Trabalho de Criptografia - INF 541

Curso de Especialização em Redes – Prof. Ricardo Dahab
Guilherme Steinberger Elias - RA 501022
Data de entrega : 14/06/2001

(i) Introdução com motivação do tema

Este trabalho terá como tema o uso de Protocolos/Produtos Específicos da Criptografia para a garantia de uma troca de informações segura entre o transmissor e o receptor através de redes públicas ou privadas. Atualmente as redes de computadores estão expostas por causa da Internet ou por trocarem informações remotamente via canais inseguros . Com isso a necessidade de se tornar os dados seguros e autênticos aumentou bastante. As mensagens e dados precisam ser protegidos de forma que somente pessoas ou processos autorizados consigam utilizá-los.A informação não pode ser alterada acidental ou propositalmente e nem destruída. Os métodos criptográficos por si só não são suficientes para garantir a segurança de um sistema. Deve-se recorrer aos protocolos criptográficos que nada mais são do que formas de se utilizar os algoritmos criptográficos. Para entendermos melhor como isso é feito, estaremos analisando os seguintes sub-temas :

- **Estabelecimento, Gerenciamento e Distribuição de Chaves**

As chaves secretas e públicas tem sido fundamentais na criptografia para o processo de ciframento e deciframento de textos claros. A idéia principal, segundo Kerckhoffs (um dos pais da criptografia), é que a tarefa de decriptação sem o auxílio da chave seja materialmente ou matematicamente impossível. Os principais desafios têm sido os seguintes:

1. Qual procedimento adotar para chegar a um acordo inicial sobre uma chave secreta comum ?
2. Como armazenar e gerenciar as chaves ?

Veremos mais detalhes nas próximas partes deste trabalho.

- **Autenticação**

A autenticação é mais uma parte essencial da criptografia.
Qual seria o processo que permite ao sistema verificar se a pessoa ou processo que está participando da comunicação é quem alega ser e de se garantir que a mensagem é atual ? Também veremos isso mais adiante.

- **Integridade**

Outro ponto importante: Como garantir que a informação é correta, original e não foi alterada acidentalmente ou intencionalmente ? O controle de integridade da informação é

um dos pré-requisitos para um criptossistema seguro que analisaremos mais a frente.

- **Identificação**

Antes de se autenticar, é necessário a existência de uma identificação que represente as partes envolvidas no processo de autenticação. Como tornar única uma entidade ou processo que participa de uma comunicação ?

- **Assinaturas Digitais**

Como autenticar as informações de forma a reproduzir no mundo digital o equivalente a uma assinatura ?
Estaremos examinando o funcionamento de uma das melhores formas de autenticação utilizadas: a assinatura digital.

(ii) Resumo do estado da arte

Hoje em dia com o poder da computação, os primeiros sistemas criptográficos que foram criados já podem ser facilmente quebrados, como por exemplo os monoalfabéticos, permuta, polialfabéticos (exemplo:Vigenère) e rotores.
Seguem abaixo as soluções mais atualizadas da criptografia para garantir a segurança das informações nos dias atuais:

1. Estabelecimento, Gerenciamento e Distribuição de Chaves

A chave deve ser de um tamanho tal que inviabilize uma busca exaustiva. O artigo Blaze et al, 1996 argumenta que para se conseguir segurança atualmente tem que se ter uma chave de aproximadamente 80 bits. Pensando-se nos próximos 20 anos, esta **chave terá que ter 128 ou mais bits.**
Como geralmente as chaves são geradas a partir de Geradores de Números Aleatórios, os mais modernos devem ter uma semente gerada por um processo verdadeiramente aleatório. Sugere-se o uso do **gerador de números aleatórios Blum-Blum-Shub** com um módulo de 2048 a 4096 bits e uma semente de pelo menos 128 bits.
Para obter o melhor desempenho e segurança , recomenda-se a **combinação de técnicas simétricas e assimétricas** , onde a chave secreta é estabelecida com a técnica Assimétrica e o seu ciframento é feito com a técnica Simétrica, utilizando-se um algoritmo bastante seguro. Estudaremos estas técnicas mais adiante.
Sugere-se como Algoritmo de ciframento da chave o **RC4** com uma chave de 128 bits para algoritmo de fluxo, ou o **IDEA** como algoritmo de bloco. Ambos já foram bastante estudados e testados e têm demonstrado um alto grau de confiabilidade.

Para o estabelecimento da chave secreta sugere-se o uso do **RSA** com um módulo de 2048 a 4096 bits. O RSA é um dos algoritmos mais estudados e confiáveis. O **AES** também é recomendado por ser teoricamente o mais difícil de se quebrar e por ser o mais recente. Ele utiliza chaves de até 256 bits, mas como ainda foi pouco testado, deve ser utilizado com um pouco de cautela. Para uma comunicação entre um Transmissor e um Receptor, recomenda-se que a distribuição e gerenciamento das chaves sejam feitos através do **uso de Entidades confiáveis Distribuidoras de Chaves Públicas** (KDC's) que utilizem para o armazenamento técnicas baseadas no sistema de **Hierarquia de chaves**, o que dificulta bastante os ataques de hackers via sniffing (escuta de pacotes) e spoofing (substituição) e facilita o gerenciamento das chaves. O uso de KDC's aumenta o grau de confiança na identidade das partes envolvidas, mas os póprios KDC's devem ter os mais modernos sistemas de segurança.

2. Autenticação

O mecanismo ideal no processo de autenticação da identidade das partes envolvidas tem sido o seguinte:

Ao iniciar uma comunicação com o receptor, o transmissor deve informar a uma Entidade Distribuidora de chaves (KDC ou Key Distribution Center) com quem deseja se comunicar. Esta entidade vai gerar uma chave de sessão Ks aleatória e de pouca duração e vai enviá-la de volta para o transmissor cifrada com a chave deste e junto com algumas informações cifradas com a chave pública do receptor. Estas informações incluem a identidade do transmissor. Em seguida o transmissor vai guardar a chave Ks e a enviará para o receptor junto com as informações cifradas recebidas desta entidade. O KDC deve conhecer as chaves públicas do transmissor e do receptor que farão a comunicação.

