z_1) e (X_2, Y_2, Z_2) dois pontos (ambos diferentes do ponto no infinito) representados em coordenadas jacobianas. Então a adição (X_3, Y_3, Z_3) pode ser calculada conforme o algoritmo apresentado na Tabela 4.

Tabela 4 – Algoritmo de adição entre dois pontos no sistema de coordenadas Jacobianas

Adição: Coordenada Jacobiana
$U1 = X_1 \cdot Z_2^2$
$U2 = X_2 \cdot Z_1^2$
$S1 = Y_1 \cdot Z_2^3$
$S2 = Y_2 \cdot Z_1^3$
if $(U1 == U2)$
if $(S1! = S2)$
return $PontoNoInfinito$
else
return Duplicação(X_1, Y_1, Z_1)
$H = U2 - U1$
$R = S2 - S1$
$X_3 = R^2 - H^3 - 2 \cdot U1 \cdot H^2$
$Y_3 = R \cdot (U1 \cdot H^2 - X_3) - S1 \cdot H^3$
$Z_3 = H \cdot Z_1 \cdot Z_2$
return (X_3, Y_3, Z_3)

Fonte: Produzido pelo autor

A multiplicação de um ponto por um escalar nas coordenadas Jacobianas ocorre utilizando o mesmo algoritmo descrito na Tabela 2, fazendo uso das operações de duplicação e soma de pontos em coordenadas Jacobianas.

2.2.4 ECDSA

O algoritmo de assinatura digital com curvas elípticas (ECDSA – Elliptic Curves Digital Signature Algorithm) é equivalente ao algoritmo de assinatura ElGamal, porém utilizando criptografia de curva elíptica (WASHINGTON, 2008). Ele foi proposto inicialmente

por Scott Vanstone em 1992 para o NIST (National Institute of Standard and Technology). Ele foi aceito em 1998 pelo ISO (International Standards Organization) [ISO 14888-3], também aceito em 1999 pela ANSI (American National Standards Institute) [ANSI X69.2] e em 2000 pelo IEEE (Institute of Electrical and Eletronic Engineers) [IEEE P1363]. Como o ECDSA foi aceito por tantas instituições de padronização, podemos assumir que ele está estável e é bastante robusto. O algoritmo de assinatura é composto por três etapas: A geração das chaves, a assinatura e a checagem da assinatura.

Suponha uma entidade Alice querendo enviar uma mensagem assinada para Bob. Primeiramente, ambos devem concordar com os parâmetros da curva ($CURVE$, G, n), onde $CURVE$ é a equação usada e o corpo da curva elíptica, G é o ponto gerador do grupo finito $E(\mathbb{Z}_p)$ e n, que é a ordem de G e inteiro tal que $n \cdot G = O$.

No processo de geração das chaves, Alice cria um par de chaves, uma consistindo de uma chave privada inteira sk_a selecionada aleatoriamente dentro do intervalo $[1, n-1]$, a outra de um ponto da curva Q_a, dada pela multiplicação de um ponto da curva elíptica por um escalar $Q_a = sk_a \cdot G$. Na etapa seguinte, para Alice assinar uma mensagem m, ela deve seguir os seguintes passos:

1 – Calcular $z = HASH(m)$, onde $HASH$ é uma função criptográfica de hash ou resumo da mensagem, o $SHA-256$ é um exemplo.

2 – Selecionar aleatoriamente um inteiro k do intervalo $[1, n-1]$.

3 – Calcular o ponto da curva $(x_1, y_1) = k \cdot G$.

4 – Calcular $r = x_1 \pmod{n}$. Se $r = 0$, então voltar ao passo 2.

5 – A string z resultante do $HASH(m)$ deve ser convertida para um inteiro, então calcular $s = k^{-1} \cdot (z + r \cdot sk_a) \pmod{n}$. Se $s = 0$, então voltar ao passo 2.

7 – A assinatura de m é o par (r, s).

No processo de verificação, para Bob autenticar a assinatura de Alice, ele deve ter uma cópia da chave pública de Alice Q_a. Bob pode verificar se Q_a é um ponto válido da curva da seguinte maneira: verificando se Q_a não é igual ao ponto no infinito O, se Q_a está na curva e se $n \cdot Q_a = O$. Depois disso, Bob segue os seguintes passos:

1 – Verificar se r e s são inteiros do intervalo $[1, n-1]$. Se não, a assinatura é inválida.

