Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Segurança da Informação

CRIPTOGRAFIA DE CURVA ELÍPTICA

GUILHERME DA SILVA NOGUEIRA

Americana, SP

2012

Faculdade de Tecnologia de Americana
Curso Superior de Tecnologia em Segurança da Informação

CRIPTOGRAFIA DE CURVA ELÍPTICA

GUILHERME DA SILVA NOGUEIRA

guilhy.nogueira@gmail.com

Trabalho de Conclusão de Curso desenvolvido em cumprimento á exigência curricular do Curso Superior de Tecnologia em Segurança da Informação, sob orientação do Prof. Me. Antônio César da Costa Barros.

Área: Criptografia

Americana, SP
2012

BANCA EXAMINADORA

Prof. Me. Antônio César da Costa Barros (Orientador)
Prof. Me. Carlos Henrique Sarro
Prof. Rogério Nunes de Freitas

AGRADECIMENTOS

Agradeço a minha família por me acompanharem nessa jornada pela perseguição dos meus sonhos; aos meus amigos, por estarem lá nos momentos de alegria e tristeza; a minha namorada, da qual o sorriso me traz forças renovadas para lutar pelos meus objetivos, aos professores, especialmente ao meu orientador do trabalho, por acreditarem e moldarem meu potencial; e a Roberto Gallo, pelo apoio técnico.

DEDICATÓRIA

Dedico esse projeto a todos que apoiaram, direta ou indiretamente, e acreditaram na conclusão do mesmo.

RESUMO

Confidencialidade, integridade, autenticidade e não repúdio são requisitos de segurança muito necessários para qualquer sistema, e a criptografia é a ferramenta utilizada para prover tais requisitos. O método varia de acordo com a cifra utilizada, porém todos utilizam como premissa problemas matemáticos de difícil resolução para proporcionar sigilo sobre uma informação. A implementação dos CCE (Criptossistemas de Curva Elíptica) apresenta um desafio diferente dos atuais algoritmos do mercado, trabalhando sobre o problema do logaritmo sobre curvas elípticas como sua premissa principal. Este trabalho faz um estudo de criptografia de chave pública baseada em curvas elípticas, comparando com outros métodos de criptografia utilizados, apresentando ao final um exemplo de aplicação deste método.

Palavras chaves: curva elíptica; criptografia; segurança.

ABSTRACT

Confidentiality, integrity, authenticity and non-repudiation are much needed security requisites for any system, and cryptography is the tool used to provide those requisites. The method varies according to the used cipher, but all of them use as premise hard-solving mathematical problems in order to provide secrecy of an information. The implementation of the ECC (Elliptic Curve Cryptosystems) presents a different challenge of the latest market algorithms, working with the elliptic curve discrete logarithm as its main premise. This paper is a study on public key cryptography based on elliptic curves, comparing with other cryptographic methods, presenting at the end an application example of this method.

Keywords: elliptic curve; cryptography; security.

SUMÁRIO

1. INTRODUÇÃO 12

2. CRIPTOGRAFIA ANTIGA 14

3. CRIPTOGRAFIA EM REDE 19

3.1. Tipos de Algoritmos 19

3.1.1. Algoritmos de chave simétrica 19

3.1.2. Algoritmos de chave assimétrica 20

3.2. Exemplos de Algoritmos 20

3.2.1. Algoritmos de chave simétrica 20

3.2.2. Algoritmos de chave assimétrica 23

4. CONCEITOS DE MATEMÁTICA DISCRETA 26

4.1. Fundamentos sobre Grupos 26

4.1.1. Grupo abeliano 27

4.2. Fundamentos sobre Corpos 27

4.3. Corpo Finito Primo F_p 27

5. CURVAS ELÍPTICAS 29

5.1. Definição das Curvas Elípticas 29

5.2. Definição dos Pontos na Curva 30

5.3. Operações em Curvas Elípticas 31

5.3.1. Adição de Pontos 31

5.3.2. Subtração de Pontos 32

5.3.3. Multiplicação de Pontos 33

6. CRIPTOGRAFIA DE CURVA ELÍPTICA 34

6.1. A utilização das curvas em criptografia 34

6.2. Parâmetros de domínio 35

6.3. Esquema criptográfico 36

6.4. Algoritmos 38
6.4.1. ECDH 38
6.4.2. ECIES 39
6.4.2.1. Padronização de parâmetros de domínio 40
6.4.2.2. Geração do par de chaves 40
6.4.2.3. Processo criptográfico 41
6.4.3. ECDSA 42
6.5. Segurança da ECC 42
6.6. Aplicações da ECC 43
7. DEMONSTRAÇÃO PRÁTICA 45
8. CONCLUSÃO 46

LISTA DE FIGURAS

Figura 1 – Bastão de Licurgo com dizeres. 15
Figura 2 - Máquina Enigma com três rotores, teclado, luzes e conexões. 17
Figura 3 - Representação da curva $y^2 = x^3 - x$. 29
Figura 4 - Representação da curva $y^2 = x^3 - x + 1$. 30
Figura 5 - Exemplos de utilização da reta calculada entre os pontos P e Q para se encontrar um terceiro ponto na curva (R). 31
Figura 6 - Ilustração do processo de adição de dois pontos P1 e P2. 32

LISTA DE TABELAS

Tabela 1 - Comparação de tamanhos de chaves em relação ao esforço computacional para criptoanálise. 43

1. INTRODUÇÃO

A criptografia tem sido utilizada há séculos em contextos militares e diplomáticos com a finalidade de prover sigilo de informações. Na atual era das comunicações eletrônicas, os requisitos de segurança – confidencialidade, integridade, autenticação e não repúdio – assumem um papel importantíssimo, papel este, auxiliado pela criptografia.

Em 1976 foi apresentado pela primeira vez por Whitfield Diffie e Martin Hellman o conceito de criptografia de chave pública. Embora ainda em caráter teórico, este novo conceito gerou uma intensa atividade de pesquisa e desenvolvimento de sistemas criptográficos práticos de chave pública.

Pouco tempo depois, em 1978, Ron Rivest, Adi Shamir e Lan Adleman viriam a apresentar o primeiro esquema de chave pública para criptografia, utilizado para assinatura e ciframento, derivado da primeira letra de seus nomes, o RSA. Este esquema utilizava um problema de difícil resolução como sua premissa criptográfica, a fatoração de números inteiros muito grandes.

Em 1984 foi apresentado por El Gammal outro sistema criptográfico baseado no problema do logaritmo discreto. Esse sistema tem sido refinado e aplicado a vários protocolos, e uma de suas extensões serve de base para o algoritmo de assinatura digital DSA.

Constantes desenvolvimentos de algoritmos eficientes na resolução do problema do logaritmo discreto sobre corpos finitos forçou o aumento no tamanho das chaves utilizadas nos protocolos Diffie-Hellman, tornando-o mais caro e menos desejado. Essa situação levou vários pesquisadores a observar a possibilidade de extensão dos algoritmos aos grupos abelianos arbitrários. Com essa extensão o problema do logaritmo discreto parece ser intratável e as operações do grupo passam a ser implementadas em software ou em hardware.

Em 1987 Neil Koblitz e Victor Miller propuseram utilizar o grupo dos pontos de uma curva elíptica sobre um corpo finito para programar criptossistemas de chave pública. Sua segurança baseia-se na suposta intratabilidade do problema do

logaritmo discreto no grupo de pontos de uma curva elíptica.

Atualmente muitos avanços foram realizados na área de criptografia sobre curva elíptica. O melhor algoritmo conhecido para o problema do logaritmo discreto em CE é de tempo exponencial. Isto possibilita o uso de chaves de tamanho menor comparadas com os algoritmos do problema do logaritmo discreto. Uma chave de 3072 bits baseada em RSA é equiparada em segurança com uma chave de 256 bits baseada em curva elíptica. Isto resulta em uma melhor velocidade de algoritmo e provê chaves e certificados de menor tamanho, adequados para o uso em rede.

2. CRIPTOGRAFIA ANTIGA

“A criptografia é a arte e a ciência da escrita secreta.” (CALLAS, 2008).

A palavra criptografia deriva das palavras Gregas “*krypto*”, que significa oculto e “*graphos*”, escrever. É a ciência que estuda o conjunto de conceitos e técnicas usados na codificação de uma informação de forma que somente o emissor e o destinatário, em posse da chave ou segredo, possam interpretá-la. Através deste mecanismo, uma terceira pessoa que não possua as chaves criptográficas não interpretará facilmente as informações interceptadas.

O processo de aplicar-se alguma técnica de criptografia a uma mensagem é denominado cifrar a mensagem, cifração ou codificação. O processo inverso, ou seja, aplicar uma técnica para retornar a informação ao seu estado original, por sua vez, é chamado decifrar a mensagem ou decifração. A mensagem na forma de texto inteligível, isto é, interpretável naturalmente é chamada de texto plano, enquanto a mensagem resultante de um processo de cifração é então chamada apenas de cifra. A ciência que estuda uma maneira de decifrar as mensagens é chamada de Criptoanálise (SINGH, 2004), que juntamente à Criptografia são ramos de estudo da Criptologia.

Há séculos a premissa de sigilo da informação da criptografia é de grande interesse para órgãos militares e diplomáticos, visto que as operações básicas destas organizações dependem de informações trocadas em segredo. Na era moderna, os requisitos de segurança tais como confidencialidade, integridade, autenticação e não repúdio, providos por mecanismos em que se têm auxílio da criptografia, têm grande importância devido à alta quantidade de informações sigilosas transmitidas através de meios inseguros, como a Internet.

A criptografia, em sua definição geral, sempre foi utilizada por governantes e pelo povo, em épocas de guerra e de paz. Segundo Singh (2004), a criptografia faz parte da história humana, pois sempre existiram fórmulas secretas, informações confidenciais e interesses dos mais diversos ramos que não podiam ser escritos de uma maneira que fosse de fácil interpretação pelo público em geral ou pelo inimigo, uma vez que esta fosse interceptada.

Segundo Tkotz (2004), nos tempos antigos, culturas como Egito, China, Índia e da Mesopotâmia desenvolveram a técnica chamada Esteganografia, que se trata não de Criptografia, onde um algoritmo era aplicado à mensagem, mas na simples ideia de ocultação da mensagem em lugares insuspeitos. Como por exemplo, um mensageiro tinha seus cabelos raspados e então a mensagem em texto inteligível era escrita em seu couro cabeludo. A velocidade da entrega não era uma prioridade, então esperavam os cabelos do mensageiro crescerem o suficiente para ocultar a mensagem e somente então o enviavam para seu destino. Mesmo que abordado por mandados de outros reinados, à primeira vista o mensageiro não carregava nada além do normal e era, então, permitida sua passagem. Ao chegar ao seu destino, o mensageiro tinha seus cabelos raspados novamente e a mensagem recebida.

Outro exemplo de Esteganografia eram tábuas que continham mensagens em texto plano, e eram então cobertas de cera para que a escrita não fosse visível a princípio. Para sua leitura, bastava-se derreter a cera preservando a tábua. Os chineses utilizavam uma fita de seda finíssima para se escrever a mensagem, que depois eram amassadas até formar uma pequena esfera para então ser coberta de cera como impermeabilizante e então engolida pelo seu transportador.

Kahn (1967) mostra que o sistema criptográfico militar mais antigo, conhecido como *Bastão de Licurgo*, consiste em um dispositivo para esconder mensagens, utilizando um bastão de madeira enrolado firmemente por uma tira longa e estreita de couro ou pergaminho. Após escrita a mensagem no sentido do comprimento do bastão, a tira era então desenrolada. Para decifrar a mensagem, um bastão de mesmo diâmetro era necessário.

Figura 1 – Bastão de Licurgo com dizeres. (Caetano, 2004)

A *Cifra de César*, conforme pesquisa de Tkotz (2004), era utilizada em aplicação de sigilo em mensagens governamentais. Para compor seu texto cifrado,

Júlio César (Ditador da República Romana de 49 a.C. a 44 a.C.) alterava letras desviando-as em 3 posições. Neste sistema, a letra *A* se tornava *D*, *B* se tornava *E,* e assim por diante. Para decifrar, bastava calcular as letras três posições atrás, chegando à mensagem original.

Embora historicamente a cifra fosse aplicada com três casas, esta chave pode ser modificada e acordada entre as duas partes, para que o processo de quebra seja dificultado, embora o número de casas que podem variar seja muito pequeno. A análise da repetição das letras baseado em estudos do idioma, que aponta quais letras tem maior utilização nas palavras que o compõem e, portanto repetidas com maior frequência, pode ajudar na quebra deste código visto que cada letra é traduzida separadamente.

Embora seja uma cifra simples, a Cifra de César é aplicada nos dias atuais para exemplificar o funcionamento da criptografia, pois trata de uma definição literal de uma cifra de substituição, mesmo que bastante rudimentar para os dias atuais.

Segundo Singh (2004) no início do século XX, o surgimento de máquinas automatizadas e o desenvolvimento tecnológico facilitaram a expansão de uso e aceleraram o desenvolvimento estrutural da criptografia. A difusão de redes de telecomunicações como telégrafo e rádio acelerou a comunicação humana, e com isso incentivou a evolução da maneira com que as informações sigilosas eram então transmitidas. A evolução das técnicas de segredo, portanto, era iminente, no qual foi exercido um forte incentivo governamental e militar nas pesquisas de seu desenvolvimento.

Singh (2004) relata que o exemplo mais relevante de criptografia não informatizada é a máquina *Enigma* (figura abaixo). Trata-se de uma máquina eletromecânica de cifração e decifração, patenteada pelo engenheiro alemão Arthur Scherbius em 1918. Este equipamento ficaria famoso pelo seu uso pelos alemães na Segunda Guerra Mundial.

Figura 2 - Máquina Enigma com três rotores, teclado, luzes e conexões. (Ellsbury, 1998)

O mecanismo consiste num teclado, num conjunto de discos rotativos chamados rotores, dispostos em fila; e de um mecanismo de avanço que faz andar alguns rotores uma posição quando uma tecla é pressionada. O mecanismo varia entre diversas versões da máquina, mas o mais comum é o rotor colocado à direita avançar uma posição com cada tecla pressionada, e ocasionalmente provocar o movimento rotativo dos restantes rotores, à sua esquerda, à semelhança do mecanismo contador de quilômetros de um automóvel.

A parte mecânica funciona de modo a variar um circuito elétrico que efetua a cifra de cada letra pressionada no teclado. Ao pressionar uma tecla, o circuito se fecha: a corrente elétrica flui pelos diversos componentes (pela ordem teclado, conexões para câmbio de codificação, rotores, rotor-espelho, rotores pela ordem inversa e placa de luzes). A luz que no fim do processo se acende codifica a letra pressionada no teclado. O movimento dos rotores provoca alterações nas conexões entre rotores, acarretando em diferentes combinações na codificação.

Em 1918, a Marinha Alemã tomou interesse pela máquina e adquiriu alguns exemplares, adaptando seus métodos de criptografia para comunicações a ela. A

partir de 1930, a máquina fora abordada por boa parte do exército alemão. Sua popularidade se deve ao fácil manuseio e operação para processamento de mensagens e pela suposta indecifrabilidade do código.

Também em 1930 foi introduzido um *plugboard* nas versões do exército alemão e posteriormente pela marinha também. O *plugboard* consiste em um painel com um *plug* para cada letra, que permitia que fossem conectadas duas a duas. O resultado da conexão entre duas letras acarretava a substituição de uma por outra no momento em que uma era acionada. Por exemplo, se as letras 'a' e 'b' estivessem plugadas, ao se pressionar 'a', era acionado o circuito de 'b' no rotor mais a direita e vice-versa.

No entanto, as codificações das comunicações alemãs foram quebradas e assim suas informações expostas aos inimigos, arruinando diversos de seus planos. A interceptação das informações levou à interferência dos inimigos nos planos da União Soviética, impulsionando então a Segunda Guerra ao seu fim em 1945. A localização dos navios Soviéticos era recebida e então era enviado um avião para que disfarçasse uma viagem de reconhecimento, fazendo a descoberta e o então bombardeio parecer acidental e não planejado através das informações interceptadas.

Com o desenvolvimento da tecnologia, o próximo passo nas técnicas de criptografia teve um grande avanço para as comunicações secretas, pois com a entrada da informática e a computação em rede, os algoritmos de criptografia se tornaram cada vez mais fortes e avançados, à medida que o poder computacional e as teorias permitiam.

3. CRIPTOGRAFIA EM REDE

Com a introdução dos computadores e seus cálculos ultrarrápidos, os algoritmos de criptografia tiveram sua evolução acelerada e sua força elevada, visto que agora havia a possibilidade de se utilizar chaves de um maior tamanho, dificultando assim a sua quebra por força bruta.

3.1. TIPOS DE ALGORITMOS

Os algoritmos criptográficos são divididos em duas categorias: algoritmos de chave simétrica e algoritmos de chave pública (ou chave assimétrica). Basicamente, a diferença entre os dois tipos de algoritmo é que no primeiro, utiliza-se a mesma chave para codificar e decodificar a mensagem, enquanto no segundo, chaves diferentes são utilizadas para tais finalidades. A seguir são descritos ambos os tipos, assim como sua funcionalidade.

3.1.1. Algoritmos de chave simétrica

De acordo com Singh (2004), os algoritmos de chave simétrica são assim chamados, pois a chave utilizada no processo de codificação e decodificação da mensagem é a mesma. Foi o primeiro tipo dos algoritmos que surgiram na era dos computadores, devido à sua simplicidade estrutural.

Sua força quanto a ataques de força bruta é razoável, levando em consideração falhas estruturais no tipo do algoritmo e de tamanho da chave empregada, porém ao interceptar a fase de troca de chaves no início de uma comunicação, um invasor tem acesso direto ao conteúdo da mensagem, afinal as chaves são iguais. Ainda assim, alguns destes algoritmos estão em uso atualmente, dependendo da finalidade ao qual é empregado.

Este fator foi crucial para o interesse em desenvolvimento de novas

tecnologias que preservassem a chave nessas duas etapas do processo, como veremos a seguir.

3.1.2. Algoritmos de chave assimétrica

Segundo Singh (2004), o surgimento dos algoritmos de chave pública, ou chave assimétrica, se deu pela necessidade de preservar a chave de decodificação, onde mesmo que uma transmissão de chave fosse interceptada, o dado ainda poderia ser transmitido com segurança, pois não haveria possibilidade de utilizar apenas a chave interceptada para se decodificar a mensagem.

Embora este esquema não cobre todas as eventuais falhas estruturais de segurança, a combinação deste com outras tecnologias (como Assinatura Digital) garantem uma entrega de informações segura e confiável, com autenticação de remetente.

3.2. EXEMPLOS DE ALGORITMOS

A seguir estão listados algoritmos dos dois tipos existentes juntamente com uma breve descrição de sua criação, funcionamento e utilização. Estes resumos foram baseados na obra de Stallings (2011).

3.2.1. Algoritmos de chave simétrica

Os algoritmos de chave simétrica apresentam certa deficiência de segurança visto que a chave deve ser portada por ambos os lados da transmissão e, assim, são aplicados em casos específicos ou em mecanismos que não necessitam transmitir as chaves e/ou mensagens por um meio inseguro.

- Máquina Enigma

A máquina Enigma, conforme abordada no primeiro capítulo, utilizava códigos que eram selecionados através de seus rotores. Estes códigos eram utilizados tanto no processo de codificação quanto no de decodificação, portanto sua cifra, embora não seja ainda considerada computacional, é classificada como simétrica.

- DES

O DES (*Data Encryption Standard*) é uma cifra selecionada como FIPS oficial (*Federal Information Processing Standard*) pelo governo dos EUA em 1976 e que foi utilizado em larga escala internacionalmente. O algoritmo era inicialmente controverso, com um tamanho de chave pequeno e suspeita de um backdoor da NSA (National Security Agency). O DES foi estudado academicamente e motivou os sistemas modernos de entendimento da criptoanálise. O DES é atualmente considerado inseguro para muitas aplicações. Isto se deve principalmente a pequena chave de 56-bit.

Em Janeiro de 1999 a *distributed.net* e a *Electronic Frontier Foundation* juntas violaram uma chave DES em 22 horas e 15 minutos. Embora existam alguns resultados analíticos obtidos teoricamente que demonstram a fragilidade da cifra, estes são improváveis de se aplicar em situações práticas.

Acredita-se que o algoritmo seja seguro na forma de 3DES embora existam ataques teóricos. Recentemente o DES foi substituído pelo AES.

- IDEA

A cifra *IDEA* (*International Data Encryption Algorithm*) foi criada em 1991 por *James Massey* e *Xuejia Lai*. O IDEA é um algoritmo de cifra de bloco que faz uso de chaves de 128 bits e que tem uma estrutura semelhante ao DES. A cifra foi concebida no âmbito de um contrato de investigação com a Fundação Hasler, que se tornou parte da Ascom-Tech AG. A cifra é patenteada em vários países, mas está disponível gratuitamente para uso não comercial.

- Blowfish

Algoritmo criado como substituto do DES e IDEA. Ele toma uma chave de tamanho variável, de 32 a 448 bits, tornando-o ideal para aplicações tanto domésticas, quanto comerciais. O Blowfish foi desenvolvido em 1993 por Bruce Schneier como uma alternativa grátis mais rápida para os algoritmos criptográficos existentes. Desde então, ele vem sendo analisado de forma considerável e está conquistando a aceitação do mercado como um algoritmo forte.

- AES

Em 1997 o governo americano, através do NIST (National Institute of Standards and Technology), lançou um processo de seleção que definiria um novo algoritmo de chave simétrica para proteger informações do governo federal. Este novo algoritmo criptográfico substituiu o DES (Data Encryption Standard), que havia sido quebrado pela máquina DES Cracker, construída pela ONG *Electronic Frontier Foundation* com apenas 250 mil dólares.

Em setembro de 1997 o NIST indicou as condições necessárias para a candidatura de algoritmos para substituir o DES: divulgação pública, direitos autorais livres, e os algoritmos deveriam ser de chave privada (simétricos) e suportar blocos de 128 bits e chaves de 128, 192 e 256 bits. Em agosto de 1998, na primeira fase de seleção dos concorrentes, apresentaram-se 15 candidatos. O NIST solicitou aos membros da comunidade criptográfica mundial uma análise dos algoritmos candidatos. Em 1999, na Segunda Conferência dos Candidatos AES, através da análise obtida foram selecionados cinco finalistas: MARS, RC6, Rijndael, Serpent e Twofish. Posteriormente esses cinco algoritmos sofreram novas análises e seus criadores participaram de debates, fóruns, etc.

Após três anos e meio do início do concurso, o NIST chega à escolha do vencedor: o algoritmo Rijndael. O nome é uma fusão de Vincent Rijmen e Joan

Daemen, os dois belgas criadores do algoritmo. Segundo o NIST, ele combina as características de segurança, desempenho, facilidade de implementação e flexibilidade. O Rijndael apresenta alta resistência a ataques como *power attack* e *timing attack* e exige pouca memória, o que o torna adequado para operar em ambientes restritos como *smart cards*, PDAs e telefones celulares.

3.2.2. Algoritmos de chave assimétrica

Os algoritmos de chave assimétrica surgiram após alguns anos de utilização dos algoritmos de chave simétrica. Sua principal intenção é de tornar segura a transmissão de dados através de uma rede insegura, como por exemplo, a internet. Utiliza-se do preceito de que é muito difícil se obter a chave privada de um conjunto de chaves a partir da chave pública.

- Diffie-Hellman

Criado em 1976 por Whitfield Diffie e Martin Hellman, daí o nome Diffie-Hellman, foi o primeiro algoritmo de chave pública a ser inventado. Isto significa que seus autores também são os donos da ideia. O algoritmo pode ser usado para a distribuição de chaves, mas não para cifrar ou decifrar mensagens. Sua segurança reside na dificuldade de calcular logaritmos discretos num corpo finito comparada com a facilidade de realizar uma exponenciação no mesmo corpo.

- El-Gammal

Algoritmo assimétrico não comutativo cuja assimetria se baseia na dificuldade de se extrair logaritmos discretos em corpos finitos. Este algoritmo não é patenteado, mas sua versão para cifragem é uma variante do algoritmo de Diffie-Hellman. A detentora de patente para o D&H (PKP Inc.) reclamou direitos para licenciar seu uso até abril de 1997. Criado pelo estudioso de criptografia egípcio Taher Elgamal em 1984.

- RSA

O algoritmo RSA deve o seu nome a três professores do Instituto MIT (*Massachusetts Institute of Technology*), *Ronald Rivest, Adi Shamir* e *Leonard Adleman*, que inventaram este algoritmo — até a data (2008), a mais bem sucedida implementação de sistemas de chaves assimétricas, e fundamenta-se em teorias clássicas dos números. É considerado dos mais seguros, já que mandou por terra todas as tentativas de quebrá-lo. Foi também o primeiro algoritmo a possibilitar criptografia e assinatura digital, e uma das grandes inovações em criptografia de chave pública.

O RSA envolve um par de chaves, uma chave pública que pode ser conhecida por todos e uma chave privada que deve ser mantida em sigilo. Toda mensagem cifrada usando uma chave pública só pode ser decifrada usando a respectiva chave privada. A criptografia RSA atua diretamente na internet, por exemplo, em mensagens de e-mails, em compras on-line e o que você imaginar; tudo isso é codificado e decodificado pela criptografia RSA.

Baseia-se no fato de que, embora seja fácil encontrar dois números primos de grandes dimensões (p.e. 100 dígitos), conseguir fatorar o produto de tais dois números é considerado computacionalmente complexo (em outras palavras, o tempo estimado para consegui-lo ronda os milhares de anos). De fato, este algoritmo mostra-se computacionalmente inquebrável com números de tais dimensões, e a sua força é geralmente quantificada com o número de bits utilizados para descrever tais números. Para um número de 100 dígitos são necessários cerca de 350 bits, e as implementações atuais superam os 512 e mesmo os 1024 bits (se divide por 8 para conseguir em bytes).

É usado comumente para transferir senhas por ser mais rápido. A senha, geralmente, tem apenas 128 bits (16 bytes) o que facilita o manuseio, já que os processadores modernos têm tipos de 16 bytes embora restringidos pelo número de operações. Geralmente o servidor, como por exemplo, o servidor HTTPS, gera um par de chaves, uma chave pública e uma chave privada, transmite a chave pública para o cliente, e este gera uma senha *hashed*, criptografa com a chave pública do

servidor e envia de volta para o servidor. Assim, tanto o receptor quanto o servidor podem usar a senha *hashed* de forma segura para codificar e decodificar.

- Curvas Elípticas

Foi proposta de modo independente por Neal Koblitz e Victor Miller em 1985. A Criptografia de Curvas Elípticas, ou ECC, das iniciais em inglês *Eliptic Curve Cryptography*, é uma variante da criptografia assimétrica ou de chave pública, baseada na matemática das curvas elípticas. Seus criadores argumentam que a ECC pode ser mais rápida e usar chaves mais curtas do que os métodos antigos — como RSA --, e proporcionar ao mesmo tempo um nível de segurança equivalente.

4. CONCEITOS DE MATEMÁTICA DISCRETA

A seguir são apresentados brevemente, baseados na obra de Menezes (2003), conceitos matemáticos utilizados no cálculo das curvas, que serão abordados no tópico seguinte. Os tópicos serão abordados brevemente como uma revisão de conceitos gerais da Matemática Discreta que são de grande importância para os cálculos sobre curvas elípticas.

4.1. FUNDAMENTOS SOBRE GRUPOS

Segundo Cheng (2004), *Grupo* é "um sistema algébrico definido sobre um conjunto G onde a operação binária ○ satisfaz um determinado número de requisitos como a existência de um elemento identidade e um inverso." Uma operação binária sempre aceita dois parâmetros e é representada pelo símbolo ○, pois se abstrai a operação matemática pela representação de uma operação qualquer.

Nesta definição, para todo *x* e *y* de um grupo, uma operação binária ○ é equivalente para ambas às posições de *x* e *y*. Esta operação é uma representação de qualquer operação matemática (+ e ×) que pode ser aplicada aos elementos.

Um grupo deve obrigatoriamente apresentar as seguintes propriedades:

- **Associatividade:** Quaisquer elementos *a, b, c* pertencentes a *G*, onde devem apresentar $(a \circ b) \circ c = a \circ (b \circ c)$;
- **Existência do elemento neutro:** Existe um elemento *e* em *G* tal que $e \circ a = a \circ e = a$, para todo elemento em *G*;
- **Existência de um elemento simétrico:** Para qualquer elemento *a* em *G*, existe outro elemento *a'* em *G*, tal que $a \circ a' = a' \circ a = e$, onde *e* é o elemento neutro previamente mencionado.

4.1.1. Grupo abeliano

Um Grupo abeliano é determinado pela satisfação, além de todas as propriedades de grupo, do teorema da comutatividade conforme segue:

$$\forall x, y \in G, x \circ \mathrm{y} = \mathrm{y} \circ \mathrm{x}$$

4.2. FUNDAMENTOS SOBRE CORPOS

Cheng (2004) descreve Corpo como "*um sistema algébrico definido em um conjunto F com duas operações binárias adição e multiplicação*", satisfazendo determinados parâmetros. São eles:

- (*F*, +) é um grupo Abeliano;
- (*F* \ {0}, ×) é um grupo Abeliano, onde {0} é a identidade da adição e zero da multiplicação;
- São verdadeiros na regra da distributividade, que o autor exemplifica como:

$$\forall x, y, z \in F : x \times (y + z) = x \times y + x \times z; (x + y) \times z = x \times z + y \times z$$

4.3. CORPO FINITO PRIMO F_P

Para F_p onde *p* é primo, se define o um conjunto numérico *F = {0, 1,..., p-1}* com duas operações:

- Adição: $\forall a, b \in F_p, r \equiv a + b \ mod \ p$;
- Multiplicação: $\forall a, b \in F_p, r \equiv a \times b \ mod \ p$;

Esta propriedade limita o corpo sobre o qual os resultados serão trabalhados. Em geral, os resultados são analisados sobre o corpo infinito, portanto não é necessária a preocupação com tais limites. A aplicação matemática de tal limite trata de uma função modular (*x mod y*), onde *x* é o número que deve ser adequado ao

limite do corpo e *y*, o próprio limite do corpo. A realização simplificada de tal cálculo se dá pela divisão de x por y, retornando o resto da mesma.

Diversas teorias matemáticas foram desenvolvidas para aceleração de cálculos executados sobre tais corpos. Esta aceleração trabalha, em casos como expoentes de grande ordem, na simplificação de tais operações, atingindo o objetivo de cálculo de uma maneira muito mais rápida e eficiente em termos de recursos computacionais.

O corpo finito em um número primo garante que uma multiplicação não resulte no elemento neutro da soma, portanto não haverá um valor 0 entre os pontos resultantes de operações matemáticas efetuadas.

5. CURVAS ELÍPTICAS

As Curvas Elípticas são definidas mediante equações cúbicas (de terceiro grau). Segundo Koblitz (2008), as Curvas Elípticas são empregadas nas comprovações de teoremas complexos, como por exemplo, o Último Teorema de Fermat, fatoração de inteiros e, claro, criptografia. O nome deriva da semelhança de sua representação com uma elipse, visto que sempre apresenta uma grande curva semicircular devido ao emprego de uma raiz quadrada na equação. Vale lembrar, porém, que estas curvas não são elipses.

5.1. DEFINIÇÃO DAS CURVAS ELÍPTICAS

As curvas elípticas são regulares, ou não singulares, que significa que não têm cúspides (pontas) nem interseções (cruzamentos da reta), e se pode definir uma operação binária para o conjunto de seus pontos de uma maneira geométrica natural, o que faz deste conjunto um grupo Abeliano.

As curvas elípticas sobre o corpo dos números reais são dadas pelas equações $y^2 = x^3 - x$ e por $y^2 = x^3 - x + 1$.

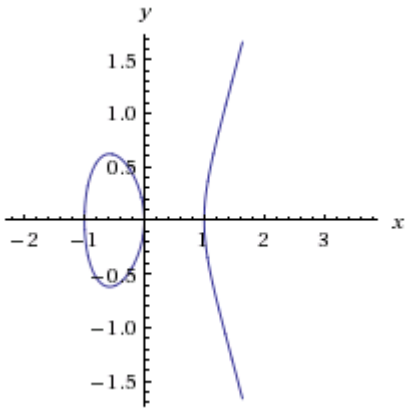

Figura 3 - Representação da curva $y^2 = x^3 - x$. (WolframAlpha, 2012)

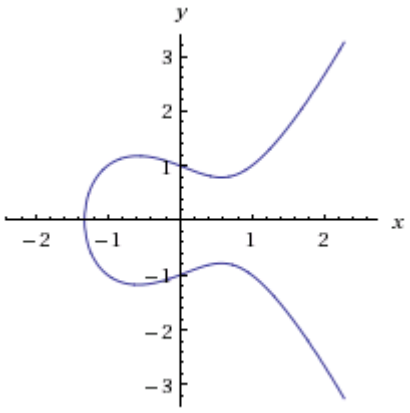

Figura 4 - Representação da curva $y^2 = x^3 - x + 1$. (WolframAlpha, 2012)

As curvas elípticas podem definir-se sobre qualquer corpo *K*; a definição formal de uma curva elíptica é a de uma curva algébrica projetiva não singular sobre *K* de gênero 1.

Se a característica de *K* não é nem 2 nem 3, então toda curva elíptica sobre *K* pode escrever-se na forma: $y^2 = x^3 - px - q$ onde *p* e *q* são elementos de *K* tais que o polinômio do membro direito $x^3 - px - q$ não tenha nenhuma raiz dupla. Se a característica é 2 ou 3, farão falta mais termos.

5.2. DEFINIÇÃO DOS PONTOS NA CURVA

De acordo com Koblitz (2008), a curva é definida como o conjunto de pontos *(x, y)* que satisfazem a equação dada de um modo que *x* e *y* sejam elementos do limite do grupo *K*. Os pontos da curva cujas coordenadas pertençam ambas a *K* são chamados de *pontos K-racionais*.

Selecionando dois pontos na curva, denominados *P* e *Q*, podemos descrever então um terceiro ponto que surge da intersecção dos pontos *P* e *Q* por uma reta, atingindo assim outro ponto na curva. Se a reta for tangente à curva em apenas um ponto, este ponto é contado duas vezes. E se a reta é paralela ao eixo y, então se define o terceiro ponto (resultante) como “no infinito”, pois este não atingirá a curva

em outro ponto.

Deste modo, uma das tais condições definirá o ponto resultante na curva. A seguir, são apresentados quatro exemplos de como o ponto é encontrado, seguindo as regras previamente apresentadas para se desenhar a reta.

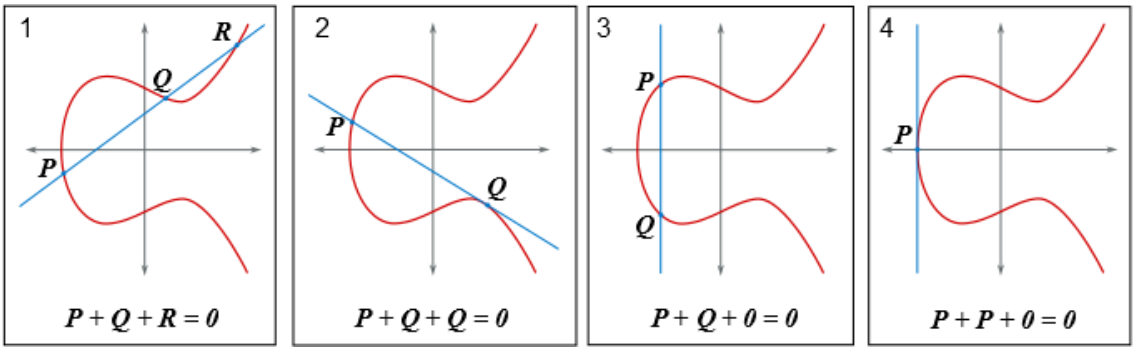

Figura 5 - Exemplos de utilização da reta calculada entre os pontos P e Q para se encontrar um terceiro ponto na curva (R). (MathWorld, 2010)

Com base nessa teoria dos pontos na curva, utiliza-se a aritmética das curvas para se codificar um ponto nesta, como veremos mais adiante.

5.3. OPERAÇÕES EM CURVAS ELÍPTICAS

A seguir estão dispostas as operações aritméticas executadas sobre as curvas elípticas. Estas foram escolhidas para criptografia, pois apresentam condições de um grupo Abeliano, do qual as propriedades aritméticas permitem retornar a um ponto através da adição de seu segundo operador e o resultado.

5.3.1. Adição de Pontos

Considere dois pontos distintos *P* e Q tal que $P = (x_P, y_P)$ e $Q = (x_Q, y_Q)$. Considere também o ponto resultante $L = P + Q$, onde $L = (x_L, y_L)$. A soma de dois

pontos resulta em um terceiro ponto na curva através do uso de uma reta *s*, que passando pelos dois pontos, resultará em um terceiro.

$$s = \left(\frac{(y_P - y_Q)}{(x_P - x_Q)}\right) mod\ p$$

A partir da reta *s* calculada acima, calculamos então os dois valores relativos ao ponto *L*, conforme segue:

$$x_L = s^2 - x_P - x_Q\ mod\ p$$

$$y_L = -y_P + s\ (x_P - x_L)\ mod\ p$$

Se $Q = -P$, ou seja, $Q = (x_P, -y_P\ mod\ p)$, então $P + Q = O$, onde *O* é o ponto no infinito.

Se $P = Q$, então $P + Q = 2P$, utilizando o método de multiplicação de pontos para cálculo de tal, que será abordada no tópico seguinte.

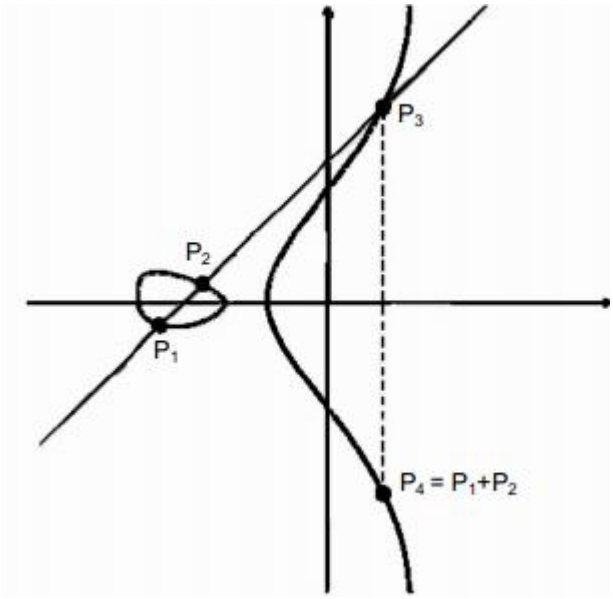

Figura 6 - Ilustração do processo de adição de dois pontos P1 e P2.
Fonte: BURNETT, Steve; PAINE, Stephe. (2002)

5.3.2. Subtração de Pontos

Considerando dois pontos *P* e Q tal que $P = (x_P, y_P)$ e $Q = (x_Q, y_Q)$, então $P - Q = P + (-Q)$, onde $-Q = (x_Q, -y_Q \ mod \ p)$. Embora não seja utilizada com frequência no cálculo das curvas elípticas, a subtração de pontos é de grande importância para a compreensão do inverso dos pontos na mesma teoria.

5.3.3. Multiplicação de Pontos

Quando se necessita adicionar um ponto a si mesmo, uma ou mais vezes, utiliza-se a fórmula de multiplicação de ponto. Esta fórmula é utilizada apenas para se somar um ponto a ele mesmo diversas vezes, e não para se multiplicar dois pontos diferentes entre si.

Considerando que $y_P \neq 0$, então $P + P = 2P = R$ tem x_R e y_R calculados da seguinte maneira:

$$x_R = \left(\frac{3x_P^2 + a}{2y_P}\right)^2 - 2x_P$$

$$y_R = \left(\frac{3x_P^2 + a}{2y_P}\right)(x_P - x_R) - y_P$$

Obtêm-se assim os pontos x e y de R, ponto qual é a multiplicação de P.

6. CRIPTOGRAFIA DE CURVA ELÍPTICA

De acordo com Koblitz (2008), a pesquisa da teoria dos números relacionada às curvas elípticas foi originalmente motivada por questões estéticas, isto é, não havia uma aplicação direta para tal. Mas nas décadas recentes tal motivação aumentou devido à importância da aplicação das curvas elípticas nas áreas de teoria de código, geração de números pseudorrandômicos e especialmente criptografia.

O primeiro uso das curvas elípticas em criptografia foi no algoritmo de fatoração de curvas elípticas de Hendrik Willem Lenstra, utilizado na denominação de pequenos fatores. Inspirados nesta aplicação incomum das curvas elípticas, em 1985 Neal Koblitz e Victor Miller independentemente propuseram o uso do grupo de pontos em uma curva elíptica definida sobre em corpo finito em sistemas criptográficos de logaritmo discreto, comumente denominada criptografia de curva elíptica (ECC, do Inglês *Elliptic Curve Cryptography*).

6.1. A UTILIZAÇÃO DAS CURVAS EM CRIPTOGRAFIA

Koblitz (2008) indica que a principal vantagem dos sistemas de ECC sobre os demais sistemas baseados em fatoração de inteiros ou no problema do logaritmo discreto no grupo multiplicativo de um corpo finito, como o RSA, é a ausência de um algoritmo de tempo subexponencial para se encontrar logaritmos discretos nesses grupos, sendo que a curva e o corpo sob ela sejam escolhidos apropriadamente.

A difícil resolução do problema das curvas elípticas dá-se, pois, considerando um limite de corpo de tamanho adequado, torna-se difícil descobrir os pontos utilizados para cálculo, dado que estes não são aleatórios e apresentam significado em relação mútua. Esta dificuldade difere-se de outros algoritmos criptográficos onde se obtém a chave apenas a partir de números de origem restrita, como, por exemplo, números primos.

Por consequência, pode-se utilizar um grupo de curva elíptica, que é menor em tamanho, mantendo o mesmo nível de segurança. Em diversas situações o

resultado é um conjunto de chaves de menor tamanho proporcionando uma economia de banda ao ser transmitido via rede, e processamento mais rápido, qualidades atrativas para aplicações de segurança em dispositivos onde o poder computacional e o espaço no circuito integrado são limitados, como em *smart cards* e telefones celulares.

Em 2005, a NSA (Agência de Segurança Nacional do Governo dos Estados Unidos) publicou um artigo no qual recomendavam que a indústria aproveitasse os avanços dos últimos 30 anos em Segurança da Informação e fizesse uso da ECC. Neste artigo indicavam o uso de chaves de tamanho médio que se equiparavam com chaves de tamanho grande em outros algoritmos, como o RSA.

6.2. PARÂMETROS DE DOMÍNIO

Segundo Stallings (2011), os parâmetros de domínio devem ser cuidadosamente selecionados para evitar uma combinação fraca de parâmetros. Porém, como tal escolha é feita aleatoriamente, os algoritmos tendem a aplicar certos conjuntos de regras a fim de limitar as escolhas aos parâmetros adequados. Embora a uma primeira análise isso pareça limitar as escolhas, os parâmetros evitam que valores que viriam a facilitar o trabalho de criptoanalistas sejam evitados, aumentando, assim, a segurança do algoritmo.

De acordo com Anoop (2001), os parâmetros para a curva elíptica em F_p são *p, a, b, G*, *n* e *h*. *p* é o número primo que define o corpo finito F_p. *a* e *b* são os parâmetros que definem a curva, aplicados à equação primária. *G* é o ponto gerador (x_G, y_G), um ponto escolhido na curva elíptica para operações criptográficas. *n* é a ordem (número de elementos) da curva elíptica. O ponto escolhido como constante para multiplicação de pontos é selecionado entre 1 e *n*-1. *h* é o cofator onde $h = \#E(F_P)/n$. $\#E(F_P)$ é o número de pontos na curva elíptica.

6.3. ESQUEMA CRIPTOGRÁFICO

O esquema criptográfico de chave pública utilizando-se Curvas Elípticas é composto de quatro passos:

1 – Um algoritmo de geração de parâmetro de domínio que gera um conjunto *D* de parâmetros;

2 – Um algoritmo de geração de chaves que aceita como parâmetro um conjunto *D* e gera pares de chaves *(Q, d)*.

3 – Um algoritmo criptográfico que aceita como parâmetros o conjunto *D*, a chave pública *Q*, uma mensagem em texto plano *m*, produzindo o texto cifrado *c*.

4 – Um algoritmo de decodificação que aceita como parâmetros o conjunto *D*, a chave privada *d*, o texto cifrado *c* e então resulta em *m*, a mensagem em texto plano original.

Assumindo que *D* é um conjunto de parâmetros válido, e *Q* é uma chave válida e associada com *D*, o algoritmo de decodificação sempre aceita como parâmetros *(D, d, c)* e resulta em *m* se *c* foi realmente gerado pelo algoritmo de codificação *(D, Q, m)*. Esta checagem é efetuada na comparação dos pontos compartilhados como chave pública. Se estes não tem a relação esperada, então as chaves são descartadas e um erro é retornado informando que a chave utilizada não tem relação com o conjunto original. Este é um mecanismo para evitar que ataques de força-bruta sejam facilmente aplicados, visto que os pontos necessitam estar relacionados e não simplesmente tentados a esmo, dificultando o cálculo para um ataque em tempo real.

De acordo com Stallings (2011), embora esta verificação tida como proteção faça parte do algoritmo aplicado, ela pode ser removida em um ambiente especialmente projetado por um criptoanalista. A desativação desta proteção não garante, no entanto, que um ataque obtenha sucesso utilizando-se pontos tentados a esmo, pois a teoria ainda espera que os pontos corretos sejam utilizados.

Utilizando o grupo de pontos de uma curva elíptica definida sobre um corpo finito ao invés do grupo multiplicativo de um corpo finito. Dado *E* como a seguinte equação de *Weierstrass:*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_6x + a_6,$$

com $a_i \in F_q$. Os grupos utilizados na ECC são os subgrupos de ordem prima *G* dos pontos F_q de *E*. No conjunto da lei da curva elíptica, que são costumeiramente escritos na notação aditiva, o problema do logaritmo discreto pede: dado *P*, *Q* ∈ *G*, encontre $x \pmod n$, tal que $Q = xP$.

Nos primeiros anos da ECC uma escolha popular de curvas para fins demonstrativos era sobre a equação $y^2 = x^3 - x$ definida sobre o corpo primo F_p. Se $p \equiv 3 \ (mod\ 4)$, conhecido como caso *super singular*, é fácil demonstrar que a ordem do grupo é *p + 1*. Pode-se então encontrar facilmente um *p* que tenha então um subgrupo de ordem prima muito grande. Segundo Koblitz (2008), em curvas elípticas escolhe-se um primo *n* para o qual *p = 4n – 1* é primo; então o grupo de pontos F_p na equação $y^2 = x^3 - x$ é o produto do grupo de 4 pontos de ordem 2 e um subgrupo de ordem prima $n = \frac{(p+1)}{4}$.

Apesar de conveniente, o exemplo acima foi depois descartado, pois em 1991 fora provado que o problema do logaritmo discreto em uma curva super singular é de muito mais fácil resolução que em curvas convencionais. Isto acontece porque, definido o primo que gera um curva super singular, um atacante pode mensurar quais são os restos modulares aos quais serão aplicados. De acordo com Elkies (1987), existem exatamente 15 primos super singulares, sendo eles: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 41, 47, 59 e 71.

De todas as curvas definidas sobre números primos, as que apresentam esta característica representam uma pequena porção, porém seu uso frequente em demonstrações determinou uma importância errônea sobre seu uso. “Uma curva selecionada aleatoriamente tem a probabilidade de ser super singular em $O(\frac{1}{\sqrt{p}})$.” (KOBLITZ, 2008), representando assim uma baixa probabilidade quando se utilizam parâmetros adequados, como tamanho de chave suficiente e consequentemente um número primo de grande ordem utilizado como limite.

6.4. ALGORITMOS

Neste trabalho será considerado o algoritmo proposto por *Koblitz* e *Miller* aplicado sobre protocolo do tipo ECDH – *Elliptic Curve Diffie-Hellman* (*Diffie-Hellman* para troca de chave, registrado como NIST 800-56A) e ECIES - *Elliptic Curve Integrated Encryption Scheme* (AES utilizando pontos de Curva Elíptica, da *Certicom*, ainda não transformado em padrão). Estes algoritmos provêm mecanismos de troca de chaves para outro esquema criptográfico à escolha, no caso do ECDH, ou o sistema criptográfico completo, como no caso do ECIES.

6.4.1. ECDH

O protocolo de acordo de chave ECDH, uma variante do protocolo Diffie-Hellman aplicado em chaves públicas, foi idealizado para que permitisse duas pessoas em posse de pares de chaves públicas e privadas estabelecerem um segredo privado em um ambiente não seguro. Este segredo compartilhado pode então ser utilizado como chave ou, melhor ainda, como um número para se derivar uma nova chave, codificando as mensagens subsequentes com uma cifra de chave simétrica.

Utilizando o caso hipotético apresentado por Stallings (2011), suponhamos que Alice queira estabelecer um segredo compartilhado com Bob, mas o único meio de transporte disponível pode ser interceptado. Inicialmente, os parâmetros de domínio *(p, a, b, G, n, h)* devem ser acordados mutuamente. Além disso, cada um deles deve possuir um conjunto de chaves públicas e privadas, sendo a chave privada *d*, um inteiro selecionado aleatoriamente no intervalo *[1, n-1]* e uma chave pública *Q*, onde $Q = dG$. O par de Alice então é (d_A, Q_A) e de Bob (d_B, Q_B).

Efetua-se a troca das chaves públicas e Alice então calcula $(x_k, y_k) = d_A Q_B$. Bob calcula $(x_k, y_k) = d_B Q_A$. O segredo compartilhado é x_k. Alguns protocolos estudam utilizar uma chave simétrica derivada de x_k usando uma função de sumário

criptográfico (*hash*). O segredo calculado pelos dois é igual, afinal $d_AQ_B = d_Ad_BG = d_Bd_AG = d_BQ_A$. O protocolo secundário então pode utilizar este segredo compartilhado para codificar as mensagens (utilizando um esquema de chave simétrica, já que ambos possuem a mesma chave) ou derivar outra chave a partir deste.

A única informação exposta à rede insegura foi, nesse caso, a chave pública de ambos e os parâmetros de domínio. Em posse destas informações, continua-se impraticável o cálculo da chave privada sem a resolução total do problema do logaritmo discreto em curvas elípticas, que é muito custoso.

As chaves podem ser tanto estáticas (através de *PKIs*, uma estrutura de distribuição de chaves públicas) quanto efêmeras. Apesar de apresentar um método seguro de transmissão, o ECDH não garante proteção contra ataques *man-in-the-middle*. Tal proteção só seria garantida com o uso de um esquema de assinatura digital através de certificados, uma vez que a chave poderia ser compartilhada assim como a assinatura.

Devido ao seu caráter experimental, nenhum algoritmo de grande porte chegou a utilizar o protocolo de acordo de chaves ECDH como base de geração de suas chaves secretas, porém alguns deles apresentam grande fama nos círculos de pesquisa, sendo utilizados como métodos inovadores de codificação de dados.

Podemos citar os *drafts* ANSI X9.42 e X9.63-2011 como os mais conhecidos pilotos de algoritmos utilizando ECDH, ainda em desenvolvimento e pendendo oficialização para distribuição.

6.4.2. ECIES

O algoritmo ECIES (*Elliptic Curve Integrated Encryption Scheme*) foi desenvolvido pela empresa Certicom no ano de 2001 e apresenta até o momento um ciclo ativo de desenvolvimento, uma vez que novos paradigmas são descobertos através da análise da aritmética das curvas elípticas. É o único esquema criptográfico ou algoritmo que engloba a geração de chaves e a codificação /

decodificação de mensagens.

O tamanho de chaves varia de acordo com as normas ditadas pela Certicom (2000), desenvolvedora do algoritmo, entre os 112, 128, 160, 192, 224, 256, 384 e 521 bits. Segundo a empresa, limitar o tamanho da chave a estes bits garante um nível de segurança e compatibilidade com diversos tipos de requerimentos, dependente da força da arquitetura desejada.

6.4.2.1. *Padronização de parâmetros de domínio*

A empresa Certicom (2000) dita quais os parâmetros de domínio devem ser utilizados para cada escolha de tamanho de chave. O exemplo a seguir é baseado no documento oficial da empresa, utilizado para basear o algoritmo. Estes parâmetros são aplicados à curva gerada pela equação de *Weierstrass* com um tamanho de chave de 112 bits:

p = DB7C 2ABF62E3 5E668076 BEAD208B = (2^{128} - 3) / 76439 = 4.451.685.225.093.714.772.084.598.273.548.427

a = DB7C 2ABF62E3 5E668076 BEAD2088 = 62.142.497.755.811.880.000

b = 659E F8BA0439 16EEDE89 11702B22 = 1.652.502.793.576.000.300

G = 020948 7239995A 5EE76B55 F9C2F098

n = DB7C 2ABF62E3 5E7628DF AC6561C5 = 108.906.765.203.850.760.000

h = 01 = 1

6.4.2.2. *Geração do par de chaves*

A função de geração de pares de chave é bastante simples. Recebe-se um conjunto válido de parâmetros de domínios da curva elíptica T = (p, a, b, G, n, h) e então se realiza as seguintes operações:

1 – Randomicamente selecionar um inteiro d do intervalo [*1, n-1*].

2 – Calcular $Q = dG$.

3 – Q representa a chave pública e d, a chave privada.

Dessa maneira, obtém-se um conjunto de pares como chave pública, que vai ser utilizado pelo destinatário para codificar a mensagem, e um número inteiro como chave privada, que é mantido pelo remetente da mensagem e então utilizado para se reverter o ponto para a mensagem original.

6.4.2.3. *Processo criptográfico*

O processo criptográfico envolve um processo de transformação da *string* de bits em um ponto no gráfico, utilizando um processo de conversão para octeto, e de octeto então para um ponto na curva (P_m).

Em posse dos parâmetros de domínio, da chave pública do destinatário (P_B), do par chave pública e privada local e do ponto representando a mensagem, para que A consiga enviar uma mensagem para B, A deve calcular:

$$C = (kG, P_m + kP_B).$$

O processo de decodificação trata-se do cálculo inverso, ou seja:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m.$$

A codificou a mensagem ao adicionar kP_B a ela. Ninguém exceto A sabe o valor de k, então mesmo que P_B seja a chave pública, ninguém consegue remover a máscara kP_B. No entanto, A incluiu uma pista, informação o suficiente para alguém em posse da chave privada n_B remover. Para um atacante recuperar a mensagem interceptada, ele primeiro teria de calcular k dado kG, o que se assume ser de grande complexidade.

Um exemplo de processo criptográfico, citado por Koblitz (2008), se dá como segue: define-se $p = 751$, $E_P(-1, 188)$ que equivale à equação $y^2 = x^3 - x + 188$; e $G =$

(0, 376). Suponhamos que A queira enviar uma mensagem para B que foi codificada em um ponto na curva P_m = *(562, 201)*. Temos também *386(0, 376) = (676, 558)*, e *(562, 201) + 386(201, 5) = (385, 328)*. Então A envia para B o texto codificado *{(676, 558), (385, 328)}*.

6.4.3. ECDSA

O algoritmo ECDSA (Elliptic Curve Digital Signature Algorithm), patenteado pela empresa Certicom e definido como padrão por ANSI X9.62 ECDSA, é um sistema de assinatura digital baseado no DSA (Digital Signature Algorithm). Este algoritmo originalmente empregava o RSA como método criptográfico.

De acordo com a empresa Certicom (2000), uma assinatura digital é um esquema de criptografia de chave pública onde se utiliza um conjunto de chaves apenas para se codificar um escrutínio (hash) de uma mensagem para eventual transmissão. Depois de calculado o hash da mensagem, a chave pública destinada a esse fim era aplicada ao hash e este então era transmitido logo após o conteúdo da mensagem.

A mensagem pode ou não estar criptografada, constituindo de dois cenários dependentes desta possibilidade, um em que a mensagem segue codificada, denominado autenticação com segredo e um segundo onde apenas o hash segue criptografado, denominado autenticação de texto plano.

6.5. SEGURANÇA DA ECC

A segurança da ECC, de acordo com a empresa Certicom (2000), depende da dificuldade de se calcular *k* dado *kP* e *P*. Este é o chamado problema do logaritmo de curva elíptica. O meio mais rápido de se resolver o problema do logaritmo de curva elíptica é o chamado método *Pollard rho*. A tabela a seguir compara vários algoritmos mostrando uma relação entre tamanhos de chaves em termos de esforço

computacional para criptoanálise.

Tabela 1 - Comparação de tamanhos de chaves em relação ao esforço computacional para criptoanálise.

Esquema Simétrico (tamanho da chave em *bits*)	Esquema baseado em ECC (tamanho de *n* em *bits*)	RSA/DSA (tamanho do módulo em *bits*)
56	112	512
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	512	15360

Fonte: Certicom (2000).

Nota-se que uma chave de tamanho consideravelmente menor pode ser usado em ECC comparado com o RSA. Além disso, para chaves de mesmo tamanho, o esforço computacional para criptoanálise da ECC e do RSA são comparáveis.

6.6. APLICAÇÕES DA ECC

Uma pesquisa realizada por Cheng (2004), diversos produtos empregam o uso de algoritmos variantes de ECC em seus mecanismos, tenham seu foco em segredo ou apenas para geração de sumário criptográfico (hash). Algumas bibliotecas de código aberto também oferecem esquemas criptográficos baseados em ECC. Estas bibliotecas tem seu uso difundido entre diversos tipos de produtos, porém por contar com um leque maior de opções que apresentam retro compatibilidade, acabam por empregar outros algoritmos não baseados em ECC.

Este cenário tende a mudar com o decorrer dos anos, uma vez que a

comunidade desenvolvedora precisa de alguns anos de validação de um algoritmo antes de sua difusão. Esta confiança só surge depois de milhares de testes realizados nos algoritmos, provando sua força característica e sua resistência a ataques.

A seguir, abordamos alguns produtos e bibliotecas que empregam algoritmos baseado em ECC em suas opções:

- OpenSSH: Serviço de código-aberto para acesso a terminal remoto para sistemas Unix e Linux, utiliza desde a versão 5.7 (24 de Janeiro de 2011) o algoritmo de assinatura digital ECDSA para autenticação de hosts e de assinatura de mensagens e possibilita a escolha de segredo utilizando o algoritmo ECDH para troca de chaves, aplicadas em AES.

- OpenSSL: Biblioteca de código-aberto de esquemas criptográficos em C, provê em suas opções desde a versão 0.9.8 (05 de Julho de 2005) os algoritmos de assinatura digital ECDSA e algoritmos de geração de chaves baseados em ECDH.

- Crypto++: Biblioteca de código semiaberto com funções criptográficas em C++, oferece os algoritmos ECDSA, ECNR, ECIES, ECDH e ECMQV.

- Oracle iPlanet Web Server: Servidor web com habilidade de integração com aplicações Java, possui suporte a algoritmos ECC utilizados em aplicações.

- RSA BSAFE: Biblioteca para desenvolvimento de aplicações.

7. DEMONSTRAÇÃO PRÁTICA

No exemplo prático preparado para esta apresentação, uma biblioteca para fins educativos e experimentais disponibilizada pelo sistema *Github* pelo usuário *bellbind* será utilizada para se demonstrar o cálculo de pontos de curvas elípticas, tomando um ponto referente a uma mensagem, realizando o processo de codificação e decodificação, mostrando passo a passo os valores processados.

O código fonte foi disponibilizado em dois arquivos, utilizando a linguagem *Python* na versão *2.7* para sua interpretação, nomeados *ecc.py* e *pn.py*. O primeiro contém o código do programa, enquanto o segundo apresenta funções utilizadas no cálculo dos módulos sob as curvas elípticas.

Esta biblioteca tem sua licença sob a GPL e foram utilizadas as chamadas Liberdade 0 (a liberdade de executar o programa, para qualquer propósito) e Liberdade 1 (a liberdade de estudar como o programa funciona e adaptá-lo para as suas necessidades), mantendo, porém, os direitos do autor por forma a não permitir que essa informação seja usada de uma maneira que limite as liberdades originais.

As configurações dos parâmetros de domínio, assim como os valores utilizados para as chaves, podem ser tanto definidas diretamente no código quanto manualmente, de acordo com opções de configuração do mesmo. Os valores utilizados, tomando o propósito dos fins educativos de tal execução, estão bem aquém dos padrões ideais propostos pela empresa Certicom (2000), porém sua eficiência computacional não permitiria que tal demonstração fosse viável.

O intuito do programa é, depois de receber os parâmetros de domínio, gerar um conjunto de três pares de chaves utilizando o algoritmo ECDH, e então derivar um segredo compartilhado entre as duas partes, em uma combinação entre os três pares. Além disso, aplica o algoritmo de codificação de El Gammal para se codificar e decodificar um ponto na curva, representando um texto plano.

8. CONCLUSÃO

A criptografia, provinda da necessidade de se comunicar secretamente através de longa distância, é uma necessidade inegável nos dias atuais. A utilização cada vez maior de computadores para tarefas envolvendo informações sensíveis como dados pessoais e financeiros torna seu uso indispensável para a tecnologia moderna.

Através da análise das principais características dos métodos criptográficos de chave simétrica (ou de chaves privadas) e chave assimétrica (ou de chaves públicas), nota-se um caráter de complementaridade entre ambos.

Enquanto a criptografia simétrica é mais indicada para grandes volumes de informações pelo desempenho dos algoritmos empregados, a criptografia assimétrica se baseia na necessidade de se transmitir mensagens através de um meio inseguro, como a Internet.

A criptografia de chaves assimétricas também pode prover um meio de transporte para as chaves simétricas, que serão então aplicadas em seu destino final. Tais podem ser diferentes a cada mensagem, complementando o uso da criptografia de acordo com seus objetivos básicos.

Além da criptografia em rede, a criptografia de curvas elípticas também é aplicada em dispositivos de baixo poder de processamento, como telefones celulares, *smart cards*, *pagers*, etc.

Embora a criptografia de curva elíptica seja superior no quesito de força de algoritmo quando se compara o tamanho da chave com o esforço computacional necessário para quebra-la, esta ainda necessita de muito estudo e maturação para ganhar a confiança do público em geral e ser adotada como um protocolo de criptografia de larga escala.

É importante lembrar que a criptografia simétrica não substitui a assimétrica e vice-versa. É de suma importância reconhecer e identificar as limitações de cada método, buscando formas de utiliza-los de forma complementar para prover segurança às partes afetadas.

REFERÊNCIAS BIBLIOGRÁFICAS

ANOOP, M.; **Elliptic Curve Cryptography – An Implementation Tutorial**. Artigo pela empresa Tata Elxsi Ltd, de Março de 2003.

BELLBIND; **[python]basics of elliptic curve Cryptography**, github. < https://gist.github.com/1414867 >. Acessado em 30 de Abril de 2012.

BLAKE, I.; SEROUSSI, G.; SMART, N., **Advances in Elliptic Curve Cryptography**, London Mathematical Society 317, Cambridge University Press, 2005.

BLAKE, I.; SEROUSSI, G.; SMART, N., **Elliptic Curves in Cryptography**, London Mathematical Society 265, Cambridge University Press, 1999.

CAETANO, Paulo; 2004. **Linha do tempo da criptografia.** Disponível em < http://www.dm.ufscar.br/~caetano/iae2004/G6/linhadotempo.htm >. Acesso em Maio, 2012.

CALLAS, J.; **An Introduction to Cryptography**, 2008, PGP Corporation.

CERTICOM Research; **Recommended Elliptic Curve Domain Parameters**, artigo publicado em 20 de Setembro de 2000.

CERTICOM Research; **Standards for efficient cryptography**, artigo publicado em 20 de Setembro de 2000.

CHENG, Z; **Simple Tutorial on Elliptic Curve Cryptography**, artigo publicado em Dezembro de 2004.

ECC Brainpool; Standards Curves and Curve Generation, artigo publicado em 19 de Outubro de 2005.

ELKIES, N.; **Inventiones Mathematicae**, Springer-Verlug, 1987.

ELLSBURY, Graham; 1998. **Description of the Enigma.** Disponível em < http://www.ellsbury.com/enigma2.htm >. Acesso em Maio, 2012.

IETF (Internet Engineering Task Force); **RFC 4492 - Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)**, 2006.

IETF (Internet Engineering Task Force); **RFC 5480 - Elliptic Curve Cryptography Subject Public Key Information**, 2009.

IETF (Internet Engineering Task Force); **RFC 6090 - Fundamental Elliptic Curve Cryptography Algorithms**, 2011.

KAHN, D. **The Codebreakers**. Macmillan, 1967.

KOBLITZ, N.; KOBLITZ, H.; MENEZES, A. **Elliptic Curve Cryptography: The Serpentine Course of a Paradigm Shift**. Artigo independente. 2008.

MALHOTRA, K.; GARDNER, S.; PATZ, R.; **Implementation of Elliptic-Curve Cryptography on Mobile Healthcare Devices, Networking, Sensing and Control**, 2007 IEEE International Conference on, London, 15–17 April 2007

MATHWORLD; 2010. **Elliptic Curve definition.** Disponível em < http://mathworld.wolfram.com/EllipticCurve.html >. Acesso em Maio, 2012.

MENEZES, A., HANKERSON, D., VANSTONE, S.; **Guide to Elliptic Curve Cryptography**, Springer, 2003.

NIST – National Institute of Standards and Technology; **Recommended Elliptic Curves for Federal Government use**. Artigo de Julho de 1999.

NSA, National Security Agency, **The case for elliptic curve cryptography**, 2009. Disponível em < https://www.nsa.gov/business/programs/elliptic_curve.shtml >. Acessado em 30 de Março de 2012.

PELZL, J.; PAAR, C.; **Elliptic Curve Cryptosystems**, Springer, 2009.

SINGH, S. **O Livro dos Códigos**, Editora Record, 2004.

STALLINGS, W. **Cryptography and Network Security – Principles and Practices 5th edition**, Pearson. 2011.

TKOTZ, V. **Criptologia**. Disponível em: <http://www.numaboa.com.br/criptografia>. Acesso em: 17 de Março de 2012.

WASHINGTON, L.; **Elliptic Curves: Number Theory and Cryptography**, Chapman & Hall / CRC, 2003.

WOLFRAMALPHA; 2012. **Elliptic Curve representation.** Disponível em < http://www.wolframalpha.com/input/?i=y%C2%B2+%3D+x%C2%B3+%E2%88%92+x >. Acesso em Maio, 2012.

WOLFRAMALPHA; 2012. **Elliptic Curve representation.** Disponível em < http://www.wolframalpha.com/input/?i=y%C2%B2+%3D+x%C2%B3+%E2%88%92+x +%2B+1 >. Acesso em Maio, 2012.