Ao receber as informações cifradas com a sua chave pública, o receptor terá a chave de sessão Ks, a identidade do transmissor e perceberá que a mensagem foi cifrada pelo KDC pois somente ele conhece a sua chave pública. O receptor enviará então um identificador da transação (nonce) para o transmissor, que responde com o mesmo identificador transformado por uma função f e cifrada com a chave de sessão Ks . Assim este receptor saberá que a mensagem recebida não é um replay e saberá que o transmissor é quem diz ser. O nonce deve ser difícil de se adivinhar (Exemplo: timestamp).

O processo de autenticação da própria mensagem trocada entre o transmissor e o receptor deve ser feito através de assinaturas digitais que utilizem funçôes fortes de hashing ,como o algoritmo SHA-1, combinadas com algoritmos fortes de ciframento como o RSA. Sugere-se o uso de sistemas de autenticação modernos como o Kerberos.

3. Integridade

Para mostrar a evidência da corrupção acidental ou proposital dos dados, a checagem de integridade deve ser suprida por uma entidade CA (Certificate Authority) para prover integridade entre chave pública e uma entidade.

Isso garante que somente as partes autorizadas sejam capazes de alterar variáveis do sistema ou a informação transmitida. Esta entidade deverá ser capaz de aplicar a checagem em um conjunto de mensagens ou em uma simples mensagem, ou mesmo em campos específicos de uma mensagem. Elas deverão utilizar Funções de Hashing combinadas com chaves secretas, gerando MAC's que permitirão uma checagem segura da integridade .
A verificação de integridade irá garantir que as mensagens sejam recebidas como foram enviadas, ou seja, sem duplicação, sem inserção, modificação, reordenação ou replays.
Ela deve executar também a destruição de dados quando necessário.
Os melhores sistemas de controle de integridade preocupam-se também com a prevenção e recuperação da perda de integridade.

4. Identificação

As melhores formas de se fazer uma identificação segura é através de Entidades Públicas Certificadoras (CA ou Certification Authority) .
A identidade deve ser composta por fatores diferenciais que individualizem o indivíduo ou processo envolvido na comunicação, como por exemplo as Assinaturas Digitais, o uso de Timestamp, endereço de Enlace, entre outros.

5. Assinaturas Digitais

Sugere-se que as assinaturas digitais utilizem o Algoritmo RSA com um módulo de 2048 a 4096 bits e que o SHA seja a função de hash que irá participar no processo de geração da assinatura.
Deve-se utilizar funções de hashing unidirecionais de forma a gerar assinaturas digitais compactas e matematicamente invioláveis.

(iii) Plano Geral do Trabalho

Este trabalho irá abranger os conceitos de Estabelecimento, Gerenciamento e Distribuição de Chaves, Autenticação, Integridade, Identificação e Assinaturas Digitais , mostrando assim que estes temas da criptografia são fundamentais para o uso seguro de transações na Internet , Aplicações de ambiente Distribuído ou em qualquer sistema onde deve-se garantir a segurança das informações e a identidade das entidades envolvidas.
Não serão feitas demonstrações matemáticas pois a intenção é a de enfatizar conceitos.

1. Estabelecimento, Gerenciamento e Distribuição de Chaves

1.1 Introdução:

1.1.1 A necessidade:

Hoje em dia com a Internet e Aplicações em ambiente distribuído, existe mais do que nunca a necessidade de uma comunicação segura entre um transmissor e um receptor fisicamente distantes. As chaves são a base para o funcionamento de algoritmos de ciframento e Deciframento. Elas garantem a proteção da informação e por isso devem ser estabelecidas, gerenciadas e distribuídas da forma mais segura possível.

1.1.2 Alguns fatores dificultadores no uso de chaves:

A tarefa de Estabelecimento, Gerenciamento e Distribuição de Chaves está entre os principais desafios da criptografia moderna pois são quase impossíveis de se realizar nos dias atuais. Isto se deve a **grande quantidade de usuários** envolvidos e ao **poder computacional atual dos hackers**, levando-se em consideração os requisitos para um criptossistema seguro.

1.1.3. Alguns conceitos necessários para um criptossistema ser seguro:

O sistema deve ser **incondicionalmente seguro**, ou seja, o fato de se conhecer o texto cifrado de uma chave não deve dar ao criptoanalista nenhuma informação adicional a respeito do texto original da chave.Existe a necessidade de que sejam geradas **chaves distintas para pares de usuários**, necessidade de **renovação frequente das chaves**, **estabelecimento das chaves em sigilo**, **aleatoriedade das chaves** e o seu **armazenamento em local seguro**. O **espaço de chaves deve ser grande o suficiente** para resistir a uma busca exaustiva. Existem **técnicas Simétricas e Assimétricas** para o estabelecimento e distribuição das chaves. Ambas são complementares pois enquanto as Assimétricas permitem maior simplicidade no estabelecimento da chave de ciframento secreta, as Simétricas permitem alta velocidade no ciframento. Desta forma não se perde nada em segurança, mas ganha-se muito em velocidade.

1.2 O Estabelecimento das chaves

1.2.1 Técnicas Assimétricas ou chave pública:

Os algoritmos das técnicas Assimétricas são mais robustos do que os das Técnicas Simétricas, portanto são mais indicados no estabelecimento da chave de ciframento secreta do que no próprio ciframento da chave. Eles geram chaves muito grandes, aumentando a segurança. É comum fazerem uso de geradores de números aleatórios e de geradores de

números primos.
Seu conceito surgiu em 1976 para atender a uma nova necessidade gerada pelo avanço da tecnologia, a de que 2 pessoas em lugares distantes possam trocar mensagens de forma segura sem terem que se encontrar fisicamente. Se baseiam no conceito de chave de deciframento distinta da de ciframento, e no uso de funções computacionalmente intratáveis por um invasor. A chave de ciframento é publicada ou disponibilizada aos usuários, sem que haja quebra na segurança da chave de deciframento. Assim, cada usuário possui uma chave para ciframento (pública) e outra para deciframento (secreta). O conjunto de chaves públicas, uma para cada usuário, pode ser colocada numa lista acessível a todos os membros da rede. Um usuário A que queira enviar uma mensagem para outro usuário B procede da seguinte maneira:
1. A consulta a lista de chaves públicas para descobrir a chave pública K_B (de ciframento) de B, e a utiliza para cifrar a mensagem m , obtendo m = $K`_B(K_B(m))$.
Desta forma, todos os usuários cifram mensagens destinadas a B utilizando a mesma chave pública K_B, mas somente B tem o segredo da chave secreta $K`_B$.
Um exemplo de algoritmo de chave pública:

RSA :

R. Rivest, A Shamir e L. Adleman ficaram conhecidos por terem proposto o sistema RSA para estabelecimento de chaves no Sistema Assimétrico em 1978. Ele é o algoritmo de chave pública mais utilizado embora trabalhe com baixas taxas de processamento, utilize chaves muito grandes e tenha sido pouco estudado em relação aos da técnica simétrica. Ele é considerado extremamente forte se utilizado de forma adequada, mas tem execução lenta. Sua segurança se baseia na intratabilidade da fatoração de produtos de 2 números primos. Um usuário B , para determinar o seu par (K_B,$K`_B$), procede da seguinte forma:

1. Escolhe ao acaso 2 primos grandes p e q, da ordem de 10^{100}
   O produto n=p*q da ordem de 10^{200} é computacionalmente facil .
2. calcula o número fi(n)=(p-1)*(q-1)
3. B escolhe ainda ao acaso um número c primo com fi(n) e determina d tal que c*d = 1 (módulo fi(n)).
4. B publica a chave (pública) K_B=(c,n).
5. Um usuário A , para enviar uma mensagem a B, quebra-a em blocos e codifica-a de forma que cada bloco seja um número m no intervalo 0<=m<n.
6. Para cifrar m, o usuário A determina $K_B(m)=m^c$ mod n.

Propriedade1: Dada a chave pública K_B , é impraticável calcular a chave privada $K`_B$ a partir de $K_B$ utilizando somente um texto cifrado com $K_B$. Idem para Ka e K`a .
Isso garante o sigilo da mensagem x dado o texto cifrado y = $C(K_B, x)$, exceto para B e A.

Propriedade 2: Dado o texto cifrado y = $C(K_B,x)$ é impraticável deduzir x a partir de y e K_B. O mesmo é verdade para um texto cifrado com Ka.

As 2 propriedaddes garantem que s = C^-1(K`a,x) é computável somente por um usuário de posse de K`a .

Outros algoritmos de chave pública :

- Knapsack:
  Também conhecido por Mochila, foi o primeiro sistema prático de chave pública, hoje em dia considerado fraco.
- El Gamal :
  Inicialmente criado para implementar assinaturas digitais, tem como destaque o fato de permitir que 2 dos seus parâmetros públicos sejam compartilhados por um grupo de usuários.

1.3 Ciframento das chaves:

1.3.1 Técnicas Simétricas ou Chave Secreta :

São ideais para fazer o ciframento pois permitem altas taxas de processamento, utilizando chaves pequenas sem comprometer a segurança (100 a 200 bits).
Utilizam a chave de deciframento igual a de ciframento , sendo que ambas devem ser mantidas em sigilo absoluto, sendo conhecidas somente pelo transmissor e pelo receptor.
Cada par de entidades deve pré-estabelecer sua chave secreta utilizando um canal seguro.
Esta chave deve pertencer a um espaço grande o suficiente para inibir a busca exaustiva.

1.3.1.1 Tipos de algoritmos simétricos :

<u>Algoritmos Bloco a Bloco</u>:

No algoritmo bloco a bloco a segurança depende do algoritmo e do segredo da chave e é mais lento do que o fluxo, pois é mais complicado. (Exemplo: DES, IDEA e RC5)

Exemplos:
IDEA : A fraqueza do protocolo IDEA está unicamente no passo onde A pega a chave pública de B, pois A não tem como verificar a autenticidade da chave.

Electronic Codebook Mode (ECB) :
Blocos iguais resultam em cifrados iguais, indenpendentes entre si. Desaconselhável se a chave vai ser utilizada repentinamente. É o mais simples e o mais usado, mas é o menos seguro. Bom para ciframento de poucos blocos.

Cipher Block Chaining (CBC) :
Blocos iguais resultam em cifrados diferentes, dependentes de todos os outros anteriores.
Com recuperação de erros. Necessita de uma valor inicial y_0 protegido mas não secreto.
Útil para o Ciframento de quantidades arbitrárias, é o mais utilizado.

Cipher Feedback Mode (CFB) :
Assemelha-se a uma cifra de fluxo, onde é possível calibrar quantos bits serão produzidos na cifra e também possui recuperação de erros.Útil em ciframento em fluxos.

Output Feedback Mode (OFB) :
Igual ao CFB mas com baixa propagação de erros. Útil em ciframentos em fluxos em presença de ruído.

Os blocos podem ser do tipo estanques ou encadeados.
Blocos Estanques permitem ciframento e deciframento não seqüencial mas a criptoanálise é mais fácil, pois há muitos pares (x,y) com a mesma chave.
Blocos encadeados são mais seguros mas propagam erros.

Algoritmos de Fluxo :

No algoritmo de fluxo a segurança depende do algoritmo que gera a seqüência de bits da chave e do segredo da semente do gerador. É bem mais rápido pois é um combinador de bits simples.

Exemplos:
One-Time Pad, conhecido por ser teoricamente inquebrável matematicamente. No entanto, utiliza uma chave do tamanho da mensagem, e que não pode ser reutilizada.

RC4, criado por Rivest, gera uma sequência pseudo-aleatória criptograficamente forte de forma mais rápida que o Blum-Shub.

Quantidade de chaves:

O uso de 1 chave aumenta a segurança se o sistema não exibir a característica onde para todo par de chaves k_1 , k_2, exista uma chave k_3 tal que $C(k_1,C(k_2,x)) = C(k_3,x)$.
Exemplo: DES e RSA.

Para múltiplas chaves, o espaço de chaves aumenta mas o sistema fica susceptível ao ataque “Meet-in-the-Middle” pois sendo $y = C(k_1,C(k_2,x))$, pode existir z tal que
$z = C(k_1,x) = C\text{-}1(k_2,y)$

1.4 Como armazenar e gerenciar as chaves ?

1.4.1 Técnicas Assimétricas ou chave pública:

São utilizadas chaves públicas e privadas, evitando a necessidade de pré-estabelecimento de chaves, mas aumentando a necessidade de um bom sistema de autenticação e certificação de chaves públicas para evitar ataques do tipo “man-in-the-middle” ou Intermediário. Este sistema não exige renovações freqüentes de chave e apenas um lado

mantém o segredo de uma determinada chave, tornando o gerenciamento mais simples. É um sistema eficiente para assinaturas. Neste caso o espaço de chaves é menor e portanto é possível fazer busca exaustiva no espaço de textos claros a partir de textos cifrados pois a chave é pública.

1.4.2 Técnicas Simétricas ou Chave Secreta :

O gerenciamento de suas chaves é mais difícil pois a chave deve ser mantida secreta no transmissor e no receptor, e isso pode exigir o envolvimento de terceiros com sistemas hierarquizados (Chaves que decifram outras chaves) e também a necessidade de que as chaves sejam renovadas com freqüência, causando um número de chaves muito grande para ser gerenciado. O ideal é que a cada nova sessão uma nova chave seja estabelecida, para aumentar a segurança do sistema. No entanto isto cria um problema: como estabelecer a chave ao início de cada sessão ? Deve-se utilizar pelo mesmo canal inseguro que transporta mensagens e portanto sujeito a escuta ? Enviá-la cifrada ? Com que chave ?
Este problema tem várias soluções, uma delas é o uso de uma função unidirecional exponencial módulo um número, isto é, dados os inteiros a, x e n e seja $f(x) = a^x \bmod n$ ($n>0$, $x>=0$). Assim, f(x) é o resto da divisão de a^x por n. O cálculo desta função é viável mas determinar os dados a, n e f(x) é computacionalmente inviável. Portanto esta função é do tipo unidirecional.

1.4.3 Hierarquia de chaves:

As chaves devem então ser armazenadas de forma segura, por exemplo através de uma Hierarquia. Isso reduz o número de chaves manualmente gerenciadas e distribuídas.
Em grandes redes, não compensa a existência de funções que evitam replay.
Como alternativa pode-se fazer uma hierarquia de KDC's , cada um em uma área (pequeno domínio) da rede, o que traz mais segurança. As chaves mestras cifram chaves de sessão que cifram chaves para ciframento de dados. Quanto maior a quantidade e freqüência da troca de chaves, menor a sua importância.
Chaves de sessão são para algoritmo simétrico e serão utilizadas para uma sessão de comunicação e depois descartadas. Elas aumentam a segurança na transmissão de chaves, pois se o inimigo descobrir uma das chaves, ele só conseguirá ler as mensagens trocadas durante a seção em que a chave que ele descobriu foi usada.
Para que a segurança não seja comprometida, as chaves devem ser no mínimo tão protegidas quanto a mensagem, mesmo que o algoritmo de ciframento seja inquebrável. Imagine um sistema criptográfico inviolável, onde está armazenada uma chave mestra CM_0. Suponha que o administrador gera a chave CM_0 bit a bit utilizando alguma forma aleatória e a armazena no dispositivo criptográfico através do acionamento de uma chave física e a execução de uma instrução especial. Por segurança, uma cópia de CM_0 é armazenada num lugar seguro, como por exemplo um cofre da instituição. A chave mestra é utilizada para cifrar chaves voláteis ou primárias, que são utilizadas uma única vez e por pouco tempo, como por exemplo chaves de sessão. A idéia básica é que estas chaves jamais ficarão expostas no sistema, a não ser internamente no dispositivo criptográfico. O

dispositivo criptográfico tem as suas funções de ciframento e deciframento sob controle do sistema operacional, para cifrar m com uma chave de sessão s, onde a função de ciframento utiliza a chave mestra e m para gerar s(m). A segurança do sistema só será comprometida se o oponente conseguir violar o dispositivo criptográfico ou controlar o sistema operacional acionando a função de deciframento baseado na chave mestra e em s(m). Em uma rede, cada computador deve possuir a sua própria chave mestra distinta das dos demais, o que reduz os pontos nos quais uma chave de sessão cifrada pode ser atacada.
Se é verdade que o sigilo das chaves é protegido pelo ciframento das mesmas, também é verdade que o uso das chaves cifradas também precisa ser controlado de forma que os usuários só tenham acesso ao uso de chaves para as quais forem autorizados. A criptografia não oferece sozinha uma solução para este problema, e precisa ser utilizada em conjunto com características de segurança providas pela arquitetura do computador e do sistema operacional. Além disso, trocas freqüentes das chaves são indicadas para limitar a quantidade de dados que possam estar comprometidos se um atacante descobrir a chave.
O controle descentralizado de chaves é uma forma de se evitar o uso e confiança em um só KDC. Isso não é prático em redes grandes, mas vale a pena para redes locais.
O inimigo pode atacar o protocolo com sucesso somente se o transmissor e o receptor não conhecerem com antecedência as chaves públicas um do outro.
Uma alternativa é se o transmissor e o receptor conhecerem uma pessoa ou entidade confiável em comum. Então ambos confiariam numa assinatura desta entidade considerando que ambos já possuem a chave pública desta entidade a qual confiam ser verdadeira. O único ataque possível é na própria entidade, por exemplo adulterando-se seus bancos de dados. Quanto maior a freqüência de troca das chaves de sessão, mais seguras elas serão, pois o inimigo terá menos texto cifrado para trabalhar. Por outro lado, isso aumenta o tempo até se iniciar a troca de informações e ocupa mais a banda de rede.
Em protocolos baseados em conexão, o ideal é utilizar uma chave de sessão durante o período em que a conexão estiver aberta. Caso ela dure muito tempo, a chave de sessão deve ser renovada após um certo período pré-definido.
Protocolos não baseados em Conexão devem ter uma nova chave a cada troca de informações, mas isso irá aumentar o overhead e o tempo por conexão, o que vai contra os princípios deste tipo de protocolo.

1.4.4 Divisão e compartilhamento de segredos:

Suponha um segredo encriptado com uma chave k. Preocupa-se com a possibilidade de a chave ser perdida e não se conseguir mais recuperar a informação.
O objetivo básico da criptografia é tornar impossível o acesso ao segredo sem a chave. Pode-se então dividir o segredo com outras pessoas mas caso não se confie em nenhuma, uma solução seria garantir que somente juntas as pessoas tenham acesso ao segredo. Deve-se garantir que nenhuma delas poderá separadamente obter informações sobre o segredo.

1.5 Como distribuir as chaves ?

Para que a encriptação tenha bons resultados, o receptor e o transmissor devem compartilhar a mesma chave e esta deve estar protegida do acesso de outros.
A força do criptossistema reside em sua técnica de distribuição de chaves, ou seja, na forma como a chave será entregue entre duas entidades que desejam trocar informações sem que os outros conheçam a chave.

1.5.1 Sistema Simétrico:

Neste sistema há 2 formas básicas de se estabelecer uma chave comum entre 2 usuários A e B :
1. Estática:
A chave é determinada a priori por A e B através de um meio seguro. Eles podem inclusive combinar diversas chaves k, uma para cada ocasião em que forem trocar mensagens.
As chaves devem ser mantidas em local seguro.

2. Dinâmica:
As chaves são geradas ã medida que é necessário, de forma independente ou através de um KDC (Key Distribution Center).

1.5.2 Sistema Assimétrico:

Um usuário A querendo se comunicar com B, pode obter a chave pública de B de 3 formas:
1. a partir de um arquivo particular
2. a partir de um KDC
3. através de uma linha insegura

O processo de distribuição pode ser de várias formas. Uma das formas mais seguras é através da dedução local da chave secreta entre o transmissor A e o receptor B após a troca de algumas informações no canal público e inseguro.
Diffie e Hellman ficaram famosos por proporem um método de distribuição pública de chaves secretas baseado na escolha de números primos grandes com mais de 200 bits.

Existem diversas outras formas de se combinar a chave :
- O transmissor pode escolher a chave e entregá-la fisicamente ao receptor
- Um terceiro pode selecionar a chave e entregá-la fisicamente ao transmissor e ao receptor.
- Se o transmissor e o receptor utilizaram uma chave recentemente, um deles pode transmitir uma nova chave ao outro cifrando-a com a chave antiga.
- Se o transmissor e o receptor tiverem uma conexão cifrada com um terceiro, este pode entregar a chave nos links cifrados do transmissor e receptor.

A opção de entrega física é a mais segura mas inviável em um cenário onde existem ambientes distribuídos distantes fisicamente e compartilhando a mesma rede. Cada par de hosts deverá possuir uma chave de comunicação. Para N hosts serão necessárias [N(N-1)]/2 chaves.

Exemplo de um bom protocolo para distribuição das chaves :

1. A pega a chave pública de B.
2. A escolhe uma chave aleatória qualquer para um algoritmo de chave simétrica.
3. A usa a chave pública de B para encriptar a chave escolhida, e manda para B o resultado.
4. B usa a sua chave particular para desencriptar a chave enviada por A .
5. A e B usam um algoritmo simétrico com a chave para trocar as mensagens.

Este protocolo é o mais comum para uso de chave pública na troca de mensagens e evita problemas de lentidão dos algoritmos de chave pública.

Exemplo de distribuição de chaves através de um KDC (Key Distribution Center):

A compartilha uma chave Ka com o KDC e B compartilha uma chave Kb com KDC.

1. A pede ao KDC uma chave de sessão para proteger uma conexão com B.
O pedido menciona a identidade de A e B e um identificador único para a transação, chamado de nonce, que pode ser um timestamp, contador, número aleatório, sempre diferindo para cada requisição e difícil de ser adivinhado.
2. O KDC envia uma mensagem cifrada com Ka , chave de sessão Ks e nonce para A, que vai ser o único capaz de decifrá-la e saberá que ela foi enviada pelo KDC.

A mensagem também contém Ks para B e identificador de A para B (ex. endereço de rede) cifrados com Kb. Esta mensagem é enviada de A para B, provando que ele é confiável.
3. Agora B sabe Ks, sabe que o transmissor é A e sabe que foi cifrada com a sua chave pública que só pode ser obtida no KDC.
4. B envia um novo nonce para A .
5. A responde a B com o nonce transformado por uma função f e cifrado com Ks . Assim B saberá que a mensagem recebida não é um replay.

2. Autenticação

Uma autenticação ocorre quando uma entidade precisa provar para outra a sua identidade. A maior parte das pessoas que trabalham com computadores faz justamente isso quando entram com uma senha para ter acesso a algum recurso.
Chama-se a quem deve provar a identidade de usuário (U) e a quem exige a autenticação de servidor (S). O servidor deve sempre armazenar alguma informação a respeito do usuário

do sistema.
Um exemplo ruim de protocolo é o uso de senhas , onde U escolhe uma senha e a comunica a S. O servidor S guarda a senha em seu banco de dados. Quando U precisa ser autenticado, S requisita a senha a U. Deposi que U fornece a senha, S verifica se ela é igual à armazenada, para aceitar a autenticação de U. É um método ruim pois se o inimigo conseguir acesso ao banco de dados do servidor, todas as senhas estarão comprometidas. Para se resolver este problema utiliza-se uma função hash para guardar as senhas. Mas isso não evita o ataque do dicionário, que baseia-se no fato de que se por um lado uma função de hash não pode ser invertida, por outro lado nada impede que se tente experimentar valores e verificar o sumário que estes produzem. Este ataque pode ser prático se utilizar um dicionário com mais de 100000 termos. Para evitar este problema, deve-se utilizar um padrão de senhas que sejam parecidas com sequências aleatórias.

2.1 Exemplo de protocolo de autenticação :

Protocolo de Schnorr:
Ele permite que diversos usuários compartilhem vários parâmetros.
É considerado muito seguro para autenticações.

Funcionamento:
1. Obtém-se um número primo q aleatório.
2. Obtém-se um número primo p aleatório tal que q seja um dos fatores de (p-1) .
3. Encontra-se um número a diferente de 1, tal que a^q ,mod p = 1 .

Estes números p, q e a são públicos. O servidor escolhe também um parâmetro k de segurança.

Um usuário do sistema escolhe uma chave da seguinte forma:
4. Escolhe uma chave secreta com um número aleatório s (s<q)
5. Obtém t a partir de a^s mod p
6. Obtém a chave pública v a partir de t^-1 mod p
7. O usuário U comunica a sua chave pública ao servidor S.

Quando uma autenticação for necessária, utiliza-se o seguinte protocolo :

8. U escolhe um número aleatório r (r < q) .
9. U obtém x a partir de a ^y mod p , e manda x para S .
10. S escolhe um número aleatório ”e” entre 0 e (2^k – 1) e o envia a U .
11. U calcula y a partir de (r+s*e) mod q, e envia y para S .
12. S verifica se x=(a^v * v^e) mod p para aceitar a autenticação de U.

2.2 Autenticação do remetente de mensagem:
No caso da chave secreta, existem basicamente 2 métodos para a autenticação de

remetente. No primeiro, a autenticação é feita incorporando à mensagem a identidade do remetente, antes do ciframento. No segundo caso, as chaves secretas são unidirecionais, ou seja, existe uma chave secreta c1 para que A cifre mensagens para B e outra c2 para que B cifre mensagens para A . Neste caso, somente B usa c2 para cifrar mensagens, somente A e B conhecem c2. Ao decifrar mensagens que trazem como remetente B, o usuário A decifra-as e conclui que de fato B é o remetente, se as mensagens originais forem devidamente recuperadas. Se c1 fosse igual a c2, alguma mensagem enviada de A para B, cifrada com c1, poderia mais tarde ser enviada novamente a A, como se viesse de B.
Em ambos os casos o sigilo é essencial para a verificaçãode autenticidade do remetente. No caso de não se desejar sigilo, senhas podem ser incorporadas ãs mensagens, nesse caso as senhas devem ser cifradas, para evitar que o oponente as determine.Isto contudo não é o suficiente , pois o oponente pode copiar o valor cifrado de uma senha e adicioná-lo a uma mensagem falsa. Isto é resolvido mediante a autenticação de conteúdo sem sigilo.

2.3 Autenticação do conteúdo de mensagens:

Dada uma mensagem m, o remetente calcula uma quantia a(m), chamada de autenticador de m, cuja propriedade é a de refletir qualquer alteração na mensagem m com alta probabilidade. Em seguida ele envia m seguido de a(m) para o destinatário.
Ao recebê-la, o destinatário precisará recalcular o autenticador e verificar se o resultado é igual a a(m). Outra propriedade importante é a de que deve ser muito difícil calcular a(m`) dada uma mensagem arbitrária m` diferente de m.
Se m for transmitida cifrada, a(m) pode ser gerado como um subproduto do ciframento.
Se ao invés de cifrarmos m, duplicarmos o primeiro bloco de m no fim de m antes do ciframento, este último bloco , após o ciframento, será o autenticador a(m).
No deciframento, o primeiro bloco deverá ser igual ao último. Como há propagação de erro, qualquer alteração será refletida no último bloco.

2.4 Autenticação da Atualidade de mensagens:

Concatenando-se antes do ciframento a cada mensagem uma quantia t, dependente do tempo, de conhecimento do remetente e destinatário, teremos como verificar se a mensagem é ou não atual. O remetente A calcula E(k,(m,t)) e envia o resultado para o destinatário B, que irá decifrá-lo obtendo t, que será comparado com o valor esperado. Para isso deve haver um sincronismo entre as 2 partes.
Outra alternativa é a de A enviar para B uma quantia r, gerada aleatoriamente. O envio seria O resultado de E(k,(m,r)).
Em seguida, B executa uma função pública em r , como por exemplo r+1, e devolve E(k,(m`,r+1)).
Desta forma, A poderá verificar que m` é atual e não a repetição de uma mensagem antiga cifrada com a mesma chave k. O remetente A poderá ainda continuar a comunicação enviando para B o resultado de E(k,(m``,r+2)) . Em seguida B também passará a se certificar da atualidade das mensagens enviadas por A .

Outra forma de autenticação é o uso de assinaturas digitais, que será vista mais adiante.

3. Integridade

Grande parte das vulnerabilidades encontradas nos modos de operação dos algoritmos de chave secreta eram devidas à falta de um mecanismo que garantisse a integridade das mensagens enviadas. Funções de hashing combinadas com uma chave secreta podem ser utilizadas para produzir MAC's (Message Autentication Code) e consequentemente verificar a integridade destas mensagens. O resultado é um valor que só pode ser computado pelas partes que conheçam a chave secreta utilizada.
Existem sistemas onde os serviços que gerenciam a integridade estão voltados apenas para a detecção e então é necessária a intervenção humana para checar a violação.

4. Identificação

A criação da identificação pode ser baseada em diversos fatores, como por exemplo uma associação do Endereço de rede com o Timestamp e a chave secreta de uma entidade.
Existem diversos mecanismos para se verificar e validar a identidade dos usuários envolvidos em uma transação/comunicação.
Como podem o destinatário e o remetente, certificarem-se da identidade um do outro ?
A autenticação da identidade das partes numa conversação é diferente da autenticação de usuários no sistema operacional. Não se pode utilizar senhas porque elas teriam que ser combinadas aos pares para cada par de usuários, criando problemas sérios de segredo e atualização das senhas. No entanto, existe uma similaridade se ambas as partes utilizarem um sistema simétrico para trocar mensagens:
Como a chave secreta k` é conhecida apenas pelas 2 partes, o deciframento correto das mensagens recebidas garante que as partes são autenticas. Se o sistema for assimétrico, não existe esta garantia pois uma das chaves é pública.

5. Assinaturas Digitais

5.1 Definição :
É um tipo de autenticação que seria o análogo computacional de um documento tradicional assinado por uma pessoa, onde o destinatário pode provar a identidade do remetente e a integridade da mensagem recebida inclusive para terceiros. E*quivalem a registrar em cartório o conteúdo de um dado documento. Elas dependem da informação e de quem assina e podem ser verificadas por terceiros.*
São feitas para que uma entidade possa digitalmente assinar um documento, esperando-se que esta assinatura eletrônica tenha as mesmas características de uma assinatura do mundo real, ou seja, fácil de produzir para quem assina, fácil de se verificar por qualquer um, muito difícil de ser falsificada, tenha uma vida útil apropriada de modo que quem assine não possa negar ter assinado.

5.2 Técnicas:

Como autenticar as informações de forma a reproduzir no mundo digital o equivalente a uma assinatura ?

Existem algumas técnicas para combinar sigilo com assinatura, reconhecer uma assinatura sem a presença da mensagem original ou gerar assinaturas pequenas.

A vantagem de se utilizar sistemas assimétricos é que nestes não é necessário a figura de terceiros e pode-se preservar a privacidade do conteúdo dos documentos assinados.

Há situações em que o algoritmo de chave pública é utilizado tanto para cifrar quanto para assinar um documento, por exemplo no RSA. Algoritmos de chave pública como RSA, El Gamal e DSS podem ser utilizados para calcular a assinatura digital de uma determinada mensagem. Recomenda-se o uso de técnicas baseadas nos algoritmos RSA e DSA.

As assinaturas derivam de Funções Unidirecionais, que são primitivas criptográficas para a construção de protocolos criptográficos específicos onde é fácil calcular y = f(x) mas muito difícil calcular x = f-1(y)

O RSA é considerado o melhor esquema para assinaturas. Ele pode ser utilizado com qualquer tamanho de módulo, o que seu concorrente DSA não permite. O RSA, ao contrário do DSA, pode ser utilizado tanto para assinar mensagens quanto para encriptá-las e pode ser utilizado para fazer assinaturas cegas que serão vistas mais adiante. Além disso ele é simples. O processo de geração de uma assinatura digital é normalmente implementado com o auxílio de uma função de hashing. Ao invés de cifrar todo o conteúdo da mensagem, cifra-se somente a saída da função de hash. Isso aumenta a velocidade de geração e verificação de assinaturas já que as funções de hashing são mais rápidas e que os algoritmos assimétricos atuais, e a assinatura pode ser gerada separadamente da mensagem propriamente dita, ou seja, a asinatura é independente do fato da mensagem original ser mantida em sigilo ou não. Para ser útil na criptografia, a função de hashing deve ser unidirecional e resistente a colisões. Uma das mais importantes aplicações de assinaturas digitais é em certificação de chaves públicas.

5.3 Estrutura:

Um esquema de assinatura é composto de :

- Geração de assinatura
- Verificação de assinatura
  ->Com apêndice : A mensagem, informação assinada, é necessária para a verificação.
  ->Com recuperação de mensagem: A mensagem não é necessária para a verificação.

Nunca se deve assinar uma mensagem cuja origem é desconhecida. Deve-se sempre verificar a procedência do que se assina, com exceção das assinaturas cegas que serão vistas mais adiante. Quando o algoritmo exigir um k aleatório, este nunca deve ser reutilizado ou divulgado. Normalmente utilizam esquemas parecidos com os algoritmos de chave pública, de forma que a chave privada é algo que quem assina utiliza, e a chave pública é o que se deve utilizar para verificar a autenticidade da assinatura.

É importante mencionar que uma assinatura não é feita para proteger uma mensagem contra um inimigo e sim para garantir que uma determinada pessoa realmente tenha

assinado a mensagem, de forma que todos que quiserem verificar a assinatura terão que ter acesso à assinatura e à mensagem, incluindo o inimigo. A idéia é evitar a falsificação da assinatura.

5.4 Uso do RSA nas assinaturas:

Além de ser utilizado para ciframento,o RSA também serve para fazer assinaturas, gerando chaves privadas e públicas. Ao se assinar uma mensagem m , a assinatura s será
S = m^d mod n
Para se verificar a asinatura utiliza-se m`=s^l mod n .
Se m = m` , então a assinatura s deve ser considerada válida. Note que para um inimigo forjar uma assinatura ele precisa saber o valor de d, o que seria equivalente a quebrar o RSA (o que seria tão difícil quanto fatorar n)

Exemplo: Deseja-se assinar o valor m=745:
S = m^d mod n = 745 ^405 mod 1003 = 539
Para verificar a assinatura, m`= s^l mod n = 539^637 mod 1003 = 745.
Como m = m`, então a assinatura s=539 é válida para a mensagem m = 745.

- A assinatura pode ser acompanhada de ciframento. Módulos e chaves são diferentes.
- Há várias funções de redundância propostas.
- O esquema apresentado pode ser transformado em “com apêndice” facilmente.
- Propõe-se um expoente de verificação igual a 3 ou 2^16 + 1 ou similares.

Exemplo de assinatura digital em sistemas simétricos comutativos:
No RSA, SB(PB(m))=m^ci mod n = PB(SB(m))
Ou seja, PB e SB comutam e é possível decifrar uma mensagem e depois cifrar o resultado obtendo a mensagem original. Sistemas em que o ciframento e o deciframento comutam se dizem comutativos e admitem uma assinatura digital natural.
Suponha que A deseja enviar uma mensagem m assinada porém sem sigilo. Basta B enviar a A o texto SB(m). Somente B é capaz de calcular SB(m) a partir de m. Qualquer usuário é capaz de recuperar m = PB(SB(m)) e certificar-se que de fato B é o remetente da mensagem m. Existe assim um compromisso entre sigilo e autenticação.
Qualquer pessoa pode verificar que B é o autor de m cifrando SB(m) com PB, pois somente B pode determinar SB(m) a partir de m, e PB é pública. A mensagem m poderia ser qualquer coisa. Para garantir sigilo, B pode enviar a A não SB(m) mas sim PA(SB(m)). O usuário A decifra PA(SB(m)) aplicando inicialmente AS e em seguida PB, recuperando m. Para provar perante um tribunal virtual que B lhe enviou m, A apresenta SB(m) e pede ao juiz para cifrá-lo com PB, repetindo o último estágio do processo, que pode ser rotulado como reconhecimento de firma.

5.5 Uso do El Gamal nas assinaturas :
Deseja-se assinar m. Em primeiro lugar, deve-se escolher um número aleatório k , 1 <= k <= (p – 2) tal que mdc (k, p-1)=1. Para assinar faz-se a =g^k mod p
Deve-se então achara inversa multipicativa de BETA de k módulo (p – 1), ou seja
Achar BETA tal que (BETA – k) mod (p-1) = 1
Para encontrar b faz-se b=[BETA(m-x*a)] mod (p – 1)
A assinatura s é o par [a,b]. O valor de k deve ser mantido em segredo e nunca deve ser reaproveitado para assinar outras mensagens. Para verificar a assinatura, testa-se a condição:
(y^a 8 a^b) mod p = g^m mod p

Exemplo:
Quer-se assinar a mensagem m =34.984. Escolhe-se k = 60.859 e
a=21595 ^60859 mod 98627 = 35703
A inversa de 60859 módulo 98626 é BETA=47737.
Então b=[47737(34984 – 4640*35703)] mod 98626 = 41366

A assinatura será s=[35703, 41366].
Para se verificar:
(22448^35703 * 35703^41366) mod 98627 = 21595^34984 mod 98627
⇨ 61284 = 61284

5.6 Uso do DSA em assinaturas digitais :
Em 1991, o governo dos USA (NIST) publicou o DSA (Digital Signature Algoritm) para ser utilizado como um padrão federal (DSS – Digital Signature Standard).
Houve muito protesto pois o RSA é considerado melhor, muitas pessoas queriam que para o padrão DSS fosse adotado o RSA.
O DSA foi projetado pelo governo e não se pode saber se houve segundas intenções no projeto. Ele possui uma chave muito pequena e que não pode ser aumentada. Utiliza a função hash SHA.
É baseado na dificuldade de se resolver logaritmos discretos em Zp*. Pode-se utilizar outros grupos cíclicos.
Também é baseado na dificuldadde de resolver-se logaritmos discretos módulo p e módulo q. Parâmetros recomendados: q=160 bits ; p múltiplo de 64 entre 512 e 1024.

Definição :
1. Obtém 2 números primos aleatórios p e q que devem ter características especiais.
2. Dado um valor l (0 <= l <=8) os números p e q são tais que 2^159 < q < 2^160, ou sweja, q tem 160 bits.
3. Seja L=512+64l, deseja-se que 2^L-1 < p < 2^L , ou seja, p tem L bits.
4. Q é divisor de p-1
5. Deve-se agora selecionar um número aleatório x tal que x<q e um número aleatório h tal que h < (p-1). Assim faz-se , g=h^(p-1)/q mod p e y=g^x mod p

Existe a restrição de que g seja maior que 1, ou seja, se g=1 deve-se escolher outro valor para h.

6. A chave pública é [p,q,g,y] e a chave particular é x.

Como El Gamal, o DSA tem parâmetros que poderiam ser comuns a um conjunto de usuários. Estes são [p,q,g]. Cada usuário poderia então escolher um x e calcular seu y. Não se fornece exemplos para este algoritmo pois ele exige que se tenha números grandes demais para uma demonstração didática.

Funcionamento:

1. Deseja-se assinar uma mensagem m. Em primeiro lugar calcula-se SHA(m). O padrão diz que deve-se sempre assinar SHA(m) e não m em si.
2. Escolhe-se então um número aleatório k, tal que 0<k<q.
3. Deve-se calcular a inversa BETA de k módulo q. Isto significa que BETA obedece à relação:

(BETA*k) mod q = 1

então r = (g^k mod p) mod q

s=[BETA(SHA(m)+x*r)]mod q

4. A assinatura será o par [r,s].
5. Para verificar a assinatura deve-se calcular a inversa GAMA de s módulo q, ou seja, GAMA obedece ã relação:

(GAMA*s) mod q = 1

Assim:

u1 = [SHA(m)*GAMA] mod q

u2 = (r*GAMA) mod q

6. v = [(g^u1 * y^u2) mod p] mod q

   Se v= r então a assinatura é autêntica.

Percebe-se que o valor de r é independente da mensagem a ser asinada m, ou seja, pode-se gerar diversos valores para k, BETA e r antes de se precisar utilizá-los.
Isto torna o processo de assinatura muito mais rápido.
Lembrando que jamais deve-se utilizar um mesmo k para assinar 2 mensagens diferentes ou divulgar k usado para assinar uma determinada mensagem.

5.7 Assinaturas cegas:

Pode-se querer que alguém assine uma mensagem m sem que esta pessoa veja o que está assinando. Por outro lado, há motivos para que alguém queira assinar uma mensagem sem olhar. Seja uma entidade A interessada em gerar uma chave para o RSA, sabe-se que o RSA necessita de 2 números primos. Suponha ainda que A só tenha autorização para utilizar um RSA cujo módulo seja composto de números primos autenticados (assinados) por outra entidade B. Então A quer que B ateste que os números primos são válidos , e ao mesmo tempo, A não que que B saiba quais são os números primos escolhidos. B que garantir que o que assina é válido mas não pode ver o número que está assinando pois isso

comprometeria a segurança para A .

(iv) Bibliografia Consultada

- Segurança de dados com criptografia – Métodos e Algoritmos.
Autor: Daniel Balparda de Carvalho

- Introdução a Criptografia Computacional
Autor: Cláudio Leonardo Lucchesi

- Apostilas I e II de Criptografia INF 541

- Tese : Alguns Aspectos da Criptografia Computacional
Autor: Ricardo Dahab

- Tese : Segurança na arquitetura TCP/IP: de Firewalls a canais seguros
Autor: Keesje Duarte Pouw

- Criptography and Network Security – Principals and Practice
Autor: William Stallings