2 – Calcular $e = HASH(m)$, onde $HASH$ é a mesma função de resumo usada anteriormente na geração da assinatura.

3 – Calcular $w = s^{-1} \pmod{n}$.

4 – Seja z os bits mais à esquerda de e, a string z é convertida em inteiro, em seguida calcular $u_1 = z \cdot w \pmod{n}$ e $u_2 = r \cdot w \pmod{n}$.

5 – Calcular o ponto da curva $(x_2, y_2) = u_1 \cdot G + u_2 \cdot Q_a$.

6 – A assinatura é válida se $r = x_2 \pmod{n}$.

2.3 Certificado Digital

Certificado digital é simplesmente uma assinatura que vincula uma entidade a alguma chave pública. Por exemplo, uma entidade Eva produziu um par de chaves (pk_E, sk_E) para um esquema de assinatura digital seguro. Suponha que outra entidade Bob também produziu um par de chaves (pk_B, sk_B) e que Eva sabe que pk_B é a chave pública de Bob, então Eva pode realizar a seguinte assinatura $cert_{E \to B} = \text{Sign } sk_E$ ('A chave de Bob é pk_B') e envia esta a Bob. Chama-se $cert_{E \to B}$ um certificado emitido por Eva para a chave de Bob. Um certificado deveria, na prática, identificar o detentor de uma determinada chave pública e, portanto, uma descrição mais completa do que "Bob" seria usada, como o nome completo de Bob, endereço de e-mail, etc.

Seguindo com a sequência dos acontecimentos, Bob quer se comunicar com alguma outra entidade Alice que já conhece pk_E. Bob pode enviar (pk_B, $cert_{E \to B}$) para Alice, que pode verificar que $cert_{E \to B}$ é de fato uma assinatura válida na mensagem "A chave de Bob é pk_B" em relação ao pk_E. Admitindo que a verificação tenha êxito, Alice agora sabe que Eva assinou a mensagem indicada. Se Alice confia em Eva, ela aceita pk_B como a chave público autêntica de Bob.

Bob e Alice podem se comunicar em um canal não autenticado e inseguro, caso um atacante queira interferir na transmissão de (pk_B, $cert_{E \to B}$) de Bob para Alice, esse terceiro malicioso não será capaz de produzir um certificado válido conectando Bob a qualquer outra chave púlica pk'_B, a menos que Eva tivesse assinado previamente algum certificado ligando Bob à chave pk'_B, o que seria muito improvável.

2.4 Infraestrutura de Chave Pública

Uma das grandes preocupações dentro da criptografia de chave pública é a distribuição segura de chaves públicas. Através da própria criptografia de chave pública e dos esquemas de assinatura digital, pode-se montar uma estrutura baseada em certificados e centrada na figura da Autoridade Certificadora, que é responsável por fazer a mediação de credibilidade e confiança entre duas ou mais entidades que queiram se comunicar.

Uma infraestrutura de chaves públicas (PKI) é definida por detalhes como: de que forma uma determinada entidade adquiriu a chave pública de uma autoridade certificadora; como uma entidade pode ter certeza da autenticidade de uma chave pública pertencente a outra entidade; como se dá a confiança de uma entidade com uma autoridade certificadora, entre outros. Existem diversos modelos diferentes de PKI sugeridos que permitem a

distribuição generalizada de chaves públicas, por exemplo, os que possuem uma única autoridade de certificação, os que possuem várias autoridades ou mesmo um modelo onde há a delegação em uma cadeia de certificados.

Como exemplo mais simples, a PKI de uma única Autoridade certificadora (CA) é uma infraestrutura em que a CA é completamente confiável por todos e produz certificados para a chave pública de todos. Uma autoridade de certificação seria mais provavelmente uma empresa cujo negócio é certificar chaves públicas, ou um órgão do governo ou um departamento dentro de uma empresa privada (neste caso, apenas para uso interno). Qualquer pessoa que queira confiar na CA teria quer obter uma cópia legítima da chave pública da CA, mas esta etapa deve ser realizada seguramente, pois se alguma entidade obtém uma chave pública incorreta, esta parte não será capaz de obter uma chave autêntica de outra pessoa.

