Universidade Federal de Pernambuco
Centro de Informática
Graduação em Ciência da Computação

Trabalho de Graduação:
Esquema de Assinaturas Digitais Tolerante a Falhas Utilizando Criptografia de Limiar

Aluno: Igor Medeiros Vanderlei
Orientador: Ruy José Guerra B. de Queiroz

Abril de 2004

R e s u m o

A idéia por trás da Criptografia de Limiar é proteger a informação através de um sistema distribuído de computadores tolerante à falha, onde a tarefa de criptografia não é realizada por apenas um único servidor, mas por um grupo de n servidores, de tal forma que se um adversário puder controlar até $t \quad 1$ servidores ainda assim não poderá descobrir o segredo nem impedir o correto funcionamento do sistema.

Este trabalho de graduação apresenta um estudo dos fundamentos matemáticos necessários para o desenvolvimento da criptografia de limiar e em seguida propõe um esquema de assinaturas digitais que utiliza a criptografia de limiar.

Embora os fundamentos matemáticos sejam abordados nesse trabalho, é necessário que o leitor tenha um conhecimento mínimo da teoria dos números em especial da álgebra modular.

Palavras-chave: Criptografia de limiar, compartilhamento de segredo, assinaturas digitais.

A g r a d e c i m e n t o s

Em primeiro lugar eu gostaria de agradecer a minha família por todo apoio, compreensão e felicidade que me ofereceram durante toda a minha vida.

Agradeço também ao meu orientador Ruy Guerra por toda ajuda prestada durante a realização desse trabalho, e por ter, mesmo que inconscientemente, me apresentado o universo da criptografia.

E por último, mas não menos importante, eu agradeço a todos os colegas com quem tive convívio durante a minha vida acadêmica.

S u m á r i o

Lista de Figuras

1. Introdução

O uso da internet para aplicações comerciais e financeiras tem se tornado cada vez maior entre pessoas e empresas que buscam maior agilidade nas transações. Neste ambiente, as assinaturas digitais vêm assumindo uma importância cada vez maior. Isto se deve ao fato de que estas criam a presunção de que quem assinou o documento eletronicamente é realmente o proprietário da assinatura, similarmente ao que ocorre com as assinaturas por escrito. Além disso, as assinaturas digitais asseguram a integridade e sigilo da informação e a sua não repudiação.

A maioria dos esquemas de assinaturas digitais envolve, além do par **A** e **B** que deseja se comunicar, uma terceira entidade altamente confiável, normalmente denominada de **Autoridade Certificadora**.

A autoridade certificadora tem a responsabilidade de garantir que uma dada assinatura realmente pertence a um usuário, impedindo assim, que alguém mal intencionado possa se fazer passar por outra pessoa.

Desta forma, caso a autoridade certificadora seja um único servidor e se este puder ser controlado por um intruso ou mesmo se por algum motivo ele sair do ar, todo o serviço de certificação fica comprometido. Por isso, é uma boa prática distribuir a tarefa da entidade certificadora entre vários servidores de modo que o adversário precise invadir e controlar vários servidores para poder forjar uma assinatura.

Uma possível forma de dividir a tarefa da autoridade certificadora é através da criptografia de limiar, que tem por princípio proteger a informação através de um sistema distribuído de computadores. Seu problema fundamental é o compartilhamento do segredo entre vários servidores utilizando um esquema que satisfaça os seguintes requisitos:

- Nenhum grupo de computadores, menor que o limiar definido, pode descobrir o segredo compartilhado, mesmo que eles cooperem entre si;

- Sempre que for necessária a recuperação do segredo, um número de servidores, igual ou maior que o limiar dado, pode fazê-la.

Este trabalho está organizado da seguinte forma. No Capítulo 3 são apresentados os fundamentos de um esquema de Assinaturas Digitais. No Capítulo 4 são explicados os conceitos de compartilhamento de segredo e criptografia de limiar. O Capítulo 5 descreve as considerações necessárias para a modelagem de um esquema de criptografia distribuída que será realizada no Capítulo 6. No Capítulo 7 são apresentadas as conclusões, no Capítulo 8 as sugestões para aprimoramento deste trabalho. O Apêndice 1 explica alguns conceitos e algoritmos básicos para quem não está familiarizado com a criptografia.

2. Contexto

2.1. TSS (Threshold Signature Scheme)

Esquemas de assinaturas de limiar provêm uma maneira de um grupo de n participantes, conjuntamente, gerar assinaturas digitais.

Diferente do que ocorre nos esquemas de assinaturas regulares onde a chave privada é mantida por uma única entidade, nos esquemas de assinaturas de limiar a chave privada é compartilhada entre as n partes de forma que uma assinatura válida só gerada com a participação de pelo menos $t \leq n$ participantes.

Para gerar uma assinatura, cada participante produz individualmente sua assinatura parcial da mensagem, em seguida as assinaturas parciais são combinadas para produzir a assinatura completa.

A assinatura resultante do esquema de assinaturas de limiar seria a mesma se tivesse sido produzida por uma única entidade de posse da chave privada completa. Além disso, a validade da assinatura pode ser verificada por qualquer entidade que tenha a única chave pública correspondente. Em outras palavras, o fato da assinatura ter sido produzida em um ambiente distribuído é transparente ao receptor da mensagem.

Esquemas de assinaturas de limiar são motivados tanto pela necessidade de algumas organizações precisarem que um grupo de empregados concorde com uma mensagem para poder assiná-la quanto pela necessidade de proteger as chaves de assinaturas tanto de ataques internos quanto de ataques externos. Um adversário precisaria controlar vários participantes a fim de fraudar o sistema uma vez que a chave privada nunca é reconstruída durante a execução do protocolo.

Existem vários estudos sobre a aplicação das assinaturas de limiar baseados em esquemas de assinatura digitais tradicionais. Estudos baseados no RSA podem ser encontrados em [13] e [14], já o [15] se baseia no DSS.

2.2. PGP

O PGP traz uma abordagem de “teia de confiança” para o problema de autenticação de chaves públicas. Sua estrutura é descentralizada e se caracteriza pela inexistência de uma Autoridade Certificadora, em vez disso, a cada participante é dado o poder de assinar a chave pública dos outros participantes que ele conhece.

A teia de confiança é formada da seguinte forma, se **A** confia em **B** e **B** assinou a chave pública de **C**, então **A** confia que a chave pública de **C** é verdadeira, mesmo sem conhecer **C,** e dessa forma continua recursivamente.

