UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Fernando Cézar de Oliveira Lopes

Denúncia Anônima Segura

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de mestre em Ciência da Computação.

Prof. Ricardo Felipe Custódio, Dr.
Orientador

Florianópolis, Fevereiro de 2003

Denúncia Anônima Segura

Fernando Cézar de Oliveira Lopes

Esta Dissertação foi julgada adequada para a obtenção do título de mestre em Ciência da Computação, área de concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

_____

Prof. Fernando Ostuni Gauthier, Dr.

Coordenador do Curso

Banca Examinadora

_____

Prof. Ricardo Felipe Custódio, Dr.

Orientador

_____

Prof. Carlos Alberto Maziero, Dr.

_____

Prof. Carlos Roberto De Rolt, Dr.

“Exultai e alegrai-vos, porque é grande o vosso galardão nos céus; porque assim perseguiram os profetas que foram antes de vós. Vós sois o sal da terra; e se o sal for insípido, com que se há de salgar? para nada mais presta senão para se lançar fora, e ser pisado pelos homens. Vós sois a luz do mundo: não se pode esconder uma cidade edificada sobre o monte; Nem se acende a candeia e se coloca debaixo do alqueire, mas no velador, e dá luz a todos que estão na casa´´. Mt. 5: 12 - 15. [Bíblia Sagrada]

Este trabalho dedico à minha esposa Jecemeri e minhas filhas Mariana e Débora com muito amor e carinho.

Agradecimentos

Em primeiro lugar a DEUS, pois fez tudo e a Ele tudo pertence, permitiu-me fazer este curso debaixo de sua graça.

À Universidade Federal de Santa Catarina, em especial ao Departamento de Pós-Graduação em Ciências da Computação.

Gostaria de agradecer especialmente ao Prof. Ricardo Felipe Custódio pelo seu apoio, compreensão e conselhos, pois um trabalho bem feito sempre tem um mentor na orientação.

Gostaria também de agradecer de uma forma muito especial à minha esposa **Jecemeri** e minhas filhas **Mariana** e **Débora**, pois foram privadas da minha companhia por vários e vários momentos.

Aos meus pais Francisco Tavares Lopes e Vera de Oliveira Lopes, pelo incentivo nos estudos e todo apoio para continuar esta caminhada.

Gostaria de agradecer a todos os meus amigos que souberam levantar-me nos momentos mais difíceis desta caminhada.

Aos amigos Amauri Sant' Anna Ghisleri, Jeziel Torres Pereira e Wagner Godinho Valente, uma equipe de incentivos para a caminhada até este dia comigo.

Agradeço à Prof. Nádia Fátima de Oliveira, pelo apoio na correção dos meus erros de português e também na orientação relacionada com as questões de direito.

Pelo apoio incondicional na confecção do sistema, gostaria de agradecer à graduanda Rosemary Francisco.

Agradecer a Gilsiley Darú no apoio a a confecção dos programas em C++.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Siglas

$\mathbb{Z}$	Conjunto de todos os números inteiros.
$\mathbb{Z}_n$	Conjunto dos números inteiros módulo n.
$\mathbb{Z}_n^*$	Conjunto dos números inteiros módulo n que são relativamente primos a n.
$\Phi(n)$	Número de elementos ou ordem de $\mathbb{Z}_n^*$.
Alice	Entidade que envia uma mensagem.
Beto	Entidade que recebe uma mensagem.
Eva	Entidade que representa um intruso no sistema.
Pablo	Entidade que representa um intruso passivo.
ku	Chave pública.
kr	Chave privada.
RSA	Padrão de cifragem assimétrica.
MD5	Algoritmo que calcula o resumo de um arquivo digital qualquer.
SHA1	Algoritmo que calcula o resumo de um arquivo digital qualquer.
PDDE	Protocoladora Digital de Documentos Eletrônicos.
ICP	Infra-estrutura de Chave Pública.
AD	Autoridade de Datação.
SD	Serviço de Datação.
K	Chave de cifragem e decifragem simétrica.
$A \rightarrow B$	Fluxo de envio da mensagem.
AC	Autoridade de Certificação.
AR	Autoridade de Registro.

Resumo

Este trabalho propõe vários protocolos criptográficos para o envio de mensagens anônimas com a possibilidade da revelação da identidade do emissor num determinado período de tempo no futuro. Todos os protocolos são analisados em relação a uma lista de requisitos de segurança que os mesmos devem atender. Finalmente foi desenvolvido um sistema para a denúncia anônima segura na Web.

Palavras-Chave: Anonimato, Criptografia Temporal, Compartilhamento de Segredo.

Abstract

This dissertation proposes several cryptographic protocols for the sending of anonymous messages with the possibility of the revelation of the identity of the sender in a certain period of time in the future. All of the protocols are analyzed in relation to a list of safety requirements that they should attend to. Finally, a system was developed for safe anonymous accusation on the Web.

Key Words: Anonymity, Time-Release Cryptography, Secret Sharing.

Capítulo 1

Introdução

O mundo das comunicações digitais[1], nunca foi tão acessível aos usuários da Internet[2]. Com esta facilidade de acesso observa-se também, a necessidade de um mecanismo que possa garantir a segurança dos dados contra intrusos[3] na transmissão através da Internet.

A segurança da informação, desde os tempos mais antigos, seja no envio de instruções de batalhas na época do Império Romano, com Júlio César, ou para esconder o conhecimento egípcio, através de hieróglifos com escrita fora de padrão, ainda se apresenta como um problema real e atual da nossa sociedade segundo Bruce Schneier [SCH 01].

Segundo Simson Garfinkel [GAR 99] "*um computador é seguro se você pode ter certeza que ele e seu software vão se comportar da maneira como você espera*". Garantir este funcionamento correto é função dos algoritmos[4] e recursos matemáticos em sistemas de segurança.

A Internet, juntamente com estes algoritmos e recursos matemáticos, formam a base para garantir transações comerciais de diversas ordens.

Nota-se que, atualmente uma empresa não utiliza sua rede de comunica-

[1]Este "mundo"representa todos os meios de comunicação digital existentes.

[2]Também conhecida como rede mundial de computadores.

[3]Denominação atribuída a uma pessoa que captura dados na Internet para ler, modificar, incluir ou apagar.

[4]Tipo de escrita que demonstra uma seqüência de tarefas a serem seguidas.

ção de dados apenas para impressão de documentos. Existem diversas aplicações em que a comunicação pela Internet é inevitável, por exemplo, uma transação bancária para pagamento de uma GPS[5] eletrônica, acesso ao banco de dados da matriz, entrega da declaração do imposto de renda, compras *on-line*[6], através do cartão de crédito, o envio de documentos na forma digital com valor legal perante a lei, compras com dinheiro digital mantendo privacidade de identidade, participação de leilões *on-line*, entre outros.

As pessoas comunicam-se através de *e-mails*, *chats*[7], *webphone*[8] e outros. Utilizando estes programas, expõem seus sentimentos, suas emoções, sua posição político-partidária, sua preferência sexual, que talvez, no dia-a-dia não sentiriam-se à vontade para expor.

Aliado ao crescimento acentuado das tecnologias e à grande utilização de computadores, a privacidade das pessoas continua cada vez mais sendo invadida. As empresas armazenam informações sobre os hábitos de navegação, consumo e preferências dos usuários da Internet. Estas informações podem ser utilizadas tanto para beneficiar como para prejudicar os usuários.

Empresas também podem utilizar estas informações com a intenção de forjar transações comerciais com os clientes, através do uso indevido de cartão de crédito, por exemplo. Inúmeros usuários podem ser prejudicados.

Em muitos casos de troca de informações através da Internet, a questão do anonimato deve ser considerada. Um usuário que faz uma compra através da Internet com dinheiro digital, por exemplo, deveria permanecer anônimo perante o comerciante e o banco, da mesma forma, que se é anônimo quando se compra um produto usando papel-moeda em um supermercado qualquer.

Em uma eleição através da Internet, no momento da entrega do voto, a autoridade de votação deve reconhecer que recebeu um voto, porém, não deve ter meios para associar este voto ao votante.

---

[5]Guia da Previdência Social, que agora pode ser paga via Internet.

[6]Termo que indica a constante conexão à Internet.

[7]Programa utilizado no computador conectado à Internet, que possibilita a troca de mensagens *on-line*, através do teclado.

[8]Programa de computador que conectado à Internet, possibilita a conversa através da fala.

Grandes empresas estatais adquirem produtos e serviços através de licitação pública. Um dos requisitos necessários para este processo, é a imparcialidade no momento da tomada de decisão por adquiri-los através de um fornecedor A ou B qualquer. O anonimato neste caso pode ajudar a garantir esta imparcialidade, pois, no momento da entrega do envelope digital contendo a proposta, faz-se esta entrega ser anônima, com recuperação de identidade apenas após a confirmação do resultado final.

Outra aplicação do anonimato para garantia de imparcialidade é o leilão de objetos através da Internet. Neste caso, o anonimato está presente no momento dos lances, até que seja definido o vencedor de cada lote.

Para resolver os problemas relacionados com o anonimato, observa-se que diversos trabalhos foram produzidos com este intuito.

1.1 O Contexto das Comunicações Anônimas na Internet

A medida que cresce a grande ”Rodovia da Informação”[9], os usuários da Internet aumentam as trocas de dados e mensagens nas comunicações pessoais e de negócios. Isto cria a necessidade de se ter meios eletrônicos seguros que permitam tais comunicações, como o *chat*, uma votação, os pagamentos ou comércio eletrônico.

Soluções que promovem comunicação anônima no mundo eletrônico apresentam-se como solução necessária para estes casos. Um exemplo simples são os cabeçalhos de e-mail, contendo o endereço de e-mail do remetente e ou do receptor. Também são requeridos endereços de rede para estabelecer uma comunicação entre dois computadores. Normalmente, estes endereços de rede já dão informações relevantes sobre identidades do remetente e receptor.

Por outro lado, oferecer uma solução técnica para comunicação anônima pode ter também desvantagens. