UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE TECNOLOGIA
DEPARTAMENTO DE ELETRICIDADE
MESTRADO EM TELECOMUNICAÇÕES

Comunicação de Dados

Criptografia Baseada nas Curvas Elípticas

Cícero Hildenberg Lima de Oliveira

CURITIBA – PR – Outubro – 2001

Por

Cícero Hildenberg Lima de Oliveira

Avaliação apresentada ao Prof. Dr. Eduardo Parente Ribeiro, como requisito avaliativo para a disciplina TE727 **– Comunicação de Dados.**

CURITIBA–PR – SETEMBRO – 2001

"…. very little do we have and inclose which we can call our own in the deep sense of the word. We all have to accept and learn, either from our predecessors or from our contemporaries. Even the greatest genius would not have achieved much if he had wished to extract everything from inside himself. But there are many good people, who do not understand this, and spend half their lives wondering in darkness with their dreams of originality. I have known artists who were proud of not having followed any teacher and of owing everything only to their own genius. Such fools!"

[Goethe, Conversations with Eckermann, 17.2.1832]

Introdução

A historia da Criptografia é longa e fascinante. A criptografia é uma das profissões mais antigas, com cerca de 4000 anos. Os egípcios já utilizavam a criptografia para cifrar alguns de seus hieróglifos, Os romanos utilizavam códigos secretos para comunicar planos de batalha. O mais interessante é que a tecnologia de criptografia não mudou muito até meados deste século, existe também o barro de *Phaistos* (1600 a.c) que ainda não foi decifrado. Depois da Segunda Guerra Mundial, com a invenção do computador, a área realmente floresceu incorporando complexos algoritmos matemáticos. Durante a guerra, os ingleses ficaram conhecidos por seus esforços para decifração de códigos. Na verdade, esse trabalho criptográfico formou a base para a ciência da computação moderna.

A proliferação de computadores e sistemas de comunicações nos anos 60 trouxe com isto uma demanda do setor privado para meios de proteger informações que circulam em forma digital e para promover os serviços de segurança. Começando com o trabalho de Feistel a IBM em meados dos anos 70 e culminando em 1977 com a adoção como um Padrão de processamento de informação Federal norte-americano por codificar informação não classificada, DES (Data Encryption Standard), os Dados Padrão de Encriptação, é o mecanismo mais mais famoso criptação da história. Permanece os meios standards por afiançar comércio eletrônico para muitas instituições financeiras ao redor do mundo. Themost desenvolvimento notável na história de cryptographycame em 1976whenDiffie e Hellman publicou Direções.

Com o aumento vertiginoso das redes de computadores e seu proporcional uso por organizações para viabilizar e controlar os seus negócios e principalmente a afirmação cada vez mais da Internet, ao mesmo tempo se criou a suprema necessidade de proteger melhor as informações, pois no mercado competitivo, elas são muito valiosas, melhorando os mecanismos para implantar e proporcionar meios de segurança mais edificáveis e confiáveis.

Uma forma de proteger e com isso evitar o acesso impróprio às informações sigilosas é por meio da codificação ou cifragem da informação, conhecida como criptografia, fazendo com que apenas as pessoas às quais estas informações são destinadas, consigam compreendê-las. A criptografia fornece técnicas para codificar e decodificar dados, tais que os mesmos possam ser armazenados, transmitidos e recuperados sem sua alteração ou exposição. Em outras palavras, técnicas de criptografia podem ser usadas como um meio efetivo de proteção de informações suscetíveis a ataques, estejam elas armazenadas em um computador ou sendo transmitidas pela rede. Seu principal objetivo é prover uma comunicação segura, garantindo serviços básicos de autenticação, privacidade e integridade dos dados.

A palavra criptografia tem origem grega (Gr. *kryptós*, oculto + *graph*, r. de *graphein*, escrever, grafia) e define a arte ou ciência de escrever em cifras ou em códigos, utilizando um conjunto de técnicas que torna uma mensagem incompreensível, chamada comumente de texto cifrado, através de um processo chamado cifragem, permitindo que apenas o destinatário desejado consiga decodificar e ler a mensagem com clareza, no processo inverso, a decifragem.

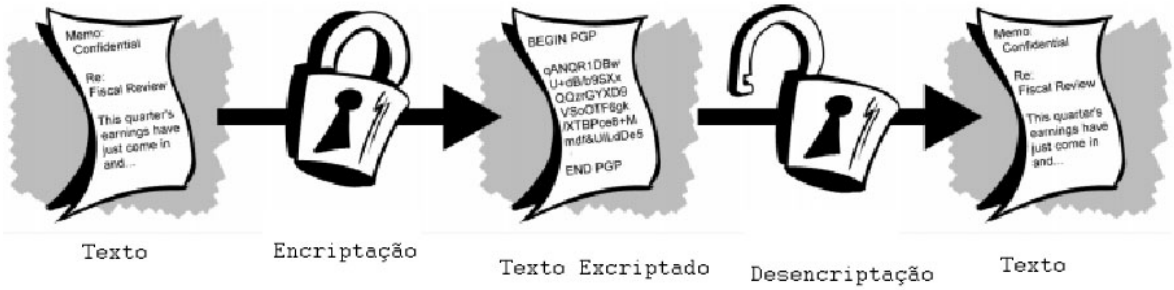
Criptografia é a arte de escrever em cifra ou em código, composto de técnicas que permitem tornar incompreensível, com observância e normas especiais consignadas numa cifra ou num código, o texto e uma mensagem escrita com clareza. A definição mais clara para criptografia, em nosso tempo moderno, é o estudo de técnicas matemáticas relacionado

a aspectos de informação segurança como confidência, integridade de dados, autenticação de entidade, e origem de dados autenticação.

Com o advento da informática, com a sua alta velocidade de processamento, as funções de criptografia ficaram mais complexas.

Encriptação e Desencriptação

Dados que podem ser lidos e compreendidos sem qualquer medida especial é chamado Texto plano ou texto limpo. O método de disfarçar o texto de tal modo que se esconda a sua substância lógica é chamada encriptação. O resultado do texto quando o mesmo é cifrado é uma geringonça ilegível chamado texto Encriptado ou texto cifrado. Usamos a encriptação para assegurar o conteúdo da informação. O processo de reverter o texto encriptado para o texto normal é chamado decriptação.



"There are two kinds of cryptography in this world: cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files. This book is about the latter."

--Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C.

Exemplo:

Suponha que marcos S. e Suzana L. são dois agentes secretos que querem se comunicar usando um código, pois suspeitam que seus telefones estão grampeados e que suas cartas estão sendo interceptadas. Em particular, Marcos quer mandar a seguinte mensagem para Suzana.

ENCONTRO AMANHÃ

Usando o esquema de substituição dado acima, Marcos envia a seguinte mensagem:

5 14 3 15 14 20 18 15 1 13 1 14 8 1 14

(Onde o Ã foi substituído por AN). Um código desse tipo pode ser quebrado sem muita dificuldade por uma série de técnicas, incluindo a análise de freqüência de letras. Para dificultar a quebra do código, os agentes procedem seguinte maneira: em primeiro lugar, ao aceitar a missão, eles escolheram uma matriz 3x3,

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 0 & 1 & 2 \end{bmatrix}$$

Marcos, então, separa a mensagem em cinco vetores R^3 (caso isso não fosse possível, adicional letras extras).
Temos, então, os vetores

$$\begin{bmatrix} 5 \\ 14 \\ 3 \end{bmatrix}, \begin{bmatrix} 15 \\ 14 \\ 20 \end{bmatrix}, \begin{bmatrix} 18 \\ 15 \\ 1 \end{bmatrix}, \begin{bmatrix} 13 \\ 1 \\ 14 \end{bmatrix}, \begin{bmatrix} 8 \\ 1 \\ 14 \end{bmatrix}.$$

Marcos, agora, usa a transformação linear $L : R^3 \to R^3$ dada por $L(x) = Ax$, de modo que a mensagem fica:

$$A = \begin{bmatrix} 5 \\ 14 \\ 3 \end{bmatrix} = \begin{bmatrix} 42 \\ 25 \\ 20 \end{bmatrix}, \qquad A = \begin{bmatrix} 15 \\ 14 \\ 20 \end{bmatrix} = \begin{bmatrix} 103 \\ 69 \\ 54 \end{bmatrix}$$

$$A = \begin{bmatrix} 18 \\ 15 \\ 1 \end{bmatrix} = \begin{bmatrix} 51 \\ 35 \\ 17 \end{bmatrix}, \qquad A = \begin{bmatrix} 13 \\ 1 \\ 14 \end{bmatrix} = \begin{bmatrix} 57 \\ 42 \\ 29 \end{bmatrix}$$

Portanto, Marcos envia a mensagem

42 25 20 103 69 54 51 35 17 57 42 29 52
37 29

Suponha, agora, que Marcos recebe a seguinte mensagem de Suzana:

43 30 14 101 67 53 96 61 55 83 58 43 40
25 24 90 56 53

que ele quer decodificar com a matriz A dada acima. Para decodifica-la, Marcos divide a mensagem em seis vetores em R^3:

$$\begin{bmatrix} 43 \\ 30 \\ 14 \end{bmatrix}, \begin{bmatrix} 101 \\ 67 \\ 53 \end{bmatrix}, \begin{bmatrix} 96 \\ 61 \\ 55 \end{bmatrix}, \begin{bmatrix} 83 \\ 58 \\ 43 \end{bmatrix}, \begin{bmatrix} 40 \\ 25 \\ 24 \end{bmatrix}, \begin{bmatrix} 90 \\ 56 \\ 53 \end{bmatrix}$$

e resolvendo a equação

$$L(x_1) = \begin{bmatrix} 43 \\ 30 \\ 14 \end{bmatrix} = Ax_1$$

para x_1. Como A é invertível,

$$x_1 = A^{-1}\begin{bmatrix}43\\30\\14\end{bmatrix} = \begin{bmatrix}0 & 1 & -1\\2 & -2 & -1\\-1 & 1 & 1\end{bmatrix}\begin{bmatrix}43\\30\\14\end{bmatrix} = \begin{bmatrix}16\\12\\1\end{bmatrix}$$

Analogamente,

$$x_2 = A^{-1}\begin{bmatrix}101\\67\\53\end{bmatrix} = \begin{bmatrix}14\\15\\19\end{bmatrix}, \qquad x_3 = A^{-1}\begin{bmatrix}96\\61\\55\end{bmatrix} = \begin{bmatrix}6\\15\\20\end{bmatrix},$$

$$x_4 = A^{-1}\begin{bmatrix}76\\48\\40\end{bmatrix} = \begin{bmatrix}8\\16\\12\end{bmatrix}, \qquad x_5 = A^{-1}\begin{bmatrix}86\\53\\52\end{bmatrix} = \begin{bmatrix}1\\14\\19\end{bmatrix}$$

Usando a correspondência entre letras e números, Marcos recebeu a seguinte mensagem:

Planos Fotográficos

Criptoanálise

A Criptoanálise, do grego: "kryptós" (oculto) e "analyein" (desfazer), é a ciência que abrange os princípios, métodos e meios para se chegar a decriptação de um criptograma, sem prévio conhecimento dos códigos ou cifras empregados na produção do texto cifrado. A Criptografia e a Criptoanálise, compõem a Criptologia ("kryptós" + "lógos" – palavra).

Existem várias técnicas de criptoanálise que podem ser usadas para quebrar sistemas criptografados. Uma delas, o ataque da força bruta (busca exaustiva da chave), consiste na média do teste da metade de todas as chaves possíveis de serem usadas.

Outra técnica é a criptoanálise diferencial uma potente técnica de criptoanálise. A idéia básica consiste em estipular textos planos e criptografá-los. Ela analisa o efeito de diferenças particulares em pares de textos planos com as diferenças de pares de textos cifrados resultantes. Estas diferenças usam probabilidades para às possíveis chaves, de forma a localizar a chave mais provável.

As mensagens que legíveis são chamadas de texto plano ou limpo. E as ilegíveis, são chamadas de texto cifrado.

Encriptar significa que a mensagem legível (texto plano) será transformada em uma mensagem ilegível (texto cifrado). A função de decriptar é o processo inverso da encriptação, a partir de um texto cifrado, obtêm-se o texto plano. Para a criptografia de textos planos, utiliza-se uma chave(senha), que junto com o algoritmo de criptografia, irá codificar e decodificar os textos. A mesma chave simétrica serve para criptografar e decriptografar.

A criptografia utiliza conceitos matemáticos para a construção de seus algoritmos criptográficos. Assim, na figura acima estão representados os símbolos matemáticos que são adotados.

Um texto cifrado fica representado da seguinte maneira:

$Y = E_K(X)$, ou seja um texto plano (X) é encriptado por um algoritmo que contém uma chave (senha) K;
$X = D_K(Y)$, para a obtenção do texto plano inicial, deve-se submeter o texto cifrado (Y) ao mesmo algoritmo e a mesma chave (K).

Onde:
•X - Texto Plano
•Y - Texto Cifrado
•E - Encriptar
•D - Decriptar
•K - Chave
$Y = E_K(X)$
$X = D_K(Y)$

Um sistema para afiançar um mínimo de segurança, ele deve fornecer os seguintes serviços:

Confidencialidade ou sigilo: garantia de que somente as pessoas ou organizações envolvidas na comunicação possam ler e utilizar as informações transmitidas de forma eletrônica pela rede;
Integridade: garantia de que o conteúdo de uma mensagem ou resultado de uma consulta não será alterado durante seu tráfego;
Autenticação: garantia de identificação das pessoas ou organizações envolvidas na comunicação;
Não-Repúdio (Não recusa): garantia que o emissor de uma mensagem ou a pessoa que executou determinada transação de forma eletrônica, não poderá, posteriormente negar sua autoria.

Métodos

O método de Esteganografia consiste na existência de uma mensagem escondida dentro de outra mensagem. Por exemplo, uma seqüência de letras de cada palavra pode formar a palavra de uma mensagem escondida.

Algumas formas de esteganografia:
Marcação de caracteres: utilização de uma tinta com composto diferente que ao coloca-la defronte a luz estes caracteres ficam de forma diferente, compondo a mensagem secreta.
Tinta invisível: pode-se utilizar uma tinta invisível para a escrita da mensagem em cima de outra pré-existente, aonde, somente com produtos químicos poderíamos obter o conteúdo.
Outra forma seria a utilização de furos no papel em letras selecionadas, habitualmente visível somente quando colocada defronte uma lâmpada.

A moderna Esteganografia utiliza o uso de bits não significativos que são concatenados a mensagem original e faz uso também de área não usada.

O princípio de Kerckhofss consiste em retirar o segredo do algoritmo e passar para uma chave. Essa chave "programa" é o algoritmo. É ela de seleciona qual das possíveis transformações será usada.

Sistemas Criptográficos

Criptossistemas podem ser tanto quando assimétricos. Num criptossistema simétrico a encriptação e a decriptação são feitas com uma única chave, ou seja, tanto o remetente quanto o destinatário usam a mesma chave. Num sistema assimétrico, ao contrário, duas chaves são empregadas. Em criptossistemas de uma chave, como por exemplo o DES (Data Encription Standart), ocorre o chamado "problema de distribuição de chaves". A chave tem de ser enviada para todos os usuarios autorizados antes que mensagens possam ser trocadas. Isso resulta num atraso de tempo e possibilita que a chave chegue a pessoas não autorizadas.

Criptossistemas assimétricos, ou de dua chaves, contormam o problema da distribuição de chaves através do uso de chaves públicas. A criptografia de chaves públicas foi inventada em 1976 por Whitfield diffie e Martin Hellman a fim de resolver o problema da distribuição de chaves. No novo sistema, cada pessoa tem um par de chaves chamadas : chave pública e chave privada. A chave pública é divulgada enquanto que a chave privada e deixada em segredo. Para mandar uma mensagem privada, o transmissor encripta a mensagem usando a chave públicado destinatário pretendido.

Segue um exemplo de como o sistema funciona.

Quando Ana quer mandar uma mensagem para Carlos, ela procura a chave pública dele em um diretório, a usa para encriptar a mensagem, e a envia. Carlos então usa a sua chave privada para decriptar a mensagem e lê-la. Este sistema também permite a autenticação digital de mensagens , ou seja é possível prover certeza ao receptor sobre a identidade do transmissor e sobre a integridade da mensagem. Quando uma mensagem é encriptada com uma chave privada, ao invés da chave pública; o resultado é uma assinatura digital, ou seja, uma mensagem que só uma pessoa poderia produzir, mas que todos possam verificar. Normalmente autenticação se refere ao uso de assinaturas digitais : a assinatura é um conjunto inforjável de dados assegurando o nome do autor ou funcionando como uma assinatura de documentos , ou seja, que determinada pessoa concordou com o que estava escrito. Isso também evita que a pessoa que assinou a mensagem depois possa se livrar de responsabilidades, alegando que a mensagem foi forjada. Um exemplo de criptossistema de chave pública é o RSA (Rivest-Shamir-Adelman) . Sua maior desvantagem é a sua capacidade de canal limitada, ou seja, o número de bits de mensagem que ele pode transmitir por segundo. Enquanto um chip que implementa o algorítmo de uma chave DES pode processar informação em alguns milhões de bits por segundo, um chip RSA consegue apenas na ordem de mil bits por segundo.

Então vejamos , sistemas de uma chave são bem mais rápidos, e sistemas de duas chaves são bem mais seguros. Uma possível solução é combinar as duas, fornecendo assim um misto de velocidade e segurança. Simplesmente usa-se a encriptação de uma chave para encriptar a mensagem, e a chave secreta é transmitida usando a chave pública do destinatário. É importante não confundir chave privada com chave secreta.A primeira é mantida em segredo, enquanto que a segunda é enviada para as pessoas que efetivarão a comunicação

Criptografia Convencional

Técnicas Clássicas

O elemento fundamental da criptografia são as Chaves simétricas para criptografar textos, sendo elas utilizadas para fornecer segurança {Burwick99}.

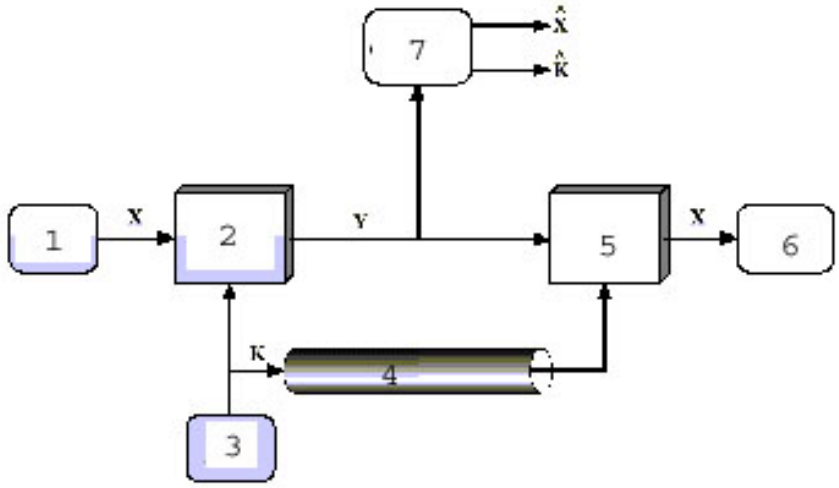
A prática de uma chave (segredo) tem como utilização a criptação e a descriptação, sendo que ela mesma deve ser de poderio tanto do emissor quanto do receptor da mesma. Em cifradores simétricos, o algoritmo de criptografia e decriptografia são os mesmos, muda-se apenas a forma como são utilizadas as chaves.

Na corrente técnica existe duas chaves, sendo que são as mesmas, que servem para criptografar e decriptografar o texto, que, por conseguinte, as mesmas devem ser de conhecimento tanto do emissor quanto de receptor na qual a mensagem se direciona. Os cifradores simétricos possuem o seu algoritmo de criptografia e decriptografia iguais, invertendo somente como as mesmas são usadas em função da chave.

1-Mensagem Fonte 2-Algoritmo de Encriptaçao 3-Chave

4-Canal Seguro 5-Algoritmo de Decriptaçao

6-Destino 7-Cripotoanalise



Modelo de um Criptosistema convencional

encriptação e desencriptação. Um algoritmo de criptografia trabalha dentro de combinações com as palavras e da chave em conjunto com um número. O mesmo texto codifica o texto encriptado de diferente modos com chaves diferentes. A segurança dos dados codificados são completamente dependente de duas coisas: a força do algoritmo de criptografia e o segredo da chave. Um algoritmo de criptografia, junto a todas as possíveis chaves e todos os protocolos que faça trabalhar esta incluído em um sistema de criptografia. PGP é um exemplo de criptografia.

A mensagem enviada é encriptada por Alice, com uma chave K que é de seu conhecimento. A mensagem é enviada para Bob através de algum meio eletrônico. Porém

para Bob conseguir decifrar esta mensagem, ele deve ter a mesma chave K utilizada por Alice. Esta chave K é então enviada por um canal seguro para Bob. Com este modelo pode-se garantir a confidencialidade da mensagem, porque somente Alice e Bob têm conhecimento da chave K.

A criptoanálise tentará descobrir qual foi à chave utilizada para cifrar a mensagem e qual é a mensagem cifrada.

O texto cifrado não sofre alteração quanto ao seu tamanho. É importante salientar também que o texto cifrado não contém qualquer parte da chave.

Uma encriptação é dita computacionalmente segura se atende estes dois critérios:

- O custo para quebrar o cifrador excede ao valor da informação encriptada.
- O tempo requerido para quebrar o cifrador excede o tempo de vida útil da informação.

Criptografia Convencional - Técnicas Modernas

Os principais modos de criptografia convencional moderna são:

- DES Simplificado
- Princípios dos Cifradores de BlocoDES
- Criptanálise Diferencial e Linear
- Projeto dos Cifradores de Bloco
- Modos de Operação
- Funções Bent

As técnicas de Encriptação mais usadas

DES ("Data Encryption Standard")

O DES é um mecanismo de cifragem tradicional ("simétrico) desenvolvido nos anos setenta, utiliza uma chave de 56 bits que é aplicada a blocos de dados com 64 bits, o objectivo destes algoritmos é que seja muito dificil calcular a chave K, mesmo conhecendo o algoritmo DES, uma mensagem cifrada C e uma mensagem original M: C = DES(K,M)

O algoritmo usado é algo complexo:

- A mensagem de 64 bits é dividida em duas partes de 32 bits cada.
- A chave de 56 bits é usada para gerar 16 chaves de 18 bits cada.

É aplicado sucessivamente 16 vezes um algorimo, usando as chaves geradas.

Devido as suas caracteristicas pequenas alterações na mensagem original provocam grandes alterações na mensagem cifrada, isto dificulta as tentativas de conhecer a chave, mesmo que se possa cifrar aquilo que se pretende.

Embora seja dificil de implementar em "software" de uma forma eficiênte, foi desenvolvido "hardware" capaz de implementar este algoritmo de forma eficiênte.

A aplicação de "força bruta" para descobrir a chave, obriga a aplicar o algoritmo um máximo de 2^{56} vezes, ou seja cerca de 72 000 000 000 000 000 vezes.

Este número não é contudo demasiado tranquilizante, este algoritmo é implementado de forma mais eficiente em "hardware", para "quebrar" uma chave DES usa-se um "chip" apropriado que pode ser montado de forma a trabalhar em paralelo com outros semelhantes. O custo depende do tempo em que se pretende quebrar a chave, em 1993 por 1 milhão de dolares podia-se montar uma máquina capaz de descobrir chaves DES em 3 horas e meia. O documento "Efficient DES key search" contém planos detalhados para a implementação desse tipo de máquina.

Para fortalecer o DES seria necessário aumentar o número de bits da chave, contudo o algoritmo exige um valor fixo de 56 bits, a aplicação sucessiva de duas chaves não é solução pois apenas duplica o número de aplicações do algoritmo necessárias para quebrar a chave (técnica "meet-in-the middle"), corresponde por isso a aumentar apenas um bit.

O triplo DES utiliza duas chaves, mas o algoritmo é aplicado três vezes segundo a seguinte equação:

$C = DES(K1 , DES^{-1}(K2 , DES(K1 , M)))$, onde $DES^{-1}($ representa o algoritmo inverso (decifragem).

O triplo DES corresponde à utilição de uma chave de 90 bits, tem ainda a vantagem de poder ser usado para DES simples, basta que K1=K2.

Além da "força bruta", existem outras abordagem para descobrir chaves:

Cripto-análise diferencial

Para usar esta técnica é necessário que se possa cifrar as mensagens que se pretende, em função de alterações nessas mensagens e resultados na mensagem cifrada, no caso do DES simples é possivel reduzir as chaves a 2^{47}.

Cripto-análise linear

Tenta definir a chave por aproximação linear em função da informação recolhida de pares (M,C), no caso do DES simples reduz o número de chaves a 2^{43}.

RC5

O RC5 é uma técnica mais recente e mais flexivel, tal como o DES é uma técnica de criptação simétrica (a mesma chave é usada para cifrar e para decifrar), também é uma tecnica de blocos, mas ao contrário do DES não está limitada a blocos de dimensão fixa, igualmente a chave não tem uma dimensão fixa.

Tal como o DES utiliza a aplicação sucessiva de um algoritmo, contudo o número de aplicações não é fixo, deste modo pode obter-se um maior grau de segurança usando um maior número de aplicações.

O RC5 é como se pode deduzir muito flexivel, estando sujeito a uma serie de parâmetros que devem ser ajustados às necessidades particulares de cada caso.

A mensagem original é fornecida ao algoritmo sob a forma de dois blocos de **w** bits, correspondendo ao alinhamento mais conveniente para o sistema em causa, os valores tipicos para **w** são: 16, 32 e 64. A mensagem cifrada possui forma idêntica.

Outro parâmetro importante é o número de aplicações do algoritmo (**r**), pode variar de 1 a 255. Para aplicar **r** vezes o algoritmo, vai ser gerada apartir da chave uma tabela com **t = 2.(r+1)** blocos de **w** bits.

A chave é especificada pelos parâmetros **b** e **k**, **b** especifica o número de bytes (octetos) que constitui a chave e **k** é a chave propriamente dita.

É habitual usar a notação RC5-w/r/b para especificar uma implementação particular RC5. Podemos dizer que o RC5-32/16/7 é equivalente ao DES.

O documento "The RC5 Encryption Algorithm" contém grandes detalhes sobre o RC5, incluido uma implementação em C.

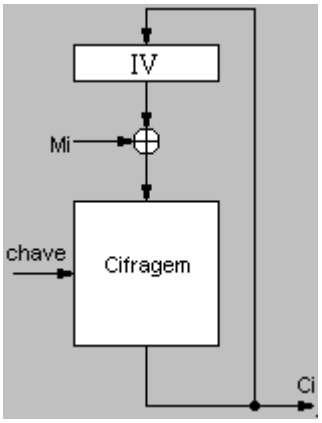
Aplicação das técnicas de cifragem em bloco

As técnicas de cifragem em bloco (Ex: DES e RC5) podem ser aplicadas de diversos modos a mensagens de comprimento diferente do tamanho de bloco.

A técnica mais simples é conhecida por **ECB ("Electronic Code Book")**, consiste em dividir a mensagem em blocos de tamanho adequado, cifrar os blocos em separado e concatenar os blocos cifrados na mesma ordem. O grande inconveniente desta técnica é que blocos de mensagem original idênticos vão produzir blocos cifrados idênticos, isso pode não ser desejável.

A técnica **CBC ("Cipher Block Chaining")** evita este inconveniente, realiza a operação **xor** entre o bloco a cifrar $\mathbf{M_i}$ e o bloco anteriormente cifrado $\mathbf{C_i}$, só depois aplica o algoritmo de cifragem:

$$\mathbf{C_i = cifragem(\ chave,\ M_i\ xor\ C_{(i-1)})}$$



Na decifragem obtém-se $\mathbf{M_i\ xor\ C_{(i-1)}}$, como $\mathbf{xor^{-1} = xor}$, utiliza-se:

$$\mathbf{M_i = decifragem(\ chave,\ C_i)\ xor\ C_{(i-1)}}$$

Como para o primeiro bloco não existe mensagem anterior, utiliza-se um bloco aleatório conhecido por IV ("Initialization Vector").

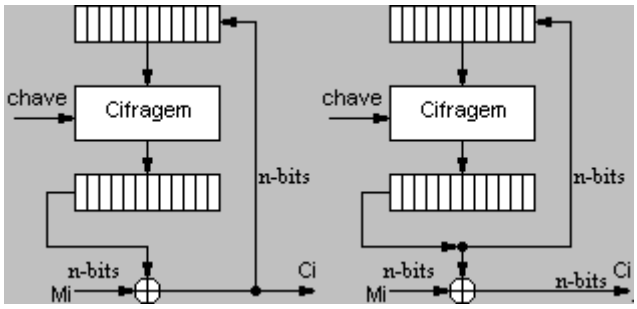
Esta técnica é pouco favorável sob o ponto de vista da propagação de erros, uma vez que um erro na transmissão de um bloco cifrado $\mathbf{C_i}$ vai inutilizar tanto o bloco $\mathbf{M_i}$ como o seguinte $\mathbf{M_{(i+1)}}$.

Nas técnicas **CFB ("Cipher FeedBack")** e **OFB ("Output FeedBack")** a mensagem não é directamente cifrada, existe um vector de inicial (IV) ao qual é aplicado o algoritmo de cifragem, aplica-se então a operação **xor** entre o vector cifrado e a mensagem.

A operação **xor** entre o vector cifrado e a mensagem é realizada do seguinte modo:
Pegam-se nos **n** bits da esquerda do vector cifrado e realiza-se a operação **xor** com os **n** seguintes da mensagem.
Realiza-se o "shift" para a esquerda de **n** bits do vector cifrado.
Injectam-se **n** bits no vector original, realizando o "shift" para a esquerda.

No caso do CFB utilizam-se os **n** bits resultantes da operação **xor**.
No caso do OFB utilizam-se os **n** bits retirados do vector cifrado.
Quando esgotam os bits do vector cifrado aplica-se novamente o algoritmo de cifragem ao vector original.

As figuras seguintes ilustram as duas técnicas: CFB (à esquerda) e OFB (à direita):



Os valores mais comuns para **n** são 1, 8 ou 64. Devido ao seu funcionamento a tecnica OFB, também conhecida por OFM ("Output Feedback Mode") apenas produz um bloco de mensagem errado quando ocorre um erro na transmissão de um bloco cifrado.

Quando o comprimento da mensagem não é multiplo do tamanho do bloco é necessário recorrer a técnicas de enchimento ("padding"), uma técnica habitual é adicionar um bit 1 seguido dos bits 0 necessários.

Distribuição de chaves

O DES e o RC5 são cifragens simétricas: básicamente a decifragem é o inverso da cifragem, usando a mesma chave que por razões obvias deve ser secreta, isto é apenas do conhecimento da entidade de origem e entidade de destino.

O primeiro problema levantado pela criptografia de chave secreta é a distribuição de chaves, poderiam ser enviadas por um sistema paralelo como correio ou "fax", mas o mais conveniente seria usar a rede de comunicação, para tal a chave deveria ser cifrada usando uma chave anterior.

Para estabelecer uma comunicação segura entre uma aplicação cliente e uma aplicação servidora, o ideal seria que o servidor e o cliente trocassem entre sí as chaves na altura do estabelecimento da conexão.

Para se conseguir uma situação deste tipo existem duas soluções:
Manter a criptografia de chave secreta e usar uma técnica especial para enviar a chave, por exemplo "puzzles".
Utilizar criptografia de chave pública que veio alterar radicalmente o modo de ver a criptografia.

"Puzzles"

Para fornecer a chave secreta é enviado um conjunto de "puzzles", geralmente na ordem das dezenas de milhar. Tomando o exemplo DES, cada "puzzle" é constituido por 120 bits zero, seguido do número do "puzzle" com 16 bits e finalmente uma chave, DES de

56 bits. Todos os "puzzles" são cifrados com chaves DES em que os últimos 22 bits são zero.

O cliente escolhe um "puzzle" à sorte e quebra a cifra usando força bruta, tem de tentar "apenas" 2^{34} chaves, quando obtém 120 zeros no inicio sabe que consegui decifrar o "puzzle" e portanto possiu já a chave DES que escolheu (últimos 56 bits do "puzzle" decifrado). Tem agora de indicar qual o "puzzle" que escolheu, envia então uma mensagem com o número do "puzzle" cifrado com a chave DES escolhida, destinatário (ex.:servidor) conhece as chaves que iam nos "puzzles" e o respectivo número de "puzzle" e pode facilmente descobrir qual foi a chave escolhida.

O tempo médio que um intruso necessita para descobrir a chave situa-se na ordem dos anos e pode ser ajustado por variação do número de "puzzles" em jogo.

Criptografia de chave pública

A criptografia de chave pública utiliza algoritmos de cifragem e decifragem que não estão diretamente relacionados, passam a existir duas chaves, uma de cifragem e outra de decifragem.

Tipicamente a chave de cifragem é pública (é divulgada a todos os utilizadores), a chave de decifragem é secreta. Quando uma entidade A pretende enviar à entidade B uma mensagem cifra-a com a chave pública de B antes do envio. Ninguem, nem sequer a entidade A é capaz de decifrar, apenas a entidade B que possui a chave secreta adequada.

Além de resolver definitivamente o problema da distribuição de chaves, a criptografia de chave pública facilita significativamente a implementação de mecanismos de autenticação de mensagens e assinatura digital.

Embora tenham sido propostos outros algoritmos, actualmente o RSA é o mais sólido, o algoritmo "Merkle-Hellman Knapsacks" demorou quatro anos a ser quebrado por Adi Shamir, uma segunda versão, supostamente mais sólida, demorou dois anos a ser quebrada.

RSA

Este algoritmo é devido a Ron **R**ivest, Adi **S**hamir e Len **A**dleman (RSA), baseia-se no seguinte: **é simples arranjar dois números primos grandes, mas é muito complicado (moroso) factorizar o seu produto**. O RSA tem aguentado todas as investidas dos cripto-analistas, contudo temos que atender ao facto de ser um problema matemático, existe sempre o risco de descoberta de uma técnica para resolver o problema de forma eficiênte.

Geração das chaves:

Escolhem-se dois número primos grandes **a** e **b**.

Calcula-se **n = a x b**.

Calcula-se **ø(n) = (a-1) x (b-1)**.

Escolhe-se um número pequeno **p** que seja primo relativo de **ø(n)** e **< ø(n)**, calcula-se **s** tal que **(p x s) mod ø(n) = 1**, onde **mod** é o operador "resto da divisão inteira" (aritmética modulo ø(n)).

(Dois números são primos relativos se o maior divisor comum é 1)

O par **(n,p)** constitui a chave pública, **d** é a chave secreta.

Aplicação:

Cifragem: $\mathbf{C = M^p \bmod n}$

Decifragem: $\mathbf{M = C^s \bmod n}$

, onde **M** e **C** são respectivamente a mensagem original e mensagem cifrada, ambas com valores possiveis de zero a **n-1**.

Uma propriedade interessante do RSA é a possibilidade de inversão das chaves, pode-se cifrar uma mensagem com a chave s, para decifrar será agora necessária a chave pública: utilizável para autenticação e assinatura digital.

Para efeitos de exemplificação tomem-se os números primos a=7 e b=17:

n = a x b = 119

ø(n) = (a-1) x (b-1) = 96

como primo relativo de ø(n) podemos escolher p=5

então para obter p x s mod 96 = 1, podemos usar s = 77

pois 5 x 77 = 385, 385 mod 96 = 1

A chave pública é (5;119) e a chave secreta é 77

Exemplos de aplicação:

Para evitar perdas de dados torna-se necessário aplicar a seguinte propriedade da aritmética modular:

(a x b) mod n = ((a mod n) x (b mod n)) mod n

Para cifrar o número **2** com a chave pública temos $C = 2^5 \bmod 119 = 32 \bmod 119 = \mathbf{32}$

Para decifrar utiliza-se $M = \mathbf{32}^{77} \bmod 119$

, para usar uma calculadora, sem perder dados podemos usar 11 parcelas:

$((32^7 \bmod 119) \times ... \times (32^7 \bmod 119)) \bmod 119$

, obtemos então $25^{11} \bmod 119$, podemos agora aplicar $((5^{11} \bmod 119) \times (5^{11} \bmod 119)) \bmod 119$, obtemos agora $45^2 \bmod 119 = \mathbf{2}$

Também se pode cifrar o número **2** com a chave secreta temos $C = 2^{77} \bmod 119$

, para usar uma calculadora, sem perder dados podemos usar 11 parcelas:

$((2^7 \bmod 119) \times ... \times (2^7 \bmod 119)) \bmod 119 = 9^{11} \bmod 119 = \mathbf{32}$

Para decifrar utiliza-se $M = \mathbf{32}^5 \bmod 119 = \mathbf{2}$

Para cifrar o número **3** com a chave pública temos $C = 3^5 \bmod 119 = 243 \bmod 119 = \mathbf{5}$

Para decifrar utiliza-se $M = \mathbf{5}^{77} \bmod 119$

, para usar uma calculadora, sem perder dados podemos usar 7 parcelas:

$((5^{11} \bmod 119) \times ... \times (5^{11} \bmod 119)) \bmod 119$

, obtemos então $45^7 \bmod 119 = \mathbf{3}$

Como se pode verificar as operações a realizar na decifragem não são simples, especialemente se atendermos a que os números **a** e **b** devem ser grandes.

Para os valores 0 e 1 a messagem e o resultado da cifragem coincidem, contudo isto não é muito grave, os valores usados para n são muito elevados (na ordem de 10^{200}), o tamanho mais comum para as mensagens a cifrar (M) é de 512 bits (que representa números até mais de 10^{154}), para este número de bits não são vulgares os valores 0 e 1, de qualquer modo isto pode ser resolvido pela adição de duas unidades a M antes de entrar no algoritmo de cifragem e subtração de duas unidades depois de sair do algoritmo de decifragem.

Gerar chaves RSA não é uma operação simples, o primeiro problema é arranjar dois números primos **a** e **b** com uma ordem de grandeza de 10^{100}, usar os algoritmos tradicionais de geração de números primos é impossivel, a solução é usar testes eliminatórios, estes testes permitem saber se um número não é primo, ou qual a probabilidade de ser primo, se

um dado número depois de testado intensivamente não é eliminado será adoptado. A segunda questão prende-se com a determinação de um primo relativo de ø(n), p ou s e de seguida é necessário determinar outro número para verificar a relação (p x s) mod ø(n) = 1.
O algoritmo RSA serve de base a muitos sistemas de segurança actuais, tais como o PGP ("Pretty Good Privacy").

Sob o ponto de vista de cripto-análise e devido ao número de bits das chaves a aplicação de força bruta (tentar todas as chaves secretas possíveis) está excluida. A abordagem é tentar obter os dois factores primos de n. Contudo tal é extremamente complexo para a ordem de grandeza usada para n, o tempo necessário cresce exponencialmente com o valor de n.

Distribuição de chaves públicas

Embora a criptografia de chave pública resolva o problema da distribuição de chaves existe ainda a questão do modo como as chaves públicas serão obtidas por quem delas necessita.

As chaves públicas destinam-se a ser divulgadas, mas esta divulgação deve ser realizada de tal modo que não possa ser forjada por terceiros, as consequências seriam obvias.

O correio electrónico ou sistemas de news não são de todo adequados, uma melhor solução será a sua colocação na página WWW pessoal.

Uma opção mais segura é definir uma **autoridade de chaves públicas**, quando A pretende enviar uma mensagem a B realiza as seguintes operações: contacta a autoridade C enviando-lhe um pedido com etiqueta temporal C responde enviando uma mensagem cifrada com a sua chave secreta (assim A sabe que a mensagem veio de C), onde consta chave pública de B e a mensagem original. Quando B recebe a primeira mensagem de A terá de realizar o mesmo procedimento para obter a chave de B e lhe poder responder.

Apesar de mais seguro a existencia de autoridades de chave pública coloca alguns problemas em termos de quantidade de comunicações necessárias, uma alternativa é a emissão de "**certificados de chave pública**". Cada entidade contacta a autoridade de chave pública que lhe fornece um certificado contendo: uma etiqueta temporal; a identificação da entidade; a chave pública da entidade. O certificado encontra-se cifrado com a chave secreta da autoridade, este facto atesta a sua origem. As diversas entidades podem agora trocar directamente entre si estes certificados, o facto de estarem cifrados pela autoridade atesta a sua veracidade.

Distribuição de chaves secretas (criptografia simétrica)

A criptografia de chave pública pode ser usada para obviar um dos grandes problemas da criptografia tradicional: a distribuição de chaves secretas. A motivação é o facto de a criptografia tradicional ser substancialmente mais rápida nas operações de cifragem e decifragem, proporcionando assim débitos de dados mais elevados.

A utilização mais direta consistiria no seguinte:

- A aplicação **A** envia à aplicação **B** uma mensagem com a chave publica da **A**.
- A aplicação **B** gera uma chave secreta convencional (simétrica) e envia-a a **A** cifrada com a respectiva chave publica.

- A aplicação **A** decifra a mensagem com a sua chave secreta e obtém a chave secreta convencional.

Este mecanismo pode ser contudo furado por uma entidade **C** com controlo sob as transmissões:

- **C** captura a primeira mensagem de **A** e fica a conhecer a chave pública de **A**.
- **C** envia para **B** uma cópia da mensagem de **A**, mas substitui a chave pública pela sua.
- **C** captura a resposta de **B** e fica a conhecer a chave secreta convencional.

para que **A** não note nada **C** envia a **A** a resposta que recebeu de **B**, cifrada com a chave publica de **A**.

- **C** passa então a actuar de modo passivo, limitando-se a decifrar a conversação entre **A** e **B**.

O problema pode ser resolvido se previamente for usado um mecanismo seguro de distribuíção de chaves publicas e forem tomadas diversas precauções quanto a confidencialidade e autenticação:

- **A** envia a **B** uma mensagem com a sua identificação e um identificador de transação I_1(número que identifica a transação em curso), cifrada com a chave pública de **B**.
- **B** enviar a **A** uma mensagem com I_1 e um novo identificador I_2, cifrada com a chave pública de **A**.
- **A** tem a certeza que a mensagem veio de **B** devido à presença de I_1, envia então a **B** I_2, cifrado com a chave pública de B.
- **A** gera a chave secreta convencional, cifra-a com a sua chave secreta e de **A**, cifra-a com a chave pública de **B**, e envia-a a **B**.

B recebe a mensagem e decifra-a aplicando primeiro a sua chave secreta e de seguida a chave publica de **A**.

Algoritmos

Algoritmos de Chave Única ou Secretas (Simétricos)

O exemplo mais difundido de cifrador computacional de chave única é o DES (Data Encryption Standard), desenvolvido pela IBM e adotado como padrão nos EUA em 1977. O DES cifra blocos de 64 bits (8 caracteres) usando uma chave de 56 bits mais 8 bits de paridade (o que soma 64 bits).O algoritmo inicia realizando uma transposição inicial sobre os 64 bits da mensagem, seguida de 16 passos de ciftagem e conclui realizando uma transposição final, que é a inversa da transposição inicial.Para os 16 passos de ciftagem usam-se 16 sub-chaves, todas derivadas da chave original através de deslocamentos e transposições.

Um passo de cifragem do DES, tem dois objetivos básicos: a difusão e a confusão. A difusão visa eliminar a redundância existente na mensagem original, distribuindo-a pela mensagem cifrada. O propósito da confusão é tomar a relação entre a mensagem e a chave tão complexa quanto possível.O DES pode ser quebrado pelo método da força bruta, tentando-se todas as combinações possíveis de chave. Como a chave tem 56 bits, tem-se um total de 2 exp 56 chaves possíveis.

Existem diversos algoritmos de cifragem de blocos de chave única, entre eles:

- Triple-DES: O DES é aplicado 3 vezes, com sequências de cifragem e decifragem, combinando a utilização de 2 chaves.

- WLucifer: precursor do DES.
- Madryga: trabalha com 8 bits, usando ou-exclusivo e deslocamento de bits.
- NewDES: blocos de 64 bits e chave de 120 bits.
- FEAL-N: baseado no DES, pode-se especificar o número de passos da cifragem, fraco se utiliza-se menos de 8 passos.
- LOKI: bloco e chave de 64 bits.
- Khufu e Khafre: trabalham de forma semelhante ao DES, usam tabelas de substituição de 256 posições de 32 bits - contra as de 6 posições de 4 bits do DES - usam chaves de 512 bits e um número de passos flexíveis, múltiplo de 8.
- IDEA: blocos de 64 bits com chave de 128 bits.
- MMB: blocos e chave de 128 bits.
- Skipjack: chave de 80 bits e 32 passos de processamento.

Estes e outros algoritmos podem ser encontrados em :
ftp://ftp.funet.fi:/pub/crypt/cryptography/symmetric

Algorítmos de Chave Pública (Assimétricos)

A chave de ciframento é públicada ou tornada acessível aos usuários, sem que haja quebra na segurança. Dessa forma cada usuário tem uma chave de ciframento, de conhecimento público, e outra de deciframento, secreta. Se um usuário A deseja mandar uma mensagem para um usuário B, ele utiliza a chave de ciframento pública PB e envia a mensagem para B, este de posse de sua chave de deciframento secreta SB decodifica a mensagem.

Um exemplo desse sistema é o RSA, anacronico de seus autores Rivest, Shamir e Adleman. Sua seguranca baseia-se na intrabilidade da fatoraçãao de produtos de dois primos. Um usuário B para determinar seu par (PB,SB), procede da seguinte maneira: escolhe ao acaso dois primos grandes "p" e "q" e computa o seu produto (n=p*q), e o número f(n)=(p-1)*(q-1); B escolhe ao acaso um número "c" relativamente primo com f(n) (ou seja, c e f(n) não possuem fatores em comum) e determinara "d" tal que c*d (módulo f(n)). Finalmente, o usuário B pública sua chave pública PB(c,n) e mantém secretos p, q, f(n) e d. A chave secreta SB(d,n) deve ser mantida em sigilo completo.

Cifragem de Blocos

Um algoritmo que realiza cifragem sobre blocos pode operar de diversas maneiras distintas. As mais conhecidas são:

1 - Modo do livro de Códigos (Electronic Code Book - ECB)

Cada bloco da mensagem original é individual e independentemente cifrado para produzir os blocos da mensagem cifrada. O bloco típico tem 64 bits, o que produz um livro de códigos de 2 exp 64 entradas. E note-se que para cada chave possível existe um livro de códigos diferentes. A vantagem do método é sua simplicidade e a independência entre os blocos. A desvantagem é que um criptoanalista pode começar a compilar um livro de códigos, mesmo sem conhecer a chave. Um problema mais grave é a chamada repetição de bloco, onde um atacante ativo pode alterar parte de uma mensagem criptografada sem saber a chave e nem mesmo o conteúdo que foi modificado. Pode-se por exemplo interceptar uma transação bancária de transferência de saldo de qualquer pessoa, a seguir pode-se realizar uma transferência de saldo de uma conta para a conta do atacante e interceptar a mensagem,

assim pode-se identificar os blocos correspondentes ao destinatário e dessa forma substituir em todas as mensagens o destinatário pelo atacante.

2 - Modo de Encadeamento de Blocos (Cipher Block Chaining - CBC)

CBC realimenta a cifragem do bloco atual com o resultado das cifragens dos blocos anteriores. A operação mais utilizada é o ou-exclusivo com o bloco anterior, dessa forma os blocos iguais serao normalmente cifrados de forma diferente, desde que no mínimo um dos blocos anteriores seja diferente da mensagem. Entretanto 2 mensagens iguais serao mapeadas para os mesmos blocos. E duas mensagens com início igual serão cifradas da mesma forma até que ocorra a diferença. A maneira empregada para evitar esse problema é a utilizacao de um vetor de inicializaçãao distinto para cada mensagem.

3 - Modo da Realimentação de Cifra (Cipher Feedback - CFB)

Quando há necessidade de enviar-se mensagens que possuem tamanho menor que um bloco usa-se o método CFB, que trabalha com grupos (8 bits por exemplo - 1 caracter), neste caso a realimentacao é feita sobre o grupo, utilizando-se também o ou-exclusivo.

4 - Cifras de Substituição

Troca cada caracter ou grupo de caracteres por outro, de acordo com uma tabela de substituição.Pode-se quebrar este método analisando-se a frequência de cada caracter no texto cifrado e comparando-se estas frequências com aquelas que normalmente aparecem em um determinado idioma.As vogais têm maior frequência que as consoantes e alguns caracteres possuem frequência baixíssima em relação aos demais. Para amenizar a frequência de caracteres, podemos utilizar várias tabelas para a cifragem de um texto. Para uma substituição monoalfabética podemos ter 261 Tabelas de Substituição. Tem-se uma chave que diz qual das tabelas será usada para cada letra do texto original. Portanto, quanto maior a chave mais seguro é o método. Entretanto, é suficiente descobrir o tamanho da chave k e analisar blocos de k caracteres no texto, verificando a frequência de repetição dos caracteres.

4.1 - Substituição Monoalfabética

Cada letra do texto original é trocada por outra de acordo com uma tabela e com sua posição no texto. A Substituição de César é um exemplo de substituição monoalfabética que consiste em trocar cada letra por outra que está 3 letras adiante na ordem alfabética. Ex: A=D. Pode-se usar outros valores ao invés de 3, o que constitui a chave de ciframento. Existem apenas 26 chaves, por isso é um método que visa proteger textos com pequeno grau de sigilo.

4.2 - Substituição por Deslocamentos

A chave indica quantas posições deve-se avançar no alfabeto para substituir cada letra. Diferente da substituição de César, as letras não são trocadas sempre por uma letra n posições a frente no alfabeto. Ex: Chave:020813, A primeira letra é trocada pela letra que está 2 posições a frente no alfabeto, a segunda pela que está 8 posições a frente, e assim por diante, repetindo a chave se necessário. (P AI=RIV).

4.3 - Substituição Monofônica

Como a anterior, mas agora cada caracter pode ser mapeado para um ou vários caracteres na mensagem cifrada. Isso evita a linearidade da substituição.

4.4 - Substituição Polialfabética

A combinação no uso de várias substituições monoalfabéticas, usadas em rotação de acordo com um critério ou chave. Por exemplo, poderiam ser utilizadas 4 tabelas, usadas em alternância a cada 4 caracteres. Substituição por Polígramos: utiliza grupo de caracteres ao invés de um caracter individual. Se fossem considerados trigramas, por exemplo, ABA poderia ser substituído por RTQ ou KXS,

5 - Cifras de Transposição

Troca-se a posição dos caracteres na mensagem. Por exemplo, pode-se rescrever o texto percorrendo-o por colunas. Ou então definir o tamanho para um vetor de trocas e também uma ordem em que as trocas serão feitas. Pode-se usar chave para isso. Ex: em um vetor de tamanho 6 pode-se trocar o primeiro caracter pelo terceiro, o segundo pelo quinto e o quarto pelo sexto.Se a frequência dos caracteres for a mesma do idioma, temos substituição por transposição. Se for diferente, temos por substituição.Também é possível combinar substituição e transposição, ou vice-versa.

6 - Máquinas de Cifragem

Um código trabalha com grupos de caracteres de tamanho variável, ao contrário da cifra. Cada palavra é substituida por outra. Quebrar um código equivale a quebrar uma gigantesca substituição monoalfabética onde as unidades são as palavras e não os caracteres. Para isso deve-se usar a gramática da língua e analisar a estrutura das frases. Máquinas de cifragem baseiam-se em engrenagens que tem tamanhos diferentes e que giram a velocidades diferentes, obtendo um substituição polialfabética com chave de 26n, onde n é o número de engrenagens.

Chaves secretas

Funções Unidirecionais

Podemos dizer que uma uma função é unidirecional se for viável computá-la e computacionalmente inviável computar a sua inversa. Imagine que temos dois números primos da ordem de 10 exp lOO: multiplicá-los é uma questão de segundos com a tecnologia atual, no entanto, dado o seu produto da ordem de 10 exp 200, o melhor algoritmo conhecido leva hoje l bilhão de anos para fatorar o produto dado. Assim a função produto de dois primos é unidirecional. Uma funcao unidirecional é com segredo se existe uma informação que torna a computação da sua inversa possível. A funçao produto de dois primos é unidirecional sem segredo.

Há casos em que uma função unidirecional sem segredo é útil, um exemplo típico é na proteção de senhas, apresenta as senhas cifradas por uma função unidirecional sem segredo (e de inviável deciframento). Quando o usuário inicia sua sessão, fornece a senha que é então ciftada e comparada com a senha cifrada armazenada. Desta maneira, exige-se apenas a integridade do arquivo de senhas, não mais exigindo controle de acesso ao arquivo.

Ao selecionar uma função unidirecional como função de ciframento, o projetista deve supor que:

- o algoritmo de ciframento é de domínio público;
- o espião, através de escuta, tem acesso ao texto cifrado.

Diz-se então que a criptoanálise é de texto cifrado conhecido.

Protocolo para a Distribuição de Chaves Secretas

Quando se adota o método de chaves secretas, é recomendável não utilizar por muito tempo a mesma. Quando ideal é a cada nova sessão uma nova chave seja estabelecida. Mas como estabelecer a chave ao início de cada sessão? Como evitar as escutas? Cifrar a mensagem? Com que chave? Aqui apresenta-se uma solução para ilustrar o conceito de funções unidirecionais. A função a ser usada é a exponencial módulo de um número, isto é, dados os inteiros 'a', 'x' e 'n', seja ~x)=aAx mod n (n>O, x>=O). Assim, ftx) é o resto da divisão de aAx por n. O cálculo desta função é viável. O procedimento abaixo mostra uma maneira de calcular esta função :

```
Procedimento expomod (a,x,n,r:inteiro); {r possui o resultado da função}
declare y, c :  tipo inteiro
inicio
r:= l;
y:=x;
c:=a mod n;

   enquanto y>O faça inicio
      se ímpar(y) então
         r=r*c mod n;
         y=y div 2;
         C=C 2 mod n;
      fim;
fim;
```

Suponha que dois usuários A e B desejam manter uma conversa sigilosa através de chave secreta. As duas partes escolheram um número primo grande , p' da ordem de 10A 100, e já concordaram também em utilizar uma base , a'. Preferivelmente deve ser uma raiz primitiva de p, de modo que ffx)=aAx mod p é uma b~eção sobre o conjunto 1..p-l dos naturais x tais que l<=x<=p-l. Para iniciar o estabelecimento da chave, A gera ao acaso um expoente x no intervalo 1..p- l e B gera outro, y Usando expomod, A calcula ffx) e B ffy). Então A envia pela rede ffx) e B envia ffy). De posse de y e ffx), B calcula, usando expomod :

K = [(ffx)]Ay mod p = (aAx mod p)Ay mod p = aA(xy) mod p = K.

Da mesma forma, A usa expomod e de posse de x e ffy) calcula:

k = [(ffy)]AX mod p = (aAy mod p)AX mod p = aA(xy) mod p - - K.

Assim A e B chegam a um número comum K, que será a chave de ciframento para as mensagens.

Suponha um espião bem informado que obtenha os valores de a e p e, através de escuta, os valores de f(f(x)) e de f(f(y)). Para determinar K, ele precisa determinar a função logaritmo módulo p, que é intratável. Mesmo A não é capaz de determinar o valor de y e B o valor de x. A função expomod é unidirecional sem segredo, permite a A e B trocarem uma chave secreta utilizando a própria rede.

Assinatura Digital

Nos sistemas com chave pública, qualquer pessoa pode cifrar uma mensagem, mas somente o destinatário da mensagem pode decifrá-la. Invertendo-se o uso das chaves podemos ter uma que só pode ser cifrada por uma pessoa e decifrada por qualquer um, obtendo-se assim umefeito de personalização do documento, semelhante a uma assinatura. Um sistema desse tipo é denominado assinatura digital. Assim para personalizar uma mensagem, um determinado usuário A codifica uma mensagem utilizando sua chave secreta e a envia para o destinatário. Somente a chave pública de A permitirá a decodificação sua chave secreta e a envia para o da mensagem, portanto é a prova de que A enviou a mensagem. A mensagem assim pode ser decodificada por qualquer um que tenha a chave pública de A. Para garantir o sigilo deve-se a primeira utilizando a própria chave secreta (para fazer a criptografia duas vezes a mensagem: a chave pública do destinatário, para que somente este possa ler a mensagem.

Propriedades

1 - a assinatura é autêntica: quando um usuário usa a chave pública de A para decifrar uma mensagem, ele confirma que foi A e somente A quem enviou a mensagem;
2 - a assinatura não pode ser forjada: somente A conhece sua chave secreta:
3 - o documento assinado não pode ser alterado : se houver qualquer alteração no texto criptografado este não poderá ser restaurado com o uso da chave pública de A;
4 - a assinatura não é reutilizável: a assinatura é uma função do documento e não pode ser transferida para outro documento;
5 - a assinatura não pode ser repudiada: o usuário B não precisa de nenhuma ajuda de A para reconhecer sua assinatura e A não pode negar ter assinado o documento.

Certificado digital

Certificado de Identidade Digital, também conhecido como Certificado Digital, associa a identidade de um titular a um par de chaves eletrônicas (uma pública e outra privada) que, usadas em conjunto, fomecem a comprovação da identidade. É uma versão eletrônica (digital) de algo parecido a uma Cédula de Identidade - serve como prova de identidade, reconhecida diante de qualquer situação onde seja necessária a comprovação de identidade. Certificado Digital pode ser usado em uma grande variedade de aplicações, como comércio eletrônico, groupware (Intranet's e Internet) e transferência eletrônica de fundos (veja o exemplo recente do Banco Bradesco S.A. na implantação do seu serviço Internet - o BradescoNet). Dessa forma, um cliente que compre em um shopping virtual, utilizando um Servidor Seguro, solicitará o Certificado de Identidade Digital deste Servidor para verificar, a identidade do vendedor e o conteúdo do Certificado por ele apresentado. De forma inversa, o servidor poderá solicitar ao comprador seu Certificado de Identidade Digital, para identificá-lo com segurança e precisão. Caso qualquer um dos dois apresente um Certificado de Identidade Digital adulterado, ele será avisado do fato, e a comunicação com segurança não será estabelecida. O Certificado de Identidade Digital é emitido e assinado (chancelado) por uma Autoridade Certificadora Digital(Certificate Authority), como a Thawte (certificadora da ArtNET), que emite o Certificado. Para tanto, esta autoridade usa as mais avançadas técnicas de criptografia disponíveis e de padrões internacionais (norma ISO X.509 para Certitncados Digitais), para a emissão e chancela digital dos Certificados de Identidade Digital.

Um certificado contem três elementos*:*

1 - Informação de atributo

Esta é a informação sobre o objeto que é certificado. No caso de uma pessoa, isto pode incluir seu nome, nacionalidade e endereço e-mail, sua organização e o departamento desta organização onde trabalha.

2 - Chave de informação pública

Esta é a chave públicada entidade certificada. O certificado atua para associar a chave pública à informação de atributo, descrita acima. A chave pública pode ser qualquer chave assimétrica, mas usualmente é uma chave RSA.

3 - Assinatura da Autoridade em Certificação (CA)

A CA assina os dois primeiros elementos e, então, adiciona credibilidade ao certificado.Quem recebe o certificado verifica a assinatura e acreditará na informação de atributo e chave públicaassociadas se acreditar na Autoridade em Certificação.

Selo Cronológico Digital

O Servico de Selo Cronológico Digital gera selos cronológicos que associam a data e a hora a um documento digital em uma forma de criptografia forte. O selo cronológico digital pode ser usado futuramente para provar que um documento eletrônico existia na data alegada por seu selo cronológico.

Por exemplo, um físico que tenha uma idéia brilhante pode descrevâ-la usando um processador de textos e selar este documento com o selo cronológico digital.O selo cronológico e o documento, juntos, podem mais tarde comprovar que este cientista é o merecedor do Prêmio Nobel, mesmo que um rival publique essa idéia primeiro.

Exemplo de uso do sistema: suponha que Paulo assine um documento e queira selá-lo cronologicamente. Ele calcula o resumo da mensagem usando uma função de hashing seguro e, então, envia este resumo (não o documento) para o DTS, que enviará de volta um seio cronológico digital consistindo do resumo da mensagem, da data e da hora em que foi recebida pelo DTS e da assinatura do DTS. Como o resumo da mensagem não revela qualquer informação a respeito do conteúdo do documento, o DTS não tem condições de saber o conteúdo do documento que recebeu o selo cronológico digital. Mais tarde, Paulo pode apresentar o documento e o selo cronológico, juntos, para provar a data em que este foi escrito. Aquele que vai comprovar a autenticidade do documento calcula o resumo da mensagem, verifica se as mensagens calculada e apresentada são iguais, e observa então a assinatura do DTS no selo cronológico. Para ser confiável, o selo cronológico não pode ser falsificável. Considere os requisitos para um DTS como descrito a seguir.

O DTS deve ser proprietário de uma chave longa (1.024 bits), se este desejar que os selos cronológicos sejam seguros por muitas décadas. A chave privativa do DTS deve ser armazenada em um local de máxima segurança, como, por exemplo, um cofre inviolável em um local seguro. A data e a hora vêm de um relógio que não possa ser alterado, (NIST) Deve ser impossível criar selos cronológicos sem usar um mecanismo que só aceite este relógio.

O uso do DTS parece ser extremamente importante, se não essencial, para manter a validade de documentos através dos anos. Suponha um contrato de leasing de vinte anos entre um proprietário de terras e um arrendatário. As chaves públicas usadas para assinar o contrato expiram após um ano. Soluções, como reafirmar as chaves ou reassinar o contrato a cada ano, com novas chaves, requerem a cooperação de ambas as partes durante vários

anos enquanto durar o contrato. Se uma das partes se torna insatisfeita com o contrato, ela pode recusar-se a cooperar. A solução é registrar o contrato com o DTS na data da primeira assinatura deste. Ambas as partes recebem então uma cópia do selo cronológico, que pode ser usada anos mais tarde para comprovar a autenticidade do contrato original.

No futuro, o provável é que o DTS será usado para tudo, desde a assinatura de contratos a longo prazo até diários pessoais e cartas. Hoje, se um historiador descobrir algum manuscrito e atribuí-lo a um escritor famoso (já falecido), sua autenticidade poderá ser comprovada por meios físicos. Mas, se um achado semelhante ocorrer adqui a 100 anos, provavelmente será em arquivos de computador (disquetes ou fitas). Talvez a única forma de comprovar sua autenticidade seja através do selo cronológico digital.

Site seguro

Um site seguro é constituído por programas de computador que são executados em um servidor seguro para atender solicitações feitas pelos usuários finais, através de seus próprios programas (clientes seguros). Dotado de características que tornam as transações eletrônicas confidenciais, mediante criptografia, o servidor seguro utiliza-se de um protocolo especial de comunicação que é o SSL ((" Secure Socket Layer" - desenvolvido originalmente pela Netscape), que utiliza criptografia de chave assimétrica, tornando a comunicação entre as partes virtualmente inviolável. Desta forma, se houver interceptação das informações trafegadas entre o cliente e o servidor por parte de pessoas não autorizadas, estas informações serão de utilidade zero, já que seria necessário o conhecimento prévio das chaves privadas de criptografia.

Para que o sigilo e a inviolabilidade da comunicação realmente existam, é necessário um Certificado de Identidade Digital válido.

Formas de Pagamento Virtual

ara se comprar coisas pela internet é necessário arranjar uma forma adequada de efectuar o pagamento. No mundo real existem muitas maneiras de de pagar: dinheiro, cartões bancários, cartões de crédito, cheque, senhas, etc... Da mesma forma na internet foram criados vários sistemas de pagamento.

Cartões de Crédito

Uma das primeiras formas de pagamento na internet foi o uso de cartões de crédito. Trata-se de um sistema que já existe no mundo real, que é usado por milhões de pessoas e que permite efetuar compras em qualquer parte do mundo, desde que seja aceite pelo comerciante. Existem vários tipos de cartões, mas todos funcionam da mesma forma: o possuidor do cartão efetua um pagamento, as informações do cartão sao dadas e o dinheiro é movimentado do possuidor para o fornecedor dos serviços.

Assim o seu uso na internet é simples: basta ter um cartão de crédito, este ser aceite pelo fornecedor de serviços e enviar informações sobre o cartão para o fornecedor. O problema reside em questões de segurança : como garantir que o comprador é mesmo o dono do cartão? Para contornar este obstáculo existem várias formas. Uns usam sistemas de criptografia e autorizaçao do cartão online. Outros preferem o uso da confirmação pelo telefone ou e-mail.

Dinheiro Virtual

Como pagar para quem não possui ou não gosta de usar cartões de crédito? A resposta reside no dinheiro virtual. O dinheiro foi uma invencão espantosa pois antes todo sistema comercial se baseava em trocas. Para obter um bem era necessário trocá-lo por outro bem, o que tinha e números inconvenientes. No entanto o dinheiro só tem valor porque se lhe é reconhecido esse valor (pelo estado, entidades bancárias, ...). No início muitas pessoas prefiriam continuar com o sistema de trocas. Levou algum tempo para que os mais precavidos reconhececem o valor do dinheiro. Foram as vantagens deste (menor peso e volume) assim como uma intervenção por parte da entidade emissora que conduziram ao seu uso generalisado .

A criação de dinheiro virtual torna-se o passo seguinte nesta evolução financeira. O dinheiro virtual tem muitas vantagens: não ocupa espaço, não tem custos de emissão , não se desgasta e não se pode perder Mas para ser bem sucedido o dinheiro virtual precisa de ser seguro, rápido e simples de usar vários sistemas de dinheiro virtual foram criados, cada um tem as suas vantagens e não é claro qual será aceito. Como exemplo destes sistemas temos o NetCash, o Netbill, o Netchex, o Netcheque, o Netmarket e o Magic Money.

ecash

De momento o candidato mais provável é o ecash da empresa holandesa Digicash. Existe um banco emissor que dá o ecash aos utilizadores em troca de dinheiro real. O utilizador gasta quanto ecash quiser. Mais tarde o fornecedor de serviços pode trocar o ecash (se o quiser) depois (no tal banco) por dinheiro. Mais tarde o fomecedor de O dinheiro real. Em cada transação são usadas assinaturas digitais públicas para manter segurança. software cliente, usado pelos clientes para encriptar a transação, é gratuito e garante anonimato.Somente as lojas e serviços participantes têm de pagar uma pequena taxa e declarar todas transações (para evitar fugas ao fisco ou lavagem de dinheiro). Devido a ser um sistema simples o ecash tem bastantes probabilidades de ser bem sucedido.

NetCheque

O NetCheque, desenvolvido pela Universidade da Califórnia do Sul , usa-se da mesma forma que os cheques tradicionais. Os NetCheques são e-mails assinados pelo pagador autorizado com uma assinatura eletrônica (código criptográfico) e enviados para o receptor. Este processo é protegido pelo sistema de kerberos. A assinatura do utilizador cria o cheque enquanto que o endosso da pessoa a quem se paga o transforma numa ordem para o computador do banco.

LETSystem

Trata-se de um sistema mais ambicioso e complexo. Funciona na base de dinheiro local, tipos de unidades monetárias vagas e usadas em certas comunidades. Como exemplo desta unidades temos o Stroud, xxxx e pedras. Um utilizador nunca passa a dever dinheiro mas fica antes comprometido, i.e., devendo um serviço à comunidade. Este tipo de sistema depende bastante da confiança de todos que o usam.

PGP

PGP é um criptsistema híbrido que combina algorítmos de chaves públicas (assimétricas) com algorítmos convencionais (simétricos), com a vantagem de utilizar a velocidade da criptografia convencional e a seguranca da criptografia por chaves públicas.

As chaves públicas são mantidas em arquivos que contém a identificação do usuário (i.e. o nome da pessoa), a hora (timestamp) da geração do par de chaves e as chaves propriamente ditas. São usados dois arquivos (key rings) diferentes, um para chaves públicas e outro para as secretas, que podem conter uma ou mais chaves cada um.

As chaves públicas são internamente referenciadas por uma Key JD, que é uma abreviação dessa chave (os 64 bits menos significativos). Enquanto muitas chaves podem ter a mesma identificação do usuário (User JD), nenhuma chave pode ter a mesma Key JD. PGP faz uso de "message digest para realizar as assinaturas. "Message digest é o nome que se dá a um conjunto de 128 bits fortemente cifrados, função da mensagem. É algo análogo ao checksum ou ao CRC, que é um código verificador de erros, e representa compactamente a mensagem, usada para detectar mudanças em seu conteúdo. Diferentemente do CRC, entretanto, é computacionalmente impraticável a qualquer pessoa descobrir uma outra mensagem que produza uma mesma " message digest", que ainda é criptografada pela chave secreta para formar a assinatura digital.

Os documentos são autenticados por um prefixo que contém o Key JD da chave secreta que foi usada para assiná-lo, o " message digest" do documento devidamente criptografado pela chave secreta doremetente e a hora (timestamp) de quando foi realizada a assinatura. O Key ID é utilizado pelo destinatário para relacioná-lo com a chave públicado remetente, afim de checar a assinatura. O software automaticamente procura a chave pública e a identificação do usuário remetente no arquivo de chaves públicas.

Arquivos cifrados são prefixados pelo Key ID da chave pública usada para cifrá-la. O receptor usa essa informação para relacionar a correspondente chave secreta que decifra a mensagem. Da mesma forma, o software do destinatário automaticamente localiza essa chave secreta no arquivo de chaves secretas.Esses dois tipos de arquivos são o principal método de armazenamento e gerenciamento das chaves públicas e privadas.

Vulnerabilidades

Nenhum sistema de segurança é impenetrável. PGP pode ser enganado de várias formas diferentes. Em qualquer sistema de segurança, temos que nos perguntar se a informação que escondemos é mais valiosa para um eventual agressor do que o custo que este teria para burlar o sistema de proteção. Isto ajudaria a nos proteger de ataques de baixo custo sem nos preocuparmos com meios mais sofisticados e caros de espionagem.

l - Comprometimento da passphrase e chave secreta:

Provavelmente o modo mais simples de quebrar o sistema é escrever em algum lugar sua passphrase. Se alguém a achar e também conseguir copiar seu arquivo de chave secreta, pode tranquilamente ler suas mensagens e enviar outras tantas com a sua assinatura, fazendo com que todos pensem que foi você o autor das tais mensagens.

Não usar passwords simples que possam ser facilmente descobertas como os nomes de esposa ou filhos já ajuda. Se você fizer de sua passphrase uma única palavra (tornando-se uma pas,S.word), ela poderá ser descoberta por um computador que teste todas as palavras do dicionário. Por isso uma passphrase é melhor do que uma password É claro que um agressor mais sofisticado poderia ter em seu computador um banco de dados com frases famosas para tentar achar sua passphrase. Uma passphrase fácil de lembrar e difícil de se descobrir poderia ser construída por alguns dizeres criativos sem sentido algum ou referências literárias obscuras.

2 - Falsificação da chave pública:

Este pode ser o ponto mais vulnerável de um criptossistema de chave pública, principalmente porque a maioria dos novatos na área não percebem a falsificação imediatamente. Quando você usar a chave públicade de alguém, esteja certo de que não foi falsificada. Só podemos confiar na chave pública de uma pessoa se a obtivermos diretamente dessa pessoa, ou se a recebemos em uma mensagem assinada por alguém em quem confiamos. Mantenha um controle fisico de seus arquivos de chave pública e secreta, mais preferivelmente em seu computador pessoal do que naqueles ligados em rede com acesso remoto.Tenha sempre uma cópia de ambos os arquivos.

3 - Arquivos não apagados completamente do disco:

Outro problema potencial é causado pelo modo de deleção de arquivos da maioria dos sistemas operacionais. Quando você criptografa um arquivo e apaga o texto original, o sistema operacional não destrói fisicamente o conteúdo do texto original. Ele apenas marca que aqueles blocos foram apagados do disco, permitindo que esse espaço seja reutilizado posteriormente. Esses blocos marcados ainda contém o texto original que queríamos destruir, e que será realmente apagado apenas quando outros dados forem gravados por cima. Se o agressor tiver acesso aos blocos marcados antes que eles sejam regravados, ele pode recuperar o texto original ou pelo menos parte dele, o que pode ser facilmente realizado com programas especializados.

De fato isto poderia até acontecer acidentalmente, se por alguma razão alguma coisa der errado na estrutura lógica de armazenamento de dados do disco e alguns arquivos forem acidentalmente apagados ou corrompidos. Programas de recuperação podem ser utilizados para tentar reaver os arquivos danificados, mas isto frequentemente faz com que arquivos previamente deletados sejam ressucitados junto com os arquivos que realmente interessam. Daí, aquele arquivo confidencial que você achava ter sumido para todo o sempre pode reaparecer e ser lido pela pessoa que estava tentando consertar seu disco. Mesmo quando você está escrevendo sua mensagem com um processador de textos,o editor estar gerando várias cópias temporárias do seu texto no disco, apenas porque é texto ele trabalha internamente. Essas cópias temporárias são apagadas pelo próprio assim processador quando o finalizamos, mas os seus fragmentos ficam no disco, em algum lugar.

A única forma de se ter certeza que nossos textos originais não reaparecerão de uma hora pra outra, é, de algum jeito, apagar fisicamente do disco o conteúdo da mensagem. A menos que você tenha certeza de que todos os blocos marcados como apagados serão reutilizados brevemente, pode-se tomar providências para reescrevê-los, e assim, destruir qualquer vestígio do texto sigiloso deixado pelo seu editor de textos favorito. Pode-se fazê-lo utilizando qualquer programa especializado disponível no mercado, como por exemplo o Norton Utilities para DOS (WIPEINFO.EXE).

Mesmo tomadas todas as providências descritas acima, tavez ainda seja possível recuperar o conteUdo original do texto por alguém cheio de recursos. Resíduos magnéticos dos dados originais ficam no disco mesmo após serem regravados. Algum hardware sofisticado de recuperação de dados pode algumas vezes ter utilidade para reaver os dados (mais aí já é demais).

4 - Víroses e Cavalo de Tróia:

Outro tipo de ataque poderia envolver um vírus especial que infectaria o PGP ou o seu sistema operacional. Este vírus hipotético seria projetado para capturar sua passphrase ou chave secreta, e gravá-la em algum arquivo no disco ou enviá-la via rede para o 'proprietário' do referido vírus. Ele poderia até alterar o comportamento do PGP para que as assinaturas não sejam checadas corretamente, por exemplo.

Defender-se desse tipo de ataque nada mais é do que evitar contaminação por virus em geral. PGP não tem defesas contra vírus, ele assume que a sua máquina está livre de vermes, o que pode ser pelo menos tentado por versões atualizadas de produtos anti-vírus disponíveis no mercado. Se por acaso um vírus especial contra PGP aparecer, esperamos que todos tomemos conhecimento logo.

Outra forma de espionagem envolveria uma cópia do PGP parecida visualmente com a original, mas alterada para que não ftlncionasse a contento. Por exemplo, alguém poderia deliberadamente alterá-lo para que não checasse as assinaturas corretamente, permitindo a falsificação das mesmas. Esta versão tipo "Cavalo de Tróia" não seria difícil de ser desenvolvida, pois o código fonte do PGP é amplamente divulgado. Para evitar tais problemas, deve-se confiar na fonte de onde foi adquirida a cópia do PGP, ou pegá-la de várias fontes independentes e compará-las com um utilitário destinado para esse fim.

5 - Falha de segurança física:

Uma descuido do próprio usuário poderia permitir a alguém adquirir seus arquivos originais ou mensagens impressas. Um oponente determinado poderia utilizar meios tais como roubo, vasculhamento de lixo, sequestro, suborno, chantagem ou infiltração entre os funcionários.

Não se iluda com a falsa sensação de segurança só porque você tem uma ferramenta de criptografia. As técnicas criptográficas protegem a informação somente quando elas estão criptografadas, violações fisicas de segurança podem comprometê-las.

6 - Espionagem "Tempest".

Outra forma de espionagem tem sido utilizada por oponentes muito bem equipados que conseguem detectar os sinais eletromagnéticos emitidos pelo computador. Este tipo de ataque caro e de intensa monitoração pode ser realizado por um caminhão suprido da maquinaria necessária, estacionado próximo ao seu local de trabalho e remotamente captando todos os seus toques de teclado, assim como os textos jogados na tela, podendo comprometer seus passwords, mensagens, etc. Este tipo de ataque, conhecido como tempest, poderia ser evitado blindando-se computadores e cabos de rede, de modo que esses sinais não sejam mais emitidos.

7 - Análise de tráfico :

Mesmo que não seja possível ler o conteúdo das mensagens criptografadas, alguma informação útil ainda poderia ser deduzida observando-se de onde as mensagens chegam e para onde elas vão, o tamanho das mesmas e a hora em que foram enviadas. É a mesma coisa que olhar para uma conta telefônica de longa distância e ver para onde você ligou, quando e por quanto tempo ficou conversando, mesmo que o conteúdo da conversa seja desconhecido para um possível espião. A isto chamamos de análise de tráfico. Para resolver este tipo de problema, precisaríamos de protocolos de comunicação especialmente desenvolvidos para reduzir à exposição dessas análises, possivelmente com alguma assistência criptográfica.

8 - Exposição em sistemas multi-usuário:

PGP foi originalmente projetado para rodar em máquinas MSDOS mono-usuárias, sob seu controle físico direto. Mas agora exitem versões do PGP que também rodam em sistemas multi-usuários como UNIX e VAXlVMS. Nesses sistemas, os riscos de descobrirem seus passwords, chaves secretas ou mensagens são maiores. Um intruso esperto o suficiente ou o próprio administrador de rede poderiam ter acesso aos seus arquivos originais, ou talvez utilizar-se de algum programinha especial para monitorar constantemente a digitação ou ver o que está aparecendo em sua tela. Os riscos reais de segurança dependem de cada situação em particular Alguns sistemas multi-usuários podem ser considerados seguros porque todos os usuários são confiáveis, ou porque não há interesse suficiente em espionar alguém. De qualquer modo, recomenda-se rodar o PGP de uma máquina isolada, em um sistema mono-usuário, diretamente sob seu controle físico.

Sistemas Criptográficos baseados em Curvas Elípticas

Introdução

Em 1985, Koblitz [1] e Miller [2] sugeriram, independentemente, que curvas elípticas poderiam ser usadas para esquemas de criptografia de chave pública. Desde então um grande esforço tem sido feito em busca de caminhos que tornem eficiente a sua computação. A matemática associada a curvas elípticas, entretanto, é mais antiga. Há mais de 100 anos esta matéria vem sendo estudada em teoria dos números e geometria algébrica. Pretende-se, neste trabalho, abordar, de maneira introdutória, sistemas baseados em curvas elípticas e demonstrar que estes sistemas proporcionam blocos de tamanho relativamente pequenos, permitem implementações em hardware e software de alta velocidade e oferecem a maior resistência a ataques por bit de chave (strength-per-key-bit) dentre os esquemas conhecidos de chave pública.

Curvas Elípticas sobre Números Reais

Para melhor entendimento do conceito matemático das curvas elípticas, iniciemos nosso estudo no campo dos reais. A equação abaixo mostra a forma de "Weierstrass" de uma curva elíptica:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

As variáveis x e y situam-se no plano. Na verdade, x e y podem ser complexos, reais, inteiros, base polinomial, base canônica ou qualquer outro tipo de elemento de corpo. Mas, consideremos números reais sobre o plano dos reais, que nos é mais familiar. Uma forma simples da equação (1) é:

$$y^2 = x^3 + a_4x + a_6 \quad (2)$$

Como exemplo, vamos representar o gráfico da curva para $a_4 = -7$, $a_6 = 5$ com x e y no conjunto dos números reais:
números reais:

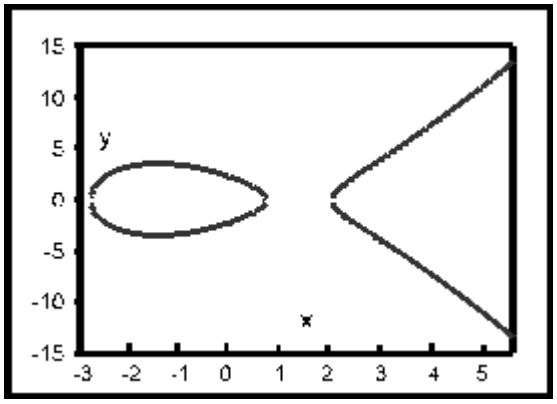


Figura 1 - Gráfico da curva elíptica $y^2=x^3-7x+5$

Nossa intenção é definir uma álgebra para curvas elípticas. Assim, devemos encontrar uma maneira de definir "adição" de dois pontos da curva, cuja soma seja outro ponto da curva. Além disso, devemos definir o elemento identidade da soma $\boldsymbol{O}$, ponto que somado com qualquer outro da curva, resulte no próprio ponto:

$$P + \boldsymbol{O} = P \quad (3)$$

Este ponto também é chamado de ponto no infinito.

Para a álgebra funcionar, o negativo do ponto de interseção é definido como a "soma elíptica" (veja figura 2). Matematicamente:

$R = P + Q$(4)

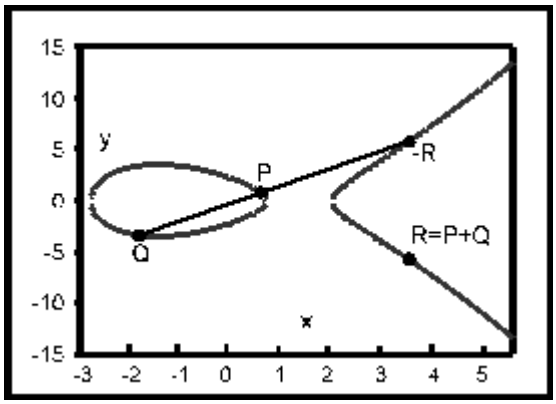


Figura 2 – Adição de pontos de uma curva elíptica sobre números reais

Adicionar um ponto a ele mesmo é um caso especial. A linha usada é a tangente à curva no ponto considerado.

Curvas Elípticas sobre Corpos Finitos Primos

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, se\ P \neq Q \\ \dfrac{3x_1^2 + a_4}{2y_1}, se\ P = Q \end{cases}$$

Para utilização em criptografia interessa-nos estudar a matemática de curvas elípticas aplicadas a corpos finitos. Analisaremos, inicialmente, corpos finitos gerados por grandes primos. Ou seja, analisaremos curvas elípticas sobre *Zp*, *p* primo maior que 3.

Uma curva elíptica *E* sobre *Zp* pode ser definida pela mesma equação (2) estudada no item anterior:

$$y^2 = x^3 + a_4 x + a_6$$

onde a_4, a_6 Î *Zp* e $4a_4^3 + 27a_6^2 \neq 0$. O conjunto *E(Zp)* é composto, então, por todos os pontos *(x,y)*, $x \in Zp$, $y \in Zp$, que satisfazem a equação de definição, juntamente com o ponto no infinito ***O***.

Por exemplo: seja $p = 23$ e considere a curva elíptica $E: y^2 = x^3 + x + 1$, definida sobre Z_{23}. Note que $4a_4^3 + 27a_6^2 = 4 + 4 = 8 \neq 0$, então *E* é uma curva elíptica. Os pontos em $E(Z_{23})$ são ***O*** e os seguintes:

(0, 1) (0, 22) (1, 7) (1, 16) (3, 10) (3, 13) (4, 0) (5, 4) (5, 19) (6, 4) (6, 19) (7, 11) (7, 12) (9, 7) (9, 16) (11, 3) (11, 20) (12, 4) (12, 19) (13, 7) (13, 16) (17, 3) (17, 20) (18, 3) (18, 20) (19, 5) (19, 18)

Vejamos a regra para adicionar dois pontos em uma curva elíptica *E(Zp)* para resultar em um terceiro ponto da curva. Junto com esta operação de adição, o conjunto de pontos *E(Zp)* forma um grupo, com ***O*** servindo como sua identidade. É este grupo que é utilizado na construção de sistemas de criptografia baseados em curvas elípticas. A regra de adição é apresentada abaixo como uma seqüência de fórmulas algébricas:

1. $P + O = O + P = P$ para todo $P \in E(Zp)$
2. Se $P = (x, y) \in E(Zp)$, então $(x, y) + (x, -y) = O$. (O ponto $(x, -y)$ é representado por $-P$ e é chamado negativo de *P*. Observe que $-P$ é, também, um ponto na curva.)
3. Seja $P = (x_1,y_1) \in E(Zp)$ e $Q = (x_2,y_2) \in E(Zp)$, onde $P \neq -Q$. Então $P + Q = (x_3,y_3)$, onde:

$$x^3 = \lambda^2 - x_1 - x_2 \quad (5)$$

$$y^3 = \lambda(x_1 - x_3) - y_1 \quad (6)$$

e

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, se\ P \neq Q & (7) \\ \dfrac{3x_1^2 + a_4}{2y_1}, se\ P = Q & (8) \end{cases}$$

Vamos ver um exemplo de adição de curva elíptica. Considere a curva elíptica definida no exemplo anterior.

1. Seja $P = (3, 10)$ e $Q = (9, 7)$. Então $P + Q = (x_3, y_3)$ é calculado como segue:

$$\lambda = \frac{7-10}{9-3} = \frac{-3}{6} = \frac{-1}{2} = 11 \in Z_{23}$$

$x_3 = 112 - 3 - 9 = 6 - 3 - 9 = -6 \in 17 \pmod{23}$, e

$y_3 = 11(3 - (-6)) - 10 = 11(9) - 10 = 89 \in 20 \pmod{23}$.

Portanto, $P + Q = (17,20)$.

2. Seja $P = (3,10)$. Então $2P = P + P = (x_3, y_3)$ é calculado como segue:

$$\lambda = \frac{3(3^2)+1}{20} = \frac{5}{20} = \frac{1}{4} = 6 \in Z_{23}$$

$x_3 = 62 - 6 = 30 \in 7 \pmod{23}$, e

$y_3 = 6\,(3 - 7) - 10 = -24 - 10 = -11 \in 12 \pmod{23}$.

Portanto, $2P = (7,12)$.

Curvas Elípticas sobre Corpos Finitos de Característica Dois.

Os corpos finitos de característica dois, $GF(2^m)$, interessam especialmente, pois permitem implementações eficientes da aritmética de curvas elípticas. Neste caso, as constantes são números de base polinomial ou canônica. Não podemos, neste caso, utilizar a versão simplificada da equação (1).

Menezes [3] afirma que é necessário uma das duas versões abaixo:

$$y^2 + y = x^3 + a_4 x + a_6 \quad (9)$$

$$y^2 + xy = x^3 + a_2 x^2 + a_6 \quad (10)$$

A equação (9) é da forma "supersingular" e, embora possa ser computada rapidamente, suas propriedades não a tornam indicadas para o uso em criptografia.

As curvas da equação (10) são chamadas "nonsupersingular". Não existe nenhum método de ataque conhecido de complexidade menor que exponencial para estas curvas. Certamente, a escolha dos coeficientes é fundamental, a fim de que se obtenha a máxima vantagem da segurança.

Para que a equação (10) seja válida, a_6 precisa ser diferente de 0. Contudo, a_2 pode ser 0. Valem aqui as mesmas regras de adição vistas para corpos primos. As fórmulas, contudo, são um pouco diferentes para adição de dois pontos sobre $GF(2^n)$, segundo Schroeppel et

al.[4]:

Se $P \neq Q$:

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_2 \qquad (11)$$

$$y_3 = \lambda(x_1 + x_3) - y_1 \qquad (12)$$

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \qquad (13)$$

Se $P = Q$:

$$x_3 = \lambda^2 + \lambda + a_2 \qquad \textbf{(14)}$$

$$y_3 = x_1^2 + (\lambda + 1)x_3 \qquad \textbf{(15)}$$

$$\lambda = x_1 + \frac{y_1}{x_1} \qquad \textbf{(16)}$$

Multiplicação sobre Curvas Elípticas

A multiplicação sobre curvas elípticas se refere, ao contrário da idéia intuitiva de se multiplicar dois pontos da curva, ao produto de um escalar por um ponto da curva:

$$Q = kP \quad (17)$$

Onde Q e P são pontos sobre uma curva elíptica e k é um inteiro. O que a multiplicação realmente significa é a soma de Pa ele mesmo k vezes. Como a própria curva elíptica – isto é, os pontos sobre ela – forma um corpo, o inteiro k não deve ser maior que a ordem do ponto P. Caso não se saiba a ordem do ponto, o cálculo não será tão eficiente quanto poderia ser.

Exemplo: suponha que desejamos calcular $Q = 15P$. Podemos expandir como:

$$Q = 15P = P + 2(P + 2(P + 2P))$$

Observe que este é o mesmo algoritmo utilizado para exponenciação modular.

Outro método para o cálculo da multiplicação é a expansão balanceada, proposta por Koblitz [5]. O algoritmo converte uma string de bits "1" em uma string de bits "0" seguido de "–1". Por exemplo, calculemos $Q = 15P$:

$$15P = (16\text{-}1)P$$
$$11112P = (1000\text{-}1)2P$$
$$Q = (2P)2 * 2 * 2 - P$$

Temos, assim cinco operações, ao invés de seis, como no caso anterior.

Outro exemplo: $Q = 10045P = 100111001111012P$. O último bit na cadeia de "1" é substituído por "–1", todos os outros bits são substituídos por "0" e o primeiro "0" é substituído por "1". Assim, a representação balanceada fica:

10100-101000-101

*Q = (((((2P * 2 + P)2 * 2 * 2 – P)2 * 2 + P)2 * 2 * 2 *2 – P)2 * 2 + P*

Ordem da Curva

O número de pontos de uma curva elíptica sobre um corpo finito deve satisfazer o teorema de Hasse. Dado um campo, *GF(q)*, a ordem da curva *N* deverá satisfazer esta equação:

$$\left|N-(q+1)\right| \le 2\sqrt{q} \quad (18)$$

Ou, de outra forma:

$$q+1-2\sqrt{q} \le N \le q+1+2\sqrt{q} \quad (19)$$

Então, o número de pontos de uma curva é, aproximadamente, o tamanho do corpo.

Criptografia

Uma vez introduzida a matemática de curvas elípticas sobre corpos finitos, analisaremos como aplicá-la à criptografia. A idéia básica para a implementação de sistemas de criptografia de chaves secretas e públicas é converter dados em pontos da curva.

Duas questões fundamentais a serem respondidas por quem pretende utilizar curvas elípticas em criptografia são:

Quantos bits são necessários?

Quanto trabalho deve ser realizado?

A decisão sobre a primeira questão determina quanto tempo se leva para computar uma chave secreta de curva elíptica. Pode-se considerar, de forma empírica, que o tempo da computação é proporcional ao cubo do número de bits. Assim, se dobrarmos o número de bits, levaremos, aproximadamente, 8 vezes mais tempo calculando.

Escolha da Curva

A segunda decisão determina como deve ser escolhida a curva elíptica. O primeiro passo é certificar-se de que há mais bits no corpo finito que o comprimento dos dados que desejamos esconder. Para a máxima segurança, uma boa prática é tomar-se 2,5 vezes a quantidade de bits que se deseja embutir. O fator 2 vem da habilidade de se quebrar curvas elípticas usando a raiz quadrada da ordem do ponto base. O fator 0,5 deve ser usado se não se conhece a ordem do ponto base. Por exemplo, suponha que queiramos usar uma chave secreta de 64 bits. O tamanho de corpo mais próximo para uma base canônica estaria entre 130 e 148 bits.

Como foi citado anteriormente, devem ser evitadas curvas supersingulares porque o problema do logaritmo discreto pode ser reduzido ao problema do logaritmo discreto sobre um corpo extensão de grau menor.

Já vimos, da equação (10), que uma curva nonsupersingular tem a forma:

$$y^2 + xy = x^3 + a_2 x^2 + a_6$$

para corpos $GF(2^n)$. Quando a_2 é diferente de zero, a curva é chamada “twist”. Por exemplo $y^2 + xy = x^3 + x^2 + 1$, onde corpos de tamanho 101, 107, 109, 113 e 163 tem ordem 2 vezes um primo muito grande. Para uma curva "nontwist", por exemplo $y^2 + xy = x^3 + 1$, os corpos 103, 107 e 131 todos tem ordem 4 vezes um primo muito grande. As tabelas 1 e 2, a

seguir, apresentam a lista de fatores para estas curvas. O símbolo*#E* representa a ordem de uma curva elíptica.

Tamanho do Corpo	*#E*
101	2 x 1 26765 06002 28230 88614 28085 08011
107	2 x 81 12963 84146 06692 18285 10322 12511
109	2 x 324 51855 36584 26701 4874486564 61467
113	2 x 5192 29685 85348 27627 89670 38334 67507
163	2 x 5846 00654 93236 11672 81474 17535 98448 34832 91185 74063

Tabela 1 – $E: y^2 + xy = x^3 + x^2 + 1$

Tamanho do Corpo	*#E*
103	4 x 2 53530 12004 56459 53586 25300 67069
107	4 x 40 56481 92073 03335 60436 34890 37809
131	4 x 6805 64733 84187 69269 32320 12949 34099 85129

Tabela 2 – $E: y^2 + xy = x^3 + 1$

Para se obter alta segurança, curvas de ordem conhecida e de difícil ataque, como as apresentadas, são necessárias. Para uma segurança moderada, entretanto, curvas aleatórias são perfeitamente adequadas.

Embutindo Dados numa Curva

Antes de computarmos somas e multiplicações sobre curvas elípticas, precisamos colocar os dados sobre a curva. Se tomarmos valores aleatoriamente, devemos nos certificar que o valor de *x* que tomamos satisfaz, de fato, a equação da curva. Vamos reescrever a

equação (10), convertendo o lado direito para uma forma simples, *f(x)*, e trazendo-o para o lado esquerdo:

$$y^2 + xy + f(x) = 0 \quad (20)$$

Esta é uma equação quadrática simples. O primeiro passo para resolvê-la é eliminar *x*. Seja $y = xz$ na equação (20). Isto nos dá:

$$(xz)^2 + x^2 z + f(x) = 0 \quad (21)$$

Multiplicando a equação por x^{-2}, temos:

$$z^2 + z + c = 0 \quad (22)$$

onde $c = f(x).x^{-2}$. A equação (22) tem solução quando o Traço de *c* é *0*. A função Traço mapeia $GF(2^m)$ em $GF(2)$. O resultado é um bit. Ela é definida por:

$$Tr(c) = c + c^{2^1} + c^{2^2} + \ldots + c^{2^{m-1}} \quad (23)$$

Por exemplo, seja *c = 10110* em *GF(25)*.
Logo,

$$Tr(10110) = 10110 + 01101 + 11010 + 10101 + 01011$$

A operação XOR de todas as colunas dá o mesmo resultado:

$$Tr(10110) = 1$$

Uma vez que o $Tr(c) = 0$, podemos resolver a equação para *z*. Conhecido *z*, temos que *z+1* também é uma solução para a equação (22). Então, colocando *z* e *z+1* na equação $y = zx$ teremos as duas soluções para a equação quadrática e os dados estarão embutidos na curva. Com os dados na curva podemos começar a manipular estes dados de uma forma significativa, sob o ponto de vista criptográfico. Vamos ressaltar novamente que o conjunto de dados que desejamos embutir na curva precisa ter menos bits que o tamanho do corpo para assegurar a escolha de valores *x*, tal que esses valores estejam na curva

Sistemas Criptográficos baseados em Curvas Elípticas
Protocolos

Koblitz [6] sugere o uso de curvas elípticas em três protocolos de chave pública: Diffie-Hellman, ElGamal e Massey-Omura. Analisaremos os dois primeiros. Em seguida, analisaremos o uso de curvas elípticas para assinatura digital, usando DSA.

Protocolo Diffie-Hellman

Sejam os usuários Alexandre e Bia, que desejam compartilhar um parâmetro secreto. Ambos devem escolher sua chave secreta que será usada como modelo de bits aleatórios sobre um corpo de tamanho mutuamente combinado. Seja k_A a chave de Alexandre e k_B a chave de Bia. Eles, primeiramente, devem combinar o uso de uma curva específica, de um tamanho de corpo e a base matemática (polinomial ou canônica). A curva elíptica deve satisfazer a equação (10). Eles ainda devem escolher um ponto base *B* comum, sobre a curva combinada. Bia, então, computa:

$$P_B = k_B B \quad (24)$$

sobre a curva elíptica escolhida e envia para Alexandre. Alexandre computa:

$$P_A = k_A B \quad (25)$$

e envia para Bia. Ambos, então, computam o parâmetro secreto:

$$PS = k_A(k_B B) = k_B(k_A B) \quad (26)$$

Por questão de segurança, apenas o valor de x obtido e um bit do valor de y são utilizados, uma vez que podemos obter y através da solução da equação quadrática vista anteriormente.

Cada lado, então, possuirá o mesmo parâmetro secreto e ninguém que estiver ouvindo a comunicação será capaz de conhecer este parâmetro, sem resolver o problema do logaritmo discreto da curva elíptica.

O problema com Diffie-Hellman é o ataque chamado de "man-in-the-middle". Suponha que Melo consiga interceptar toda a comunicação entre Alexandre e Bia. Ele, então, pega a chave pública de Bia por um lado e de Alexandre pelo outro e manda sua própria chave pública para ambos. Nem Bia nem Alexandre são capazes de perceber a intercepção, uma vez que eles conseguem cifrar e decifrar normalmente as mensagens. Melo decifra os dados vindos de Alexandre com o parâmetro secreto que compartilha com o mesmo, e depois cifra-os novamente usando o parâmetro que compartilha com Bia, enviando os dados para ela. Obviamente, Melo pode alterar os dados da maneira que quiser antes de retransmití-los.

Protocolo ElGamal

A versão de curva elíptica de ElGamal requer que sejam públicos o tamanho de corpo, a base matemática (polinomial ou canônica), a curva elíptica E que satisfaça a equação (10) e um ponto base B sobre a curva. Tanto Alexandre quanto Bia escolhem aleatoriamente um modelo de bits k_A ek_B, respectivamente, e calculam suas chaves públicas:

$$P_A = k_A B \quad (27)$$

$$P_B = k_B B \quad (28)$$

Cada um envia sua chave pública ao outro. Agora, Alexandre e Bia podem enviar mensagens um ao outro usando esses pontos públicos, de tal forma que ninguém pode descobrir os dados sem resolver o problema do logaritmo discreto para curvas elípticas.

Para que Alexandre mande uma mensagem para Bia, ele primeiro embute a informação da mensagem na curva elíptica E, usando o método de conversão mostrado anteriormente. Vamos chamar este ponto da mensagem P_m. Alexandre, então, escolhe um modelo aleatório de bits, r, e computa dois pontos:

$$P_r = rB \quad (29)$$

$$P_h = P_m + rB \quad (30)$$

Alexandre envia, então, ambos os pontos P_re$P_h$para Bia. Para extrair o ponto mensagem, Bia computa:

$$P_s = k_B P_r \quad (31)$$

E subtrai este de P_h:

$$P_m = P_h - P_s \quad (32)$$

Vamos expandir esta última equação para ver como ela funciona. Na equação (30), o segundo termo do lado direito pode ser expandido usando a equação (28):

$$rP_B = r(k_B B) \quad (33)$$

O termo $P_h$na equação (32) é, na verdade:

$$P_h = P_m + r(k_B B) \quad (34)$$

Colocando a equação (29) na equação (31) para expandirP_S, e combinando esta com a equação (34) na equação (32), temos:

$$P_m = P_m + r(k_B B) - k_B(rB) = P_m \quad (35)$$

Os únicos pontos que o atacante vê são P_h, P_A, P_BeP_r. O interessante deste protocolo é que as chaves públicas podem permanecer públicas. Não há necessidade trocá-las. Toda vez que os dados são transferidos, um novo valor aleatório *r*é escolhido. Nenhum dos lados precisa lembrar de *r*, e, se o tamanho do corpo for suficientemente grande, será bastante difícil descobrir os números secretos k_Aek_B. Isto nos fornece tanto compartilhamento do parâmetro secreto quanto autenticação.

O ataque do Melo ("man-in-the-middle") ainda é possível. A maneira mais segura de evitar este tipo de ataque é pela verificação da chave pública através de um canal alternativo, como um telefone.

Assinatura Digital

Analisaremos, agora, um esquema de assinatura digital, que é a versão para curvas elípticas do DSA (Digital Signature Algorithm).

Seja P um ponto base com ordem *n* na curva *E*, que satisfaz a equação (10). Chamaremos a chave privada do assinante *s* e a chave pública $Q = sP$. Tomemos um valor aleatório *k* e um ponto aleatório $R = kP$. Gera-se uma função hash *e* a partir da mensagem, tal que *e* seja menor que *n*. O primeiro passo do DSA é tomar a componente *x* de *R* módulo a ordem da curva para pegar o primeiro componente da assinatura:

$$c = x \bmod n \quad (36)$$

O segundo componente é computado como:

$$d = k^{-1}(e+sc) \quad (37)$$

O processo de verificação requer o cálculo de três valores depois de computar o valor hash da mensagem (chamado *e'*):

$$h = d^{-1} \bmod n \quad (38)$$
$$h_1 = e'h \bmod n \quad (39)$$
$$h_2 = ch \bmod n \quad (40)$$

Estes valores são usados para computar um ponto sobre a curva elíptica pública com a fórmula:

$$R' = h_1 P + h_2 Q \quad (41)$$

Se a componente *x* da equação (41) não for igual ao da equação (36), assumimos que a mensagem é diferente do documento original assinado. Vamos entender a razão. A equação (38) pode ser reescrita com a equação (37):

$$h = k(e+sc)^{-1} \quad (42)$$

Assim, as equações (39) e (40) expandem-se para:

$$h_1 = e'k(e+sc)^{-1} \quad (43)$$
$$h_2 = ck(e+sc)^{-1} \quad (44)$$

Colocando os termos expandidos na equação (41):

$$R' = e'k(e+sc)^{-1}P + sck(e+sc)^{-1}P \quad (45)$$
$$R' = k(e'+sc)(e+sc)^{-1}P \quad (46)$$

Os fatores do meio só serão cancelados se o valor hash da mensagem original, a chave do assinante e a assinatura publicada estiverem corretos.

Sistemas Criptográficos baseados em Curvas Elípticas

Aspectos de Segurança

A base para a segurança de sistemas criptográficos baseados em curvas elípticas é a aparente intratabilidade do "problema do logaritmo discreto de curvas elípticas", que pode ser resumido da seguinte forma: dada uma curva elíptica E definida sobre um corpo finito, um ponto P da curva de ordem n, e um ponto$Q$, determine o inteiro k, $0 \le k \le n-1$, tal que $Q = kP$.

Nos últimos 12 anos, o problema de logaritmos discretos de curvas elípticas tem recebido considerável atenção de matemáticos no mundo inteiro, e nenhum fraqueza significativa foi relatada. Um algoritmo devido a Pohlig e Hellman reduz a determinação de k à determinação de k módulo cada um dos fatores primos de n. Logo, para se obter o máximo nível de segurança,$n$deve ser primo.

O melhor algoritmo para quebrar os protocolos baseados em curvas elípticas, segundo Jurisic e Menezes [7], em geral, é o método rho de Pollard que leva, aproximadamente, $\sqrt{\pi n / 2}$ passos, onde um passo significa uma adição de curva elíptica. Em 1993, Oorschot e Wiener [8] mostraram como o método rho de Pollard pode ser distribuído em paralelo, tal que, se r processadores forem usados, o número esperado de passos por cada processador antes de um único logaritmo discreto ser obtido é

$$\sqrt{\pi n / 2} / r \quad (47)$$

Em seu trabalho, eles estimam que um atacante (bem financiado), considerando $n \approx 10^{36} \approx 2^{120}$, precisaria de uma máquina com 325.000 processadores (a um custo de US$10 milhões à época) para computar um único logaritmo discreto em 35 dias.

Comparemos, agora, a complexidade de algoritmos para quebra da chave. Algoritmos baseados em fatoração de inteiros, como o RSA, tem ataques bem conhecidos. O problema básico para este tipo de criptografia de chave pública é se encontrar dois primos grandes (p eq), cujo produto seja um inteiro grande N. O tempo necessário para fatorar este tipo de número é:

$$tempo_{fatoracao} \approx \exp(c\sqrt[3]{(\log N)(\log\log N)^2}) \quad (48)$$

Isto ocorre para um método em particular (por sinal, usando curvas elípticas). Pode haver métodos mais rápidos, em função do tamanho de N. Mas, de maneira geral, podemos considerar o problema da fatoração com complexidade:

$$\exp\left((\log N)^{\frac{1}{3}}\right) \quad (49)$$

Sua complexidade recai, assim, na classificação de sub-exponencial.

Sistemas criptográficos baseados em curvas elípticas usam pontos ou pares de números para esconder a informação. A idéia básica é que se tenha um número total de pontos disponíveis (m) bastante grande. O tempo necessário para encontrar um particular ponto é, aproximadamente, o seguinte:

$$tempo_{eliptica} \approx \exp(c\sqrt{m}) \quad (50)$$

Este tempo é completamente exponencial.

O resultado é que, segundo Rosing [9], embora a comparação não seja trivial, a quantidade de bits que um sistema usando curvas elípticas necessita, para ser compatível

com o RSA 1024 bits, está abaixo de 200 bits. Assim, apesar de precisar de dois elementos de 200 bits para representar cada ponto, a quantidade de hardware necessária para se atingir o mesmo nível de segurança é menor que o necessário para o mesmo nível de segurança usando métodos de fatoração de inteiros.

Referências Bibliográficas

[1] KOBLITZ, N. Elliptic Curve Cryptosystems. Mathematics of Computation (1987).

[2] MILLER, V. S. Use of Elliptic Curves in Cryptography em CRYPTO'85. (New York: Springer-Verlag,1986).

[3] MENEZES, A. J. Elliptic Curve Public Key Cryptosystems. (Boston: Kluwer Academic Publishers, 1993).

[4] SCHROEPEL, R.; ORMAN, H.; O'MALLY, S. Fast Key Exchange with Elliptic Curve Systems. (Tucson, AZ: University of Arizona, Computer Sciences Department, 1995).

[5] KOBLITZ, N. CM – Curves with Good Cryptographic Properties em CRYPTO'91 (New York: Springer-Verlag, 1992).

[6] KOBLITZ, N. A Course in Number Theory and Cryptography. (New York: Springer-Verlag, 1987).

[7] JURISIC, A.; MENEZES, A. J. Elliptic Curves and Cryptography. (Ontario: Certicom White Paper em http://www.certicom.ca/ecc/weccrypt.htm, 1997).

[8] OORSCHOT, P. van; WIENER, M. Parallel Collision Search with Cryptanalytic Applications. (Journal of Cryptography, 1993).

[9] ROSING, M. Implementing Elliptic Curve Cryptography. (Greenwich: Manning, 1998).

Bibliografias

Advances in Cryptology – **EUROCRYPT '94**, Perugia, Italy. Springer-Verlag LNCS 950 (1995).
Editor: A. De Santis

M. Bellare, P. Rogaway, Optimal asymmetric encryption, 92–111.
E. Biham, On Matsui's linear cryptanalysis, 341–355.
E. Biham, A. Biryukov, An improvement of Davies' attack on DES, 461–467.
C. Blundo, A. Cresti, Space requirements for broadcast encryption, 287–298.
C. Blundo, A. Giorgio Gaggia, D.R. Stinson, On the dealer's randomness required in secret sharing schemes,
35–46.
M. Burmester, Y. Desmedt, A secure and efficient conference key distribution system, 275–286.
C. Cachin, U.M. Maurer, Linking information reconciliation and privacy amplification, 266–274.
J.L. Camenisch, J.-M. Piveteau, M.A. Stadler, Blind signatures based on the discrete logarithm problem,
428–432.
F. Chabaud, On the security of some cryptosystems based on error-correcting codes, 131–139.
F. Chabaud, S. Vaudenay, Links between differential and linear cryptanalysis, 356–365.
C. Charnes, L. O'Connor, J. Pieprzyk, R. Safavi-Naini, Y. Zheng, Comments on Soviet encryption algorithm,
433–438.
D. Chaum, Designated confirmer signatures, 86–91.
L. Chen, I.B. Damg°ard, T.P. Pedersen, Parallel divertibility of proofs of knowledge, 140–155.
L. Chen, T.P. Pedersen, New group signature schemes, 171–181.
L. Csirmaz, The size of a share must be large, 13–22.
S. D'Amiano, G. Di Crescenzo, Methodology for digital money based on general cryptographic tools,
156–170.
F. Damm, F.-P. Heider, G. Wambach, MIMD-factorisation on hypercubes, 400–409.
P. de Rooij, Efficient exponentiation using precomputation and vector addition chains, 389–399.
T. Eng, T. Okamoto, Single-term divisible electronic coins, 306–319.
M. Franklin, M. Yung, The blinding of weak signatures, 67–76.

Advances in Cryptology – **EUROCRYPT '95**, Saint-Malo, France. Springer-Verlag LNCS 921 (1995).
Editors: L.C. Guillou and J.-J. Quisquater

P. B´eguin, A. Cresti, General short computational secret sharing schemes, 194–208.
J. Bierbrauer, r s
-codes from universal hash classes, 311–318.

S. Brands, Restrictive blinding of secret-key certificates, 231–247.
L. Chen, T.P. Pedersen, On the efficiency of group signatures providing information-theoretic anonymity,
39–49.
C. Cr´epeau, L. Salvail, Quantum oblivious mutual identification, 133–146.
S. D'Amiano, G. Di Crescenzo, Anonymous NIZK proofs of knowledge with preprocessing, 413–416.
Y. Desmedt, Securing traceability of ciphertexts – Towards a secure software key escrow system, 147–157.
G. Di Crescenzo, Recycling random bits in composed perfect zero-knowledge, 367–381.
M.K. Franklin, M.K. Reiter, Verifiable signature sharing, 50–63.
C. Gehrmann, Secure multiround authentication protocols, 158–167.
R. Gennaro, S. Micali, Verifiable secret sharing as secure computation, 168–182.
J.D. Goli´c, Towards fast correlation attacks on irregularly clocked shift registers, 248–262.
C. Harpes, G.G. Kramer, J.L. Massey, Ageneralization of linear cryptanalysis and the applicability of Matsui's
piling-up lemma, 24–38.
W.-A. Jackson, K.M. Martin, C.M. O'Keefe, Efficient secret sharing without a mutually trusted authority,
183–193.

Journal of Cryptology papers

Journal of Cryptology papers (Volume 1 No.1 – Volume 9 No.3, 1988-1996)
M. Abadi, J. Feigenbaum, Secure circuit evaluation, 2 (1990), 1–12.
C. Adams, S. Tavares, The structured design of cryptographically good S-Boxes, 3 (1990), 27–41.
G.B. Agnew, T. Beth, R.C. Mullin, S.A. Vanstone, Arithmetic operations in
.7._:.#...
, 6 (1993), 3–13.
G.B. Agnew, R.C. Mullin, I.M. Onyszchuk, S.A. Vanstone, An implementation for a fast public-key cryptosystem,
3 (1991), 63–79.
P. Beauchemin, G. Brassard, A generalization of Hellman's extension to Shannon's approach to cryptography,
1 (1988), 129–131.
P. Beauchemin, G. Brassard, C. Cr´epeau, C. Goutier, C. Pomerance, The generation of random numbers
that are probably prime, 1 (1988), 53–64.
D. Beaver, Secure multiparty protocols and zero-knowledge proof systems tolerating a faulty minority, 4
(1991), 75–122.
M. Bellare, M. Yung, Certifying permutations: noninteractive zero-knowledge based on any trapdoor permutation,
9 (1996), 149–166.
I. Ben-Aroya, E. Biham, Differential cryptanalysis of Lucifer, 9 (1996), 21–34.

S. Bengio, G. Brassard, Y.G. Desmedt, C. Goutier, J.-J. Quisquater, Secure implementation of identification
systems, 4 (1991), 175–183.
C.H. Bennett, F. Bessette, G. Brassard, L. Salvail, J. Smolin, Experimental quantum cryptography, 5
(1992), 3–28.
E. Biham, New types of cryptanalytic attacks using related keys, 7 (1994), 229–246.
E. Biham, A. Shamir, Differential cryptanalysis of DES-like cryptosystems, 4 (1991), 3–72.
S. Blackburn, S. Murphy, J. Stern, The cryptanalysis of a public-key implementation of finite group mappings,
8 (1995), 157–166.
C. Blundo, A. De Santis, D.R. Stinson, U. Vaccaro, Graph decompositions and secret sharing schemes, 8
(1995), 39–64.
J. Boyar, Inferring sequences produced by a linear congruential generator missing low-order bits, 1 (1989),
177–184.
J. Boyar, K. Friedl, C. Lund, Practical zero-knowledge proofs: Giving hints and using deficiencies, 4
(1991), 185–206.
J. Boyar, C. Lund, R. Peralta, On the communication complexity of zero-knowledge proofs, 6 (1993), 65–
85.
J.F. Boyar, S.A. Kurtz, M.W. Krentel, A discrete logarithm implementation of perfect zero-knowledge
blobs, 2 (1990), 63–76.
E.F. Brickell, D.M. Davenport, On the classification of ideal secret sharing schemes, 4 (1991), 123–134.
E.F. Brickell, K.S. McCurley, An interactive identification scheme based on discrete logarithms and factoring,
5 (1992), 29–39.
E.F. Brickell, D.R. Stinson, Some improved bounds on the information rate of perfect secret sharing schemes,
5 (1992), 153–166.
J. Buchmann, H.C. Williams, A key-exchange system based on imaginary quadratic fields, 1 (1988), 107–
118.