Veja o seguinte exemplo:

Suponha que *João* conhece pessoalmente *Ana* e *Paulo* e tenha assinado a chave de ambos e que *Paulo* assinou a chave de *João*.

Paulo ainda não conhece *Ana,* mas quer lhe mandar uma mensagem. Mesmo sem conhecer *Ana* pessoalmente, *Paulo* confia que *João* é uma pessoa responsável e só assinou a chave de *Ana* após verificar cuidadosamente seus documentos pessoais, logo, *Paulo* pode confiar que a chave de *Ana* é realmente dela, pois esta foi assinada por *João*.

Note que *Paulo* só confiou na assinatura de *João* porque, como ele já tinha assinado a chave de *João*, ele sabe que foi *João* mesmo quem assinou a chave de *Ana*.

Um ponto fundamental a se tratar é que todos os participantes têm a responsabilidade pela teia de confiança, caso os participantes assinem irresponsavelmente as chaves sem ter a certeza de que elas realmente pertencem ao suposto proprietário, a teia de confiança enfraquece.

3. Assinaturas Digitais

As assinaturas digitais, semelhantemente às assinaturas manuais, têm por finalidade garantir a identificação do autor do documento ou mensagem assinado, assim como a integridade de seu conteúdo até a data da assinatura. Elas podem também ser utilizadas para indicar a aceitação de contratos por parte do titular da assinatura ou para assegurar a identidade das partes que estão se comunicando.

Nos esquemas de assinaturas digitais baseados em *criptografia assimétrica*, cada usuário possui uma identidade, representada pela sua chave pública, isto é, uma seqüência de bits disponível para qualquer usuário do sistema. Para cada chave pública, existe uma chave privada, que deve estar disponível apenas ao titular da assinatura.

Uma assinatura S de uma mensagem M é calculada com a chave privada de um usuário e pode ser verificada com a respectiva chave pública, além disso, a menos que o usuário tenha assinado a mensagem M, nenhum adversário poderia produzir sua assinatura válida, nem mesmo se baseando em outras mensagens à sua escolha assinadas pelo usuário legítimo.

O primeiro esquema de assinatura digital foi apresentado em 1978 por *Rivest, Shamir* e *Adleman* no mesmo artigo em que eles propuseram o primeiro sistema criptográfico de chave pública conhecido como RSA. A segurança desse esquema é baseada na suposição de que não existe um algoritmo eficaz capaz de fatorar um produto de números primos grandes em tempo hábil. Muito embora não tenha sido provado que tal algoritmo não existe, até hoje ele não foi descoberto.

Vários outros esquemas de assinaturas digitais foram propostos, porém o mais utilizado ainda continua sendo o RSA.

Embora o esquema apresentado abaixo tenha sido baseado no RSA, a maioria dos esquemas de assinaturas digitais tem pelo menos uma das três etapas descritas abaixo.

3.1. O Algoritmo de Assinaturas Digitais passo a passo

A primeira etapa consiste na criação de um resumo da mensagem através da aplicação de uma função *hash*. Esse resumo pode garantir que a mensagem não foi modificada após ter sido assinada, pois é praticamente impossível modificar a mensagem, deixando-a com sentido, e mantendo o mesmo resumo. Além disso, os primeiros esquemas de assinaturas digitais, onde a aplicação dessa etapa não estava presente, estavam sujeitos a uma espécie de ataque conhecida como “assinaturas escolhidas” ou “mensagens escolhidas”. Nele, o adversário solicitava ao usuário que fossem assinadas mensagens sem significado, mas que poderiam ajudá-lo a descobrir o segredo de em uma mensagem real, mesmo sem a necessidade de descobrir a chave privada(mais informações sobre esse tipo de ataque podem ser vistas em [12]). Outro problema que também motivou o uso da função *hash* nas assinaturas digitais foi o fato de que a criptografia de chave pública necessitava de grandes recursos computacionais para ser processada. Aplicando a função *hash*, o resumo gerado é de tamanho fixo e pequeno, viabilizando dessa forma o uso de assinaturas digitais em mensagens grandes ou de tamanho arbitrário.

A segunda etapa utiliza o algoritmo de chave pública para encriptar o resumo da mensagem através da chave privada do remetente, gerando assim a assinatura digital. A assinatura digital, apesar de poder ser desencriptada por qualquer pessoa com a chave pública do remetente, garante a identidade do remetente, pois apenas ele, que conhece a própria chave privada, poderia ter gerado a assinatura. Opcionalmente, a mensagem em conjunto com a assinatura pode ser encriptado com a chave pública do destinatário de modo que apenas este teria acesso ao seu conteúdo.

A terceira e última etapa realiza a verificação da assinatura digital e a certificação da identidade. Nesta etapa, o destinatário desencripta a mensagem com sua chave privada (caso ela tenha sido encriptada com sua chave pública). Em seguida, ele aplica a mesma função *hash* utilizada pelo emissor na mensagem para gerar seu resumo da mensagem e compara o resumo calculado com o obtido através da desencriptação da assinatura digital da mensagem através da chave pública do emissor. Caso os dois resumos forem iguais as seguintes conclusões podem ser tiradas:

> **A mensagem está íntegra.** Caso ela tivesse sido modificada, os dois resumos – o calculado pelo destinatário e o obtido da desencriptação da assinatura digital – seriam diferentes.
>
> **O remetente da mensagem é quem diz ser**, pois somente ele possui a sua chave privada, logo, somente ele poderia ter produzido sua assinatura digital.
>
> **O remetente não pode negar que tenha enviado a mensagem**. Como somente ele possui a sua chave privada, ninguém poderia ter produzido a sua assinatura digital em seu lugar.

A figura a seguir ilustra o funcionamento de um esquema de assinaturas digitais.

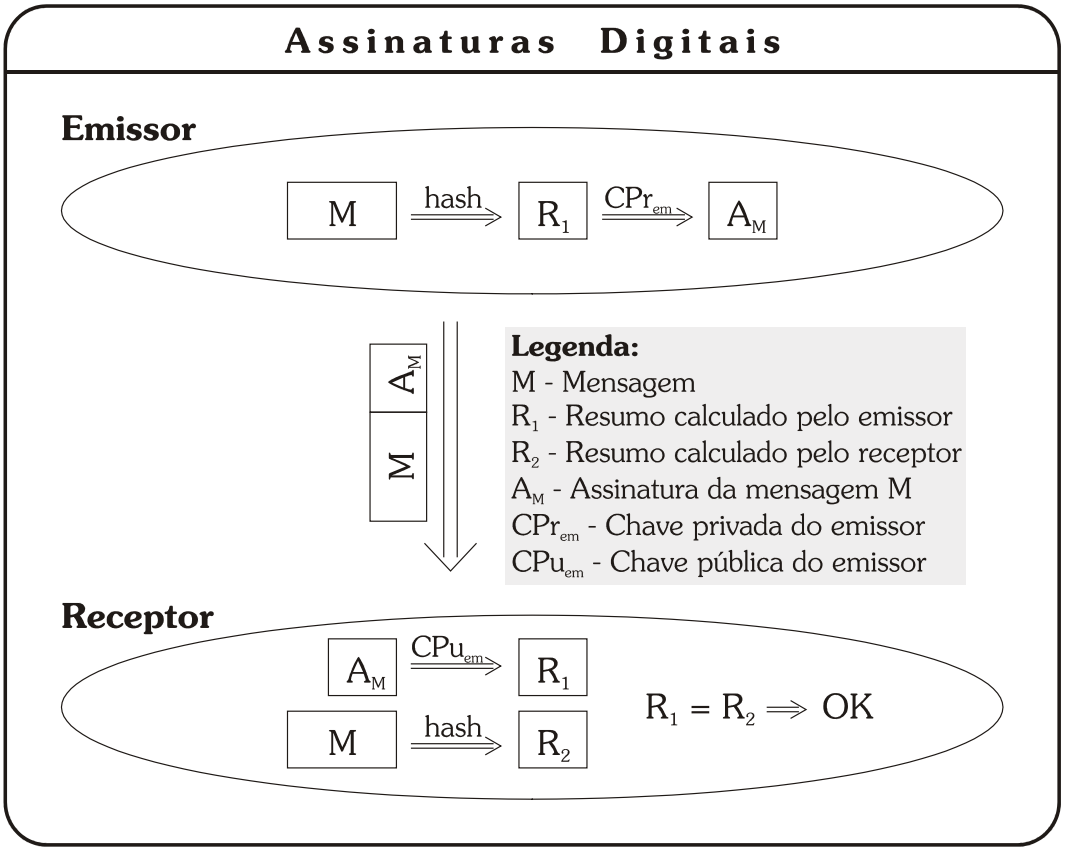

Figura 1: Esquema de Assinaturas Digitais passo a passo

Podemos concluir que todo o esquema de Assinaturas Digitais necessita que duas condições básicas sejam satisfeitas.

i) Ninguém, além do titular, conhece uma determinada chave privada;

ii) As chaves públicas estão disponíveis para o conhecimento de todos e a partir delas podemos obter informações sobre o seu titular.

A primeira condição deve ser garantida pelo titular da chave privada, que tem a obrigação de mantê-la em segurança, e se ocorrer de alguém ter descoberto a sua chave privada, ele deve comunicar o fato imediatamente à autoridade certificadora que tem seu cadastro para que sua chave pública seja revogada. Existem vários problemas que o usuário enfrenta para manter sua chave em segredo. Esses problemas estão fora do escopo deste trabalho, uma descrição deles pode ser encontrada em [10].

A segunda condição diz respeito ao problema de autenticação e distribuição de chaves. Essas tarefas normalmente são realizadas pelas Autoridades Certificadoras através dos Certificados Digitais. As autoridades certificadoras podem ser entendidas como “cartórios digitais”, pois elas recebem requisições de certificação digital de outras entidades, validam as requisições por meio de análise de documentação da entidade solicitante e emite o certificado digital a partir do momento que estiver convencida sobre a veracidade dos dados que estarão contidos no certificado.

O certificado digital é um documento que contém diversas informações sobre o titular de uma chave pública. Entre as informações contidas no certificado digital podemos destacar o nome do dono do certificado, sua data de emissão, seu prazo de validade, o nome do emissor do certificado, a chave pública do dono do certificado, entre outras informações. Este documento é assinado digitalmente pela autoridade certificadora emissora do certificado.

Desta maneira, uma chave pública obtida a partir de um certificado digital assinado por uma autoridade certificadora, à qual se atribui confiança, pode ser considerada autêntica.

Até aí tudo parece perfeito, mas como podemos garantir que a assinatura digital contida em um certificado qualquer é de uma autoridade certificadora válida? Ou seja, se um certificado digital não foi emitido por uma autoridade certificadora conhecida de ambas as partes em nada ele ajudaria, pois teríamos agora apenas mais uma assinatura digital desconhecida.

Por esse motivo, a autoridade certificadora também deve ter o seu próprio certificado digital emitido por uma outra autoridade certificadora, e esse procedimento se repete até atingir uma autoridade certificadora que é de confiança de todos os participantes. Essas autoridades certificadoras que estão no nível mais alto da hierarquia são chamadas de autoridades certificadoras raízes e elas não precisam ter o certificado digital para garantir sua idoneidade, pois elas possuem as suas chaves públicas conhecidas por

qualquer aplicação que estiver preparada para utilizar as assinaturas digitais. Normalmente as autoridades certificadoras raízes são gerenciadas pelo órgão competente que regulamenta o uso das assinaturas digitais.

Sempre que se fizer necessário verificar um certificado digital, deve ser percorrido o caminho a partir do certificado digital até chegar à autoridade certificadora raiz.

3.2. Cálculo da assinatura RSA

Geração das Chaves

- Gere aleatoriamente dois números primos grandes p e q;
- Calcule o módulo público n que é o produto de p e q $(n = pq)$**;**

  Calcule $\phi(n) = (p \quad 1)(q \quad 1);$
- Gere aleatoriamente um número e que seja inversível módulo $\phi(n)$ e que $1 \leq e \leq \phi(n)$**;**
- Por fim calcule d utilizando o algoritmo de Euclides estendido de modo que $de \equiv 1 \ (mod \ \phi(n))$.

A chave pública será o par (e, n) e a chave privada será o par (d, n). Todos os demais números devem ser mantidos secretos, sobretudo p e q.

Calculando a Assinatura

- Calcule o *hash* da mensagem M;
- Pré-codifique o *hash*, representando-o como um inteiro, caso o inteiro seja maior que n, ele deve ser dividido em blocos de inteiros menores que n**.** Chamaremos esse *hash* pré-codificado de H**.**
- Calcule a assinatura $A = H^d \ mod \ n$**.**

Verificando a Assinatura

- Calcule $H = A^e \ mod \ n$.

Note que $A^e \ mod \ n = (H^d)^e \ mod \ n = H^{de} \ mod \ n = H$**.**

3.3. Cálculo da assinatura DSS

Geração das Chaves

• Gere aleatoriamente um número primo p de tamanho l, onde l é múltiplo de 64 e $512 \leq l \leq 102$;

• q é um primo com 160 bit de tamanho, divisor de $p \quad 1$;

• g é um elemento da ordem de q em Z_p^*, a tipla (p, q, g) é público;

• x, a chave secreta, é um número aleatório tal que: $1 \leq x \leq q$;

• $y = g^x \bmod p$ é a chave pública de verificação.

Calculando a Assinatura

• Calcule o *hash* da mensagem M;

• Pré-codifique o *hash*, representando-o como um inteiro, caso o inteiro seja maior que q, ele deve ser dividido em blocos de inteiros menores que q. Chamaremos esse *hash* pré-codificado de H.

• Escolha um número k aleatório $(1 \leq K < q)$ e com $K^{-1} \bmod q$ calcule:

$$r = (g^{k^{-1}} \bmod p) \bmod q$$
$$s = k(H + xr) \bmod q$$

• o par (r, s) é a assinatura.

Verificando a Assinatura

• Verifique que:

$r = (g^{ms^{-1}} y^{rs^{-1}} \bmod p) \bmod q$, com s^{-1} calculado $\bmod q$

4. Compartilhamento de Segredo

Nos esquema de compartilhamento de segredo, o segredo **S** é dividido em várias partes, distribuídas entre os **P** participantes de uma maneira que qualquer **subconjunto qualificado** de participantes **K** possa colaborar com suas partes a fim de recuperar o segredo. Esquemas de compartilhamento de segredo têm inúmeras aplicações em diferentes áreas onde a responsabilidade para alguma determinada ação não pode estar nas mãos de apenas uma pessoa, como, por exemplo: setor de compras de uma grande empresa, lançamento de mísseis, transações envolvendo grandes quantias entre agências bancárias e etc.

Além dos participantes que compartilham o segredo, existe também um participante especial, chamado distribuidor que é responsável por repartir o segredo **S** e distribuir as partes entre os participantes.

O conjunto formado por todos os subconjuntos qualificados de participantes é chamado de estrutura de acesso A. A geralmente é monotônica, isto é, se $X \in A$ e $X \subseteq X' \Rightarrow X' \in A$**.**

Um subconjunto qualificado mínimo é aquele em que se $Y \in A$**,** qualquer $Y' \subset Y$, $Y' \neq Y \Rightarrow \ Y' \notin A$. A base de A, chamada A_0 é o conjunto que contém todos os subconjuntos qualificados mínimos.

Um esquema de compartilhamento de segredo é perfeito quando qualquer subconjunto não qualificado de participantes não possa obter nenhuma informação sobre o segredo.

4.1. Esquema de Criptografia de Limiar

É um caso especial de esquema compartilhamento de segredo onde qualquer subconjunto de participantes com pelo menos um número determinado de participantes é um subconjunto qualificado, esse número determinado de participantes é chamado de limiar. Nesse tipo de esquema, a parte do segredo que cada participante recebe é denominada de sombra.

Os esquemas de criptografia de limiar recebem a notação **(m, n)**, onde **m** é o número mínimo de sombras necessárias para a recuperação do segredo e **n** é o número total de participantes.

Os seguintes requisitos devem ser satisfeitos nos esquemas de criptografia de limiar:

• Nenhum grupo de participantes, menor que o limiar **m** definido, pode descobrir o segredo compartilhado, mesmo que eles cooperem entre si;

• Sempre que for necessária a recuperação do segredo, um número de participantes, maior que o limiar dado, pode fazê-la.

4.2. Implementação por Interpolação Polinomial de Lagrange

Esta implementação foi proposta por Adi Shamir em [6], por isso também é conhecido como esquema de compartilhamento de segredo de Shamir. Escolha um primo p, que é maior que o número das possíveis sombras e maior que o maior segredo possível. Para compartilhar o segredo, gere um polinômio arbitrário de grau $m - 1$, onde m é o limiar.

$$P(x) = (ax^{m-1} + bx^{m-2} + cx^{m-3} + ... + M) \; mop \; p$$

Onde:

- p é um primo aleatório maior que todos os coeficientes, que deve ser mantido público.
- Os coeficientes $(a, b, c,...)$ são escolhidos aleatoriamente por uma distribuição uniforme entre 0 e p; eles são mantidos em segredo e descartados após a geração das sombras;
- M é a mensagem;
- As n sombras são obtidas calculando o polinômio em n diferentes pontos. $K_i = F(x_i)$;

Como o polinômio de grau $m - 1$ tem m coeficientes desconhecidos (incluindo M), quaisquer m sombras podem ser utilizadas para reconstruir o polinômio através do algoritmo de interpolação de Lagrange. $m - 1$ sombras não podem recriar o segredo. $m + 1$ sombras recriam o segredo, contendo informação redundante.

Propriedades:

a) O tamanho de cada sombra não excede o tamanho do segredo;
b) Mantendo o m fixo, K_j sombras podem ser adicionados ou removidos dinamicamente, sem a necessidade da reconstrução de todas os sombras (para adicionar ou remover novos participantes);
c) É fácil mudar todas as sombras K_i sem a necessidade de mudar o segredo M, para isso, basta utilizar outro polinômio P_2.
d) Não está restrito à esquemas de criptografia de limiar podendo ser utilizado na implementação de esquemas de compartilhamento de segredo mais complexos.

4.3. Exemplo:

Suponha um esquema de criptografia de limiar $(3,5)$, logo, o polinômio $P(x)$ será de grau 2: $P(x) = ax^2 + bx + M \bmod n$

1. Suponha o segredo $M = 80$.
2. Gera-se o primo $n > M$, digamos $n = 113$.
3. Os coeficientes a e b são gerados aleatoriamente entre os valores de 0 e n: considere $a = 12$ e $b = 47$, temos:

$$P(x) = 12x^2 + 47x + 80 \bmod n$$

4. Calcula-se as sombras para os 5 participante:

$P(1) = 26;$

$P(2) = 109;$

$P(3) = 103;$

$P(4) = 8;$

$P(5) = 50;$

5. Cada participante K_i recebe a sombra $P(i)$.

6. Suponha agora que os participantes K_1, K_3 e K_4 desejam reconstruir o segredo. Através da interpolação polinomial de Lagrange temos:

$$P(x)=\frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1+\frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2+\frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

com:

$$x_1=1; x_2=3; x_3=4; y_1=26; y_2=103; y_3=8$$

Logo:

$$P(x)=\frac{(x-3)(x-4)}{(1-3)(1-4)}26+\frac{(x-1)(x-4)}{(3-1)(3-4)}103+\frac{(x-1)(x-3)}{(4-1)(4-3)}8 \pmod{113}\therefore$$

$$P(x)=\frac{x^2-3x-4x+12}{(-2)(-3)}26+\frac{x^2-1x-4x+4}{(2)(-1)}103+\frac{x^2-1x-3x+3}{(3)(1)}8 \pmod{113}\therefore$$

$$P(x)=\frac{x^2-7x+12}{6}26+\frac{x^2-5x+4}{-2}103+\frac{x^2-4x+3}{3}8 \pmod{113}\therefore$$

$$P(x)=\left(\frac{26(x^2-7x+12)}{6} \pmod{113}\right)+\left(\frac{103(x^2-5x+4)}{-2} \pmod{113}\right)$$
$$+\left(\frac{8(x^2-4x+3)}{3} \pmod{113}\right) \pmod{113}\therefore$$

$$mas,\ \frac{1}{6}\equiv 19 \pmod{113}; \frac{1}{-2}\equiv 56 \pmod{113}; \frac{1}{3}\equiv 38 \pmod{113}:$$

$$P(x)=\left(19\cdot 26(x^2-7x+12) \pmod{113}\right)+\left(56\cdot 103(x^2-5x+4) \pmod{113}\right)$$
$$+\left(38\cdot 8(x^2-4x+3) \pmod{113}\right) \pmod{113}\therefore$$

$$P(x)=\left(42(x^2-7x+12) \pmod{113}\right)+\left(5(x^2-5x+4) \pmod{113}\right)$$
$$+\left(78(x^2-4x+3) \pmod{113}\right) \pmod{113}\therefore$$

$$P(x)=\left(42x^2+45x+52 \pmod{113}\right)+\left(5x^2+88x+20 \pmod{113}\right)$$
$$+\left(78x^2+27x+8 \pmod{113}\right) \pmod{113}\therefore$$

$$P(x) = \left(42x^2 + 45x + 52 \pmod{113} \right) + \left(5x^2 + 88x + 20 \pmod{113} \right) + \left(78x^2 + 27x + 8 \pmod{113} \right) \pmod{113} \therefore$$

$$P(x) = \left(42x^2 + 45x + 52\right) + \left(5x^2 + 88x + 20\right) + \left(78x^2 + 27x + 8\right) \pmod{113} \therefore$$

$$P(x) = \left(125x^2 + 160x + 80\right) \pmod{113} \therefore$$

$$P(x) = 12x^2 + 47x + 80 \pmod{113} \therefore$$

O polinômio foi reconstruído com apenas as seguintes informações K_1, K_3, K_4 e n, permitindo assim redescobrir o segredo.

Obs: Para calcular os *inversos mod n* foi usado o Algoritmo de Euclides Estendido.

O algoritmo apresentado acima é bastante flexível e pode ser utilizado para construir esquemas de compartilhamento de segredo mais complexos que os esquemas de limiar, podendo ser usado para construir esquemas onde uma pessoa é mais importante que outras, ou satisfazendo as mais diversas condições.

Por exemplo, se quisermos criar um esquema onde para recriar o segredo são necessários 4 funcionários de um departamento, mas, se o diretor do departamento estiver presente, apenas mais 1 funcionário qualquer é necessário, basta disponibilizar 3 sombras para o diretor mais 1 sombra para cada funcionário.

Outros tipos de cenários podem ser imaginados, por exemplo, se quisermos distribuir o segredo de modo que precisemos ter dois funcionários do departamento **A** e dois funcionários do departamento **B** de uma empresa para recuperar o segredo, mas nem 4 funcionários do departamento **A**, nem 4 funcionários do departamento **B**, nem mesmo 3 funcionários de um departamento e 1 funcionário de outro departamento possam recuperar o

segredo. Para implementar esse esquema nós geramos um polinômio de grau 2 sendo o produto de dois polinômios de grau 1, um para cada departamento, e geramos as sombras dos funcionários de cada departamento utilizando seus respectivos polinômios de grau 1. Quaisquer 2 sombras de funcionários pertencentes a um mesmo departamento pode recriar o polinômio utilizado para gerar as sombras de seu departamento, mas não importa quantos participantes do mesmo departamento estejam envolvidos, eles nada podem descobrir sobre o segredo, salvo com a colaboração de pelo menos dois funcionários do outro departamento.

Esquema de Junção do Compartilhamento de Segredo de Shamir

É um esquema de compartilhamento de segredo que não necessita do distribuidor de sombra, e cujo segredo não é conhecido por nenhum participante. Cada participante gera um valor aleatório y_i, e, usando o esquema de compartilhamento de segredo de Shamir, compartilha sombras de seus valores y_i aos demais participantes. A sombra resultante é calculada adicionando-se cada sombra *parcial* proveniente de cada participante.

5. Desenvolvendo um Esquema

Desenvolver protocolos de criptografia distribuída é um desafio muito grande, ainda mais difícil é desenvolver protocolos bem analisados. A principal dificuldade na análise está em definir um modelo matemático que seja ao mesmo tempo forte o suficiente para modelar as situações reais e ainda assim não seja muito complexo a ponto de inviabilizar a análise.

A principal característica dos modelos de segurança para protocolos distribuídos é se o adversário tolerado é *estático* ou *dinâmico*. Os dois tipos de adversários podem corromper um subconjunto conforme especificado no modelo. A diferença está relacionada ao comportamento do adversário, pois, enquanto que o adversário estático escolhe o subconjunto de participantes que irá atacar antes do início da execução do protocolo, o adversário dinâmico toma essa decisão durante a execução do protocolo baseado nas informações adquiridas durante a execução.

Uma grande quantidade trabalhos publicados sobre a segurança de protocolos criptográficos tem obtido sucesso apenas no modelo estático de adversário, isto se deve à grande dificuldade de se provar a segurança no modelo de adversário dinâmico.

Outro conceito importante para prova de segurança de protocolos de criptografia distribuída é o modelo do oráculo aleatório. O modelo do oráculo aleatório é um modelo computacional em que é assumido que uma dada função *hash* se comporta como uma função aleatória ideal, ou seja, como uma função cujo domínio de entrada é o conjunto de todas as *strings* binárias, onde cada string binária é mapeada para uma string binária de mesmo tamanho. Embora esta suposição seja evidentemente falsa, ela corrobora a observação de que funções *hash* comportam-se essencialmente como uma caixa preta, e suas saídas não podem ser determinadas até serem calculadas.

Provas de segurança no modelo do oráculo aleatório pode apenas ser considerado como uma heurística e nem sempre implicam em prova de

segurança no mundo real. Mas apesar de que os esquemas comprovadamente seguros no modelo do oráculo aleatório não poderem ser considerados como seguros no mundo real, eles não são considerados inseguros, apenas se considera que ainda não foi provada sua segurança em um modelo mais forte.

Gennaro, Halevi e Rabin em [17], e Cramer e Shoup em [18] propuseram o primeiro esquema de assinaturas cuja eficiência é apropriada para uso prático e cuja análise de segurança não assume uma função aleatória ideal. Seus esquemas são seguros sobre o modelo da suposição RSA forte.

As seguintes considerações também devem ser feitas durante a modelagem do protocolo:

O tamanho do limiar. Que fração de participantes podem ser pode ser controlada sem prejudicar o serviço ou revelar informações confidenciais? Quanto menor o limiar, menos participantes o adversário precisa controlar para descobrir o segredo, quanto maior o limiar, mais mensagens precisam ser trocadas entre os participantes durante a execução do protocolo, reduzindo assim a eficiência do sistema.

Eficiência. Quanto de comunicação, armazenamento e poder computacional o protocolo requer?

Modelo de Comunicação. Que características do modelo de comunicação são requeridas? Comunicação síncrona ou parcialmente síncrona? Broadcast autenticado? Conexão segura entre os participantes?

Capacidade do Adversário. Os adversários podem ser divididos em três grupo de acordo com as suas habilidades.

• *Escuta (Eavesdropping)* – tem acesso a todas as informações dos participantes controlados, além das mensagens que passam pelo canal de broadcasting.

• *Interrupção (Halting)* – pode impedir os participantes corrompidos de enviar suas mensagens durante a execução do protocolo.

• *Intrusão (Malicious)* pode agir sobre os participantes controlados de qualquer forma, inclusive, participando do protocolo como se fosse um participante autêntico.

5.1. Conceitos Preliminares

As seguintes definições são importantes no desenvolvimento do esquema.

Primos Seguros. Um primo p é chamado primo seguro se $p = 2p + 1$ e p também é um número primo.

Módulo RSA. Um número ***n*** é chamado módulo RSA se $n = pq$, com p e q sendo números primos.

Módulo RSA especial. Um módulo RSA $n = pq$ é especial se p e q forem números primos seguros.

Suposição RSA. Dada uma chave pública RSA (n, e) e um valor aleatório $u < \phi(n)$, é computacionalmente difícil calcular o valor v tal que $v^e \equiv u \bmod n$.

Suposição RSA forte. Dados um módulo RSA n e um valor $u < \phi(n)$, é computacionalmente difícil computar valores v e e tal que $v^e \equiv u \bmod n$.

Protocolos robustos. São os protocolos que mantêm seu correto funcionamento mesmo na presença de adversários controlando alguns participantes.

6. O Esquema Básico

O esquema proposto neste trabalho de graduação tem por objetivo aplicar um modelo de criptografia de limiar nas autoridades certificadoras de esquemas de assinaturas digitais RSA regulares, com o objetivo de anular um possível ponto de falha, no caso, a autoridade certificadora. Dividindo a tarefa da autoridade certificadora em vários servidores, dividimos também o seu poder de emitir certificados digitais, desta forma, um adversário que puder controlar apenas um subconjunto de servidores não poderá forjar um certificado.

Como um certificado digital nada mais é do que um documento assinado digitalmente pela autoridade certificadora, e toda a estrutura dos esquemas de assinaturas digitais tradicionais descritas no Capítulo 3 será mantida (fica totalmente transparente ao usuário final as modificações sugeridas), a definição deste esquema se resume a descrever o protocolo de assinaturas digitais RSA por limiar.

6.1. Modelagem do esquema:

Modelos de Comunicação. O modelo computacional é composto de l participantes $\{P_1, ..., P_l\}$ que podem ser modelado por máquinas de Turing randômicas de tempo polinomial. Eles são completamente conectados por canais ponto-a-ponto seguros. Além disso, existe um canal broadcast no qual uma mensagem enviada por um participante P_i é reconhecida por todos participantes como sendo originada de P_i.

O Adversário. Assume-se que o adversário é estático e pode corromper até t - 1 dos l servidores, considerando o mais perigoso tipo de adversário, isto é, o *Intruso (Malicious),* que pode fazer os servidores corrompidos se comportarem de qualquer forma maliciosa. O adversário tem o poder computacional adequadamente modelado uma por máquinas de Turing randômicas de tempo polinomial, de fato, é suficiente considerar que o adversário não pode forjar uma chave RSA regular.

Existe um distribuidor de chaves confiável. A geração da chave só precisa ser feita uma única vez e pela própria autoridade certificadora, desta forma, podemos assumir que ela é feita durante a inicialização do sistema, onde pode se presumir que não há presença de adversários.

6.2. Geração de Chaves.

A propriedade do módulo RSA ser composta por um produto de dois primos seguros causa uma grande dificuldade no problema de geração distribuída de chaves. Apesar das inúmeras tentativas para solucionar este problema, até a presente data ainda não existe um algoritmo capaz de faze-lo de forma eficiente e mantendo todas as características desejadas.

Neste esquema, a geração de chaves é feita por apenas um servidor (o distribuidor de sombras) durante a inicialização do sistema, seguindo os passo abaixo:

- Escolha primos seguros *(p e q)* de igual tamanho, onde $p = 2p + 1$ e $q = 2q + 1$.
- Calcula-se módulo RSA $n = pq$, e $m = p \ q$.

  Gera aleatoriamente um número primo $e > 1$ para ser o expoente público RSA.
- Calcule d como o inverso de $e \bmod m$ ($de \equiv 1 \pmod m$).
- Faça $a_0 = d$ e escolha a_i aleatoriamente $(0 \le a_i \le m - 1)$, para $1 \le i \le t \quad 1$. Os valores $a_0,..., a_{t\ 1}$ definem o polinômio $P(x) = \sum_{i=0}^{t-1} a_i x^i$
- Calcule as sombras $s_i = P(i) \bmod m$.
- Cada servidor i recebe a sombra s_i.
- Seja Q_n um subgrupo de quadrados em Z_n^*. Escolhe-se aleatoriamente um número $v \in Q_n$ e calcula-se os valores $v_i = v^{s_i} \in Q_n$ para todos $1 \le i \le l$. Esses elementos são as chaves de verificação. $VK = v$ e $VK_i = v_i$

6.3. Assinando uma Mensagem.

Cada participante calcula sua assinatura parcial da mensagem utilizando sua sombra, em seguida as assinaturas parciais são combinadas na assinatura final da mensagem.

Antes de partir para o cálculo da assinatura combinada, algumas considerações devem ser feitas:

Para qualquer subconjunto de t pontos em $\{0,...,l\}$, o valor de $P(x) \bmod m$ determina unicamente os coeficientes de $P(x)$. Seja $\Delta = l!$. Para qualquer subconjunto S de k pontos em $\{0,...,l\}$, e para qualquer $i \in \{0,...,l\} \setminus S$, e $j \in S$, podemos definir:

$$\lambda_{i,j}^{S} = \Delta \frac{\prod j' \in S \setminus \{j\}(i - j')}{\prod j' \in S \setminus \{j\}(j - j')} \in Z \qquad (1)$$

Esses valores foram derivados da fórmula de interpolação polinomial de Lagrange. Temos também que:

$$\Delta \cdot P(i) \equiv \sum_{j \in S} \lambda_{i,j}^{S} p(j) \bmod m \qquad (2)$$

Cálculo das Assinaturas Parciais

Primeiro utiliza-se uma função H (*hash*) na mensagem que mapeia mensagens em elementos de Z_n^*. Seja $x = H(m)$. A assinatura parcial de cada participante na mensagem m é dada por: $x_i = x^{2\Delta s_i} \in Q_n$, Em [19] encontra-se a prova de corretude da fórmula acima.

Combinando as Assinaturas Parciais

Supondo que temos assinaturas parciais de um conjunto de S participantes, onde $S = \{i_j,...,i_k\} \subset \{i_1,...,i_l\}$. Assumindo que $x_{ij}^2 = x^{4\Delta S_{ij}}$, para combinar as assinaturas parciais, calcula-se:

$$w = x_{i_j}^{2\lambda_{0,i_1}^{S}} ... x_{i_k}^{2\lambda_{0,i_k}^{S}},$$

onde os λ's são inteiros definidos em (1). De (3) temos que $w^e = x^e$,com:

$$e = 4\Delta^2$$

Sabe-se o $mdc(e, e) = 1$, logo, é fácil calcular a assinatura y ($y^e \ mod \ n = x$), usando um algoritmo padrão: $y=x^a w^b$, onde a e b são inteiros tal que $e^a + e^b = 1 \ (mod \ n)$. a e b e podem ser obtidos pelo teorema de Euclides estendido.

6.4. Verificação de Assinaturas:

A verificação da assinatura é feita da mesma forma que nos esquemas de assinaturas RSA regulares:

- Calcule $x = y^e \ mod \ n$.

A segurança deste modelo é provada utilizando o modelo do oráculo aleatório e assumindo que o esquema de assinaturas RSA padrão é seguro. Detalhes da prova podem ser encontrados em [19].

7. Conclusões

O uso da criptografia de limiar deve ser levado em conta para o aumento de segurança em esquemas onde um “poder” está concentrado em uma única entidade. No caso das assinaturas digitais este “poder” diz respeito à geração de certificados digitais, estando concentrado nas entidades certificadoras.

Outro fator que incentiva a utilização da criptografia de limiar nos servidores das autoridades certificadoras é o fato de que a migração para o novo sistema seria feita de forma muito direta. Toda estrutura de certificação continua da mesma forma, os usuários finais nem mesmo precisariam ter conhecimento desta mudança, que se faz totalmente apenas nos servidores das autoridades certificadoras.

Dificuldades Encontradas

As principais dificuldades encontradas dizem respeito à complexidade dos conceitos matemáticos em que se fundamenta a criptografia distribuída.

Desenvolver um protocolo de criptografia distribuída é uma tarefa inerentemente difícil, ainda mais difícil é desenvolver um protocolo que seja totalmente analisável por modelos matemáticos, onde possa ser provada sua segurança.

Outra dificuldade é que esta é uma área muito nova e muitos de seus tópicos ainda não foram exaustivamente estudados, não existe, por exemplo, um livro de referência sobre criptografia de limiar.

8. Trabalhos Futuros

Uma sugestão de continuidade deste trabalho seria o aprimoramento do protocolo geração das chaves RSA, na tentativa de torna-lo distribuído e eficiente.

Pode-se também estender a segurança oferecendo suporte contra adversários dinâmicos.

Inclui-se também como sugestão de aprimoramento deste trabalho uma análise de segurança do esquema de criptografia distribuída utilizando primeiramente os modelos de oráculo aleatório em seguida o da suposição RSA forte.

A realização de testes práticos para definição dos parâmetros ***(m, n)*** e análise de performance também seria um bom exercício para continuidade deste trabalho.

Apêndice 1 Glossário

Conceitos de Segurança

- **Confidencialidade** Garante que a mensagem transmitida possa ser lida apenas por pessoas autorizadas.
- **Autenticação** Garante que o emissor da mensagem não é falso.
- **Integridade** Garante que a mensagem não foi modificada.
- **Não Repúdio** Garante que o emissor de uma mensagem não negue sua transmissão.
- **Disponibilidade** Garante que um recurso estará sempre disponível para uma entidade autorizada.

Função Unidirecional (One-Way) São funções que são relativamente fáceis de computar, porém difíceis de reverter. Isto é, dado *x*, é fácil calcular *f(x)*, porém, a partir de *f(x)* é muito difícil encontrar *x*.

Funções Unidirecionais com arapuca (Trapdoor) Similar ao que ocorre com a função unidirecional, é fácil de computar e difícil de reverter, porém, com a utilização de uma informação secreta adicional (arapuca) se torna fácil reverter.

Funções Hash É uma função unidirecional que recebe como entrada um string de tamanho variável, e o converte em um string de tamanho fixo (geralmente menor) denominado hash. Cada bit modificado na mensagem original acarreta numa grande modificação do hash. Desta forma, gerar uma mensagem com sentido a partir do hash é extremamente difícil.

Criptografia Simétrica Se caracteriza por possuir uma única chave para criptografar e decriptografar. Por esse motivo, a chave deve necessariamente ser do conhecimento do emissor e do receptor da mensagem. Sua principal vantagem é que ela é extremamente eficiente quando comparada com a criptografia assimétrica em relação ao tempo de execução. Como sua principal desvantagem podemos citar a necessidade de distribuir a chave secreta entre as partes.

Criptografia Assimétrica Também conhecida como criptografia de chave pública, se caracteriza por possuir duas chaves distintas, uma privada, que deve ser de conhecimento apenas do proprietário da chave e a outra pública, que pode ser livremente divulgada sem que implique na quebra da segurança do sistema. As duas chaves se relacionam de forma que tudo o que for criptografado com uma chave só poderá ser decriptografado com a outra. Tem como vantagem ter solucionado o problema de distribuição de chaves, pois a chave pública pode ser livremente divulgada. Tem como desvantagem principal o desempenho muito inferior quando comparada com a criptografia simétrica.

Interpolação Polinomial de Lagrange é um método para determinação de um polinômio que passa por um conjunto de pontos dados. Para cada conjunto de pontos existe apenas um polinômio que passa por todos eles.

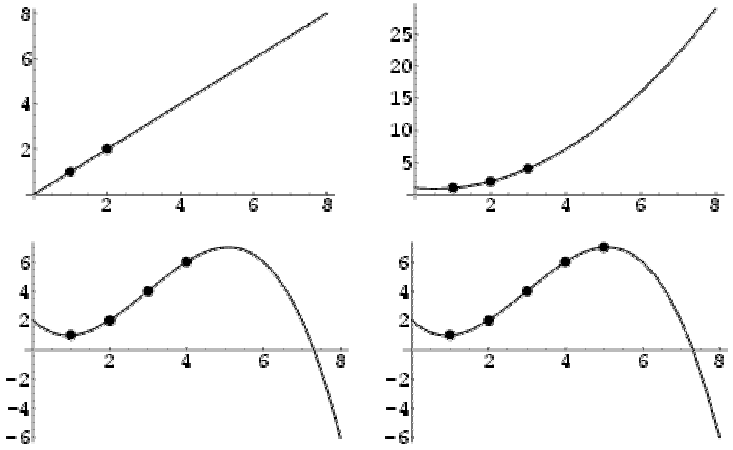

Figura 2: Interpolação Polinômial de Lagrange

O polinômio interpolado de Lagrange é o polinômio de grau ***n*** ***1*** que passa por todos os ***n*** pontos $y_i = f(x_i)$. Ele é calculado por:

$$P(x) = \sum_{j=1}^{n} P_j(x),$$

Onde:

$$P_j(x) = y_j \prod_{\substack{k=1 \\ k \neq j}}^{n} \frac{x - x_k}{x_j - x_k}.$$

Escrevendo explicitamente:

$$\begin{aligned} P(x) &= \frac{(x-x_2)(x-x_3)\cdots(x-x_n)}{(x_1-x_2)(x_1-x_3)\cdots(x_1-x_n)}y_1 \\ &\quad + \frac{(x-x_1)(x-x_3)\cdots(x-x_n)}{(x_2-x_1)(x_2-x_3)\cdots(x_2-x_n)}y_2 + \cdots \\ &\quad + \frac{(x-x_1)(x-x_2)\cdots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\cdots(x_n-x_{n-1})}y_n. \end{aligned}$$

Por exemplo, para 3 pontos:

$$P(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

R e f e r ê n c i a s

[1] Schneier, Bruce
"*Applied Cryptography: protocols, algorithm, and source code in C*", Second Edition, John Wiley & Sons, Inc.

[2] Coutinho, S. C.
"*Números Inteiros e Criptografia RSA*". Rio de Janeiro, IMPA/SBM, 1997.

[3] Shafi Goldwasser, Mihir Bellare
"*Lectures Notes on Cryptography*",
Summer course in criptography. MIT, 1996-2001

[4] Cryptography and Information Security Group Research Project:
"Threshold Cryptology",
http://theory.lcs.mit.edu/~cis/cis-threshold.html

[5] Threshold Cryptography,
http://www.cs.fsu.edu/~desmedt/topics-threshold.html

[6] Shamir, Adi
"How to Share a Secret", Massachusetts Institute of Technology. 1979

[7] Lysyanskaya, Anna
"*Signature Schemes and Applications to Cryptographic Protocol Design*", Phd Thesis, Massachusetts Institute of Technology. Setembro de 2002.

[8] Hung-Min Sun and Shiuh-Pyng Shieh
"Constructing Perfect Secret Sharing Schemes for General And Uniform Access Structures",
Journal of Information Science and Engineering 15, 678-689 (1999)

[9] Eric W. Weisstein.
"Lagrange Interpolation Polynomial".
From MathWorld – A Wolfram Web Resource
http://mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html

[10] Rezende, Pedro Antônio Dourado
"Certificados Digitais, Chaves Públicas e Assinaturas: O que são, como funcionam e como não funcionam". Universidade de Brasília, 2000.
http://www.cic.unb.br/docentes/pedro/trabs/cert.htm

[11] National Institute of Standards and Technology, NIST FIPS PUB 186, "Digital Signature Standard", U.S. Department of Commerce, May, 1994
http://www.itl.nist.gov/fipspubs/fip186.htm

[12] Denning, Doroty E.
"Digital Signatures with RSA and Other Public Key Cryptosystems",
Communications of the ACM, April 1984 Volume 27 Number 4.

[13] Y. Desmedt and Y. Frankel.
"Shared generation of authenticators and signatures",
CRYPTO 91, pages 457–469. Springer-Verlag, 1992.
Lecture Notes in Computer Science No. 576.

[14] A. De Santis, Y. Desmedt, Y. Frankel, and Moti Yung.
"How to share a function securely",
ACM Symp. on Theory of Computing, pages 522–533. Santa Fe, 1994.

[15] R.Gennaro, S.Jarecki, H.Krawczyk, T.Rabin
"Robust Threshold DSS Signatures", Advances in Cryptology:
Proc. Eurocrypt'96, Lecture Notes in Computer Science 1070,
Springer, (1996), pp. 354--371.

[16] R. L. Rivest, A. Shamir, and L. Adleman.
"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". *Communications of the ACM*. Fevereiro de 1978.

[17] Rosario Gennaro, Shai Haveli, and Tal Rabin.
"Secure hash-and-sign signaturees without the random oracle".
Advances in Cryptology, Eurocrypt' 99.

[18] Ronald Cramer and Victor Shoup.
"Signature schemes based on the strong RSA assumption".
ACM press. Nov 1999.

[19] Victor Shoup
"Practical Threshold Signatures", *Proceedings of EuroCrypt 2000*, Springer Verlag LNCS series nr. 1807.

[20] Ivan Damgard and Maciej Koprowski
"*Pratical Threshold RSA Signatures Without a Trusted Dealer*"
Technical report, Aarhus University, BRICS, November 2000.