Os modelos em que há várias autoridades certificadoras dão maior liberdade de escolha às partes que se comunicam, podendo uma CA ter o seu processo de verificação de identidade e proteção de chave privada questionados por alguma entidade. Já o modelo de delegação é outra abordagem que alivia parte da carga sobre uma única CA, já que usa cadeias de certificados. Por exemplo, suponha que Eva atua como CA, gera um certificado para Bob e que a chave pk_B é uma chave pública para um esquema de assinatura; Bob pode produzir um certificado para Alice do formulário $cert_{B\to A}$ = Sign sk_B ('A chave de Alice é pk_A'). Agora, se Alice quiser se comunicar com outra entidade que conhece a chave pública de Eva, mas que não conhece a de Bob, então Alice envia (pk_A, $cert_{B\to A}$, pk_B, $cert_{C\to B}$) para esta entidade. Esta quarta entidade pode primeiro verificar Eva, em quem confia, para em seguida, já de posse da chave pk_B, verificar o certificado $cert_{B\to A}$ emitido por Bob, assim obtendo a chave pública autêntica de Alice. A importância da delegação é: quando Eva assina um certificado para Bob, Eva está na verdade delegando sua capacidade de gerar certificados para Bob.

2.4.1 Componentes da PKIX

A PKIX é uma infraestrutura de chaves públicas que usa o padrão X.509 (SLAGELL; BONILLA; YURCIK, 2006). Este especifica um formato de certificado e procedimentos para distribuição de chaves públicas via certificados de chave pública(PKCs) assinados por Autoridades certificadoras (CAs).

A arquitetura de uma PKIX consiste em cinco componentes que são ilustrados na Figura 9. A seguir é é descrito a função de cada componente e os seus relacionamentos:

(i) Autoridades Certificadora (CA): emitem e revogam PKCs.

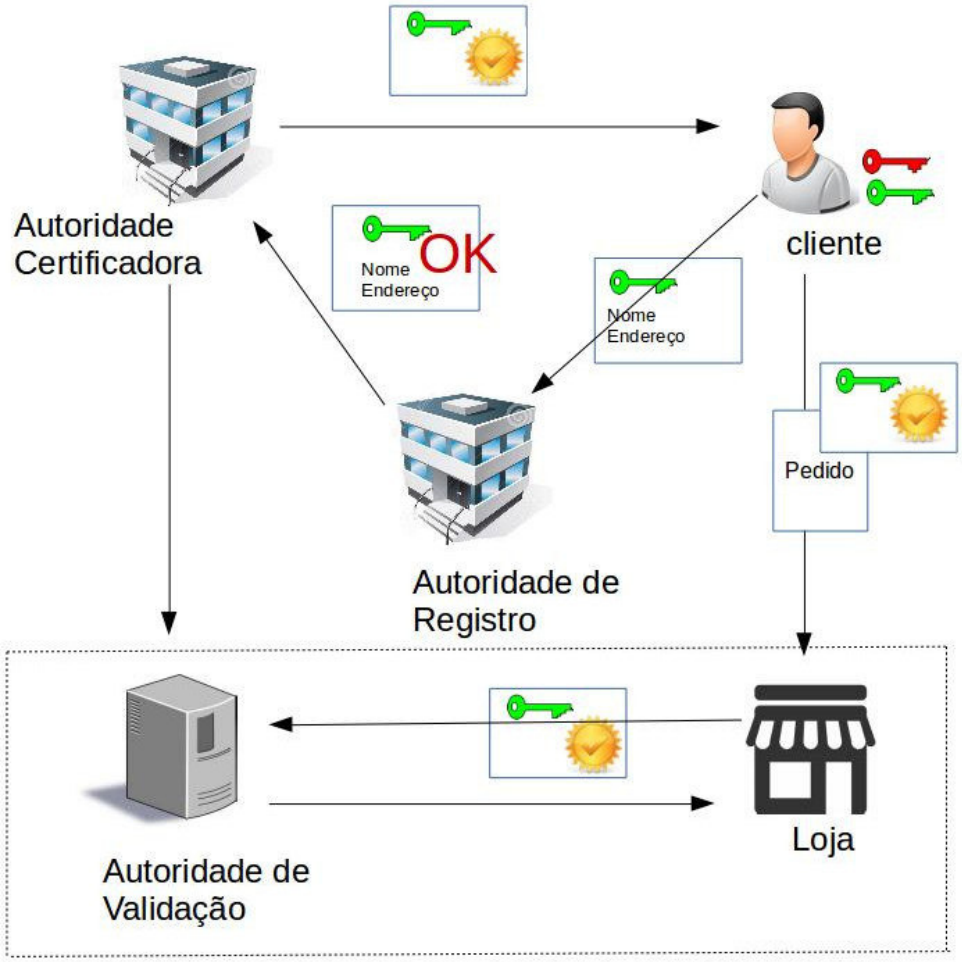
(ii) Autoridades de Registro (RA): garantem a ligação entre chaves públicas e identidades de titular de certificado ou outros atributos.

(iii) Cliente proprietário do PKC: pode assinar documentos digitais e descriptografar documentos usando chaves privadas.

(iv) Parte confiante (Loja): estabelecimento que confia em alguma autoridade para a verificação de certificados.

(v) Autoridade de Validação (VA): valida certificados digitais, impedindo a utilização de certificados revogados.

Figura 9 – Componentes da Infraestrutura de Chaves públicas baseada em certificados X.509



Fonte: Produzido pelo autor

Para começar a usar a PKIX, o cliente primeiro precisa registrar-se enviando uma solicitação para um PKC para uma CA. Este pedido contém algumas informações necessárias, como o nome do cliente e alguns atributos colocados em seu PKC. A CA, antes de emitir o certificado, irá verificar as informações fornecidas através de uma consulta à autoridade de registro (RA), em seguida, a CA irá assinar essas informações com a sua chave privada. Um PKC contém algumas informações como o nome da CA, o nome da entidade final com sua chave pública, um número de série de certificado, um período de validade e outras informações associadas. Para validar o certificado, a parte confiante

(Loja) envia o PKC à autoridade de validação (VA), que verifica a assinatura do certificado, verifica se a data está dentro do período de validade e também pode realizar outras verificações online.

3 Estudo de Caso

O estudo de caso considera a implementação do algoritmo ECDSA em diferentes sistemas de coordenadas de pontos, utilizando diferentes curvas elípticas e diferentes cargas de pedidos. A variável de resposta analisada aqui será o tempo médio de resposta que o servidor levará para validar o certificado de um cliente dado, estando o servidor já atendendo muitos outros pedidos de usuário, além do throughput, que é a quantidade de assinaturas que sai do servidor por segundo(taxa de saída). Cada experimento será executado 10 vezes. O tipo de projeto adotado é o fatorial completo. Na Tabela 5 é mostrado cada um dos fatores considerado nos experimentos e os níveis de cada um deles.

Tabela 5 – Configuração dos experimentos. Fatores considerados e os níveis de cada fator

Experimentos	Fatores			
	Algoritmo de Assinatura	Curva elíptica	Número de iterações (assinaturas enviadas)	Sistema de Coordenadas
1 2	ECDSA	secp160k1	1000	Afim Jacobiano
3 4		secp160k1	2000	Afim Jacobiano
5 6		secp256k1	1000	Afim Jacobiano
7 8		secp256k1	2000	Afim Jacobiano

Fonte: Produzido pelo autor

3.1 Resultados e Discussões

Os experimentos foram implementados em linguagem de programação Java. Para os testes, foi utilizado um notebook Acer Aspire AS5741-7840 com Intel Core i3-330M, 3GB de memória RAM e sistema operacional Windows 10. Foram desenvolvidos um servidor de validação e um cliente responsável por criar a carga de trabalho e analisar o tempo de reposta do último envio. Os níveis de carga de trabalho estão definidos em 1000 e 2000 iterações, ou seja, o envio de 1000 e 2000 assinaturas. Os envios obedecem a processos de Poisson (GOODMAN, 2006).

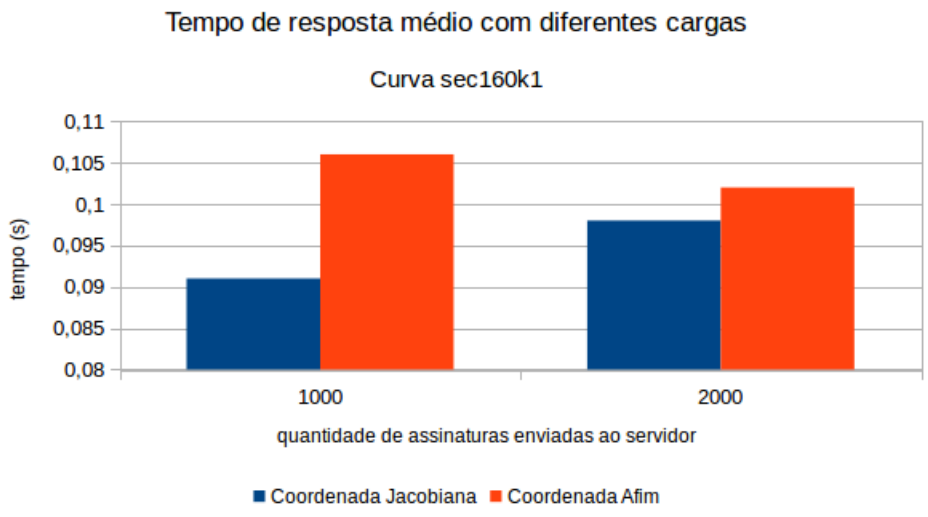
3.1.1 Experimentos 1, 2, 3 e 4

Nos experimentos 1 e 2, a curva elíptica utilizada no algoritmo de assinatura é a secp160k1 e o cliente envia 1000 assinaturas para o servidor. Como é percebido no gráfico representado pela Figura 11, o servidor implementando coordenadas Jacobianas produz um

throughput médio superior ao Afim, ou seja, ele consegue enviar mais respostas para clientes em uma unidade de tempo, nesse caso, 133,968 assinaturas por segundo. Observando o gráfico na Figura 10, mesmo com o servidor de validação Jacobiano recebendo um estresse maior, nesse caso uma média de 123,668 assinaturas por segundo contra 117,546 do outro, ele consegue oferecer um tempo médio de resposta ao cliente inferior ao servidor que implementa coordenadas afim.

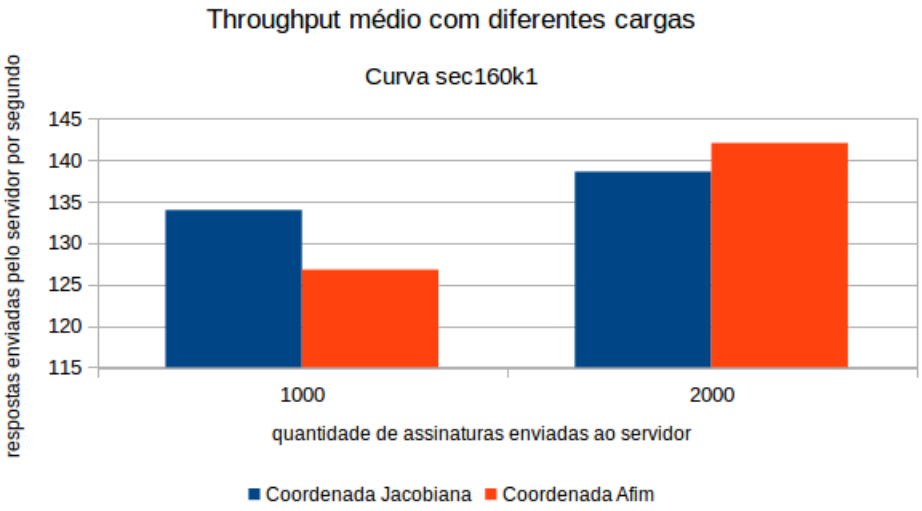
Já nos experimentos 3 e 4, o cliente envia 2000 assinaturas para o servidor utilizando a mesma curva, a secp160k1. Percebe-se nas figuras 10 e 11 uma inconsistência nos resultados, ou seja, o servidor que faz uso de coordenadas afim tem um throughput superior ao servidor de coordenadas Jacobianas, ainda assim o servidor afim oferece um tempo de resposta superior ao servidor jacobiano.

Figura 10 – Tempos de resposta médio nos experimentos 1, 2, 3 e 4



Fonte: Produzido pelo autor

Figura 11 – Throughput médio nos experimentos 1, 2, 3 e 4



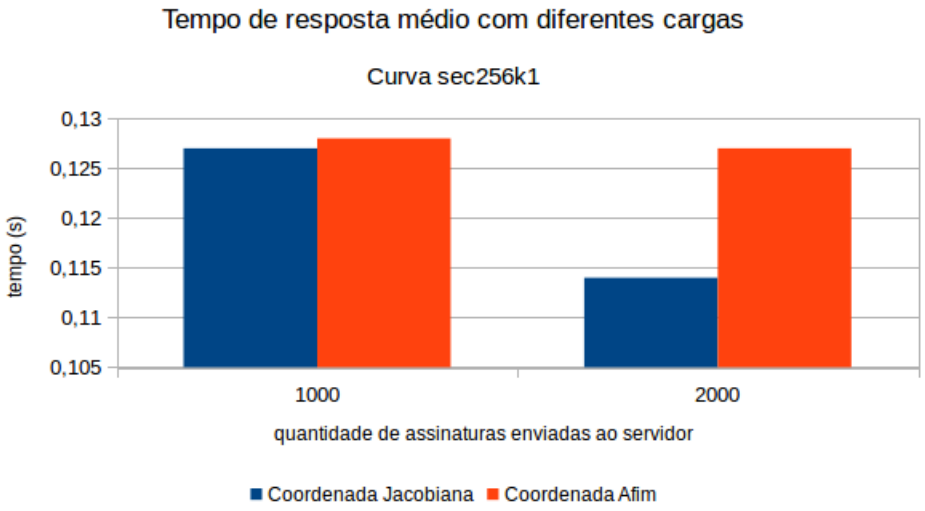
Fonte: Produzido pelo autor

3.1.2 Experimentos 5, 6, 7 e 8

Nos experimentos 5 e 6, a curva elíptica utilizada no algoritmo de assinatura é a secp256k1 e o cliente envia 1000 assinaturas para o servidor. Observa-se nos gráficos das Figuras 12 e 13 que o throughput médio é inferior e o tempo de resposta médio é superior em relação aos experimentos 1 e 2, pois a curva secp256k1 está sobre um corpo primo maior que a secp160k1, causando uma lentidão no algoritmo de verificação de assinatura e, por consequência, um maior tempo de resposta ao cliente. Já a comparação entre servidores de validação, o servidor que utiliza coordenadas jacobianas permanece sendo ligeiramente superior ao servidor afim, produzindo um throughput médio maior e um tempo de resposta médio ao cliente menor.

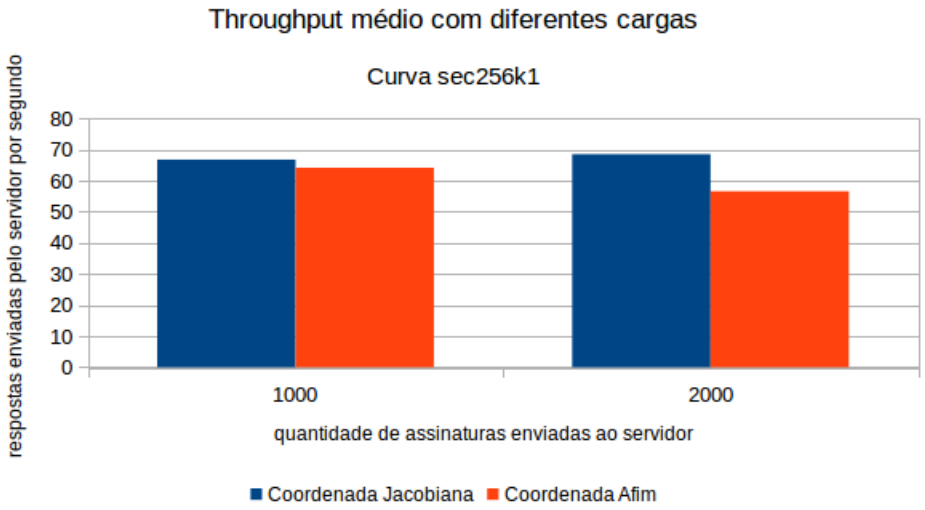
Nos experimentos 7 e 8, o cliente envia 2000 assinaturas ao servidor de validação utilizando a mesma curva, a secp256k1. O comportamento segue o mesmo padrão dos experimentos 5 e 6 comparados anteriormente, ou seja, o servidor que faz uso de coordenadas jacobianas produz um throughput médio superior ao servidor afim. Nota-se uma diferença maior entre os tempos de resposta médio produzidos pelos servidores e, mais uma vez, o servidor utilizando coordenadas jacobianas leva vantagem oferecendo um menor tempo de resposta médio.

Figura 12 – Tempos de resposta médio nos experimentos 5, 6, 7 e 8



Fonte: Produzido pelo autor

Figura 13 – Throughput médio nos experimentos 5, 6, 7 e 8



Fonte: Produzido pelo autor

4 Conclusão

A segurança é um item muito importante em sistemas que trabalham com informações sigilosas; não só a confidencialidade, como também a autenticidade e a integridade também são princípios básicos que norteiam esses tipos de aplicações. Além dos mecanismos de segurança, o desempenho também é muito importante para que a aplicação atenda a demanda, ou seja, forneça serviços a todos os usuários de uma infraestrutura em tempo hábil.

Neste trabalho, apresentou-se um referencial teórico com os principais conceitos utilizados na elaboração de um servidor de validação, que atua como uma autoridade de validação em uma infraestrutura de chaves públicas. Foram explicados os conceitos de assinatura digital, criptografia de curvas elípticas, sistemas de coordenadas, certificado digital, algoritmo ECDSA, entre outros. Em posse desses conceitos e definições, foram realizados experimentos para uma análise comparativa de parâmetros de um servidor de validação. Os parâmetros alterados foram: a curva elíptica utilizada pelo algoritmo de verificação, o número de assinaturas enviadas ao servidor e o sistemas de coordenadas usado no algoritmo ECDSA.

Após a realização dos experimentos, notou-se de um modo geral que o desempenho é ligeiramente superior no servidor de validação que utiliza as coordenadas Jacobianas em seu algoritmo de verificação ECDSA, apesar do aparecimento de uma inconsistência em um dos resultados. Cada experimento foi executado apenas 10 vezes para a extração de uma média para cada saída, devido as limitações de tempo para a realização do mesmo.

Como um trabalho futuro, devem ser testadas outras curvas elípticas, juntamente a outras representações de sistemas de coordenadas de ponto, se possível em um número maior de repetições para obtenção de médias mais precisas nas variáveis de saída. Também é pretendido avaliar o desempenho de servidores de validação utilizando implementações eficientes em software de curvas elípticas.

Referências

FIGUEIREDO, L. M. *Introdução à Criptografia*. [S.l.]: Centro de Estudos de Pessoal (CEP), 2010. v. 2. ISBN 85-7648-331-9. Citado na página 28.

GOODMAN, R. *Introduction to Stochastic Models*. second. New York: Dover Publications Inc., 2006. (Dover books on mathematics). Citado na página 41.

JAIN, R. *Art of Computer Systems Performance Analysis: Techniques for Experimental Design, measurement, simulation and modeling*. [S.l.]: Wyley Professional Computing, 1991. ISBN 0471503363. Citado 2 vezes nas páginas 24 e 25.

KATZ, J.; LINDELL, Y. *Introduction to Modern Cryptography, Second Edition*. 2nd. ed. [S.l.]: Chapman & Hall/CRC, 2014. ISBN 1466570261, 9781466570269. Citado 3 vezes nas páginas 23, 27 e 28.

MILLER, V. S. Use of elliptic curves in cryptography. In: *Advances in Cryptology*. London, UK, UK: Springer-Verlag, 1986. (CRYPTO '85), p. 417–426. ISBN 3-540-16463-4. Disponível em: <http://dl.acm.org/citation.cfm?id=646751.704566>. Citado na página 28.

N.KOBLITZ. *Elliptic Curve Cryptosystem*. [S.l.: s.n.], 1987. v. 48. 203–209 p. Citado na página 28.

SLAGELL, A.; BONILLA, R.; YURCIK, W. A survey of pki components and scalability issues. In: *2006 IEEE International Performance Computing and Communications Conference*. [S.l.: s.n.], 2006. p. 10–pp. Citado na página 38.

WASHINGTON, L. *Elliptic curves: number theory and cryptography*. [S.l.: s.n.], 2008. v. 50. Citado 4 vezes nas páginas 28, 29, 34 e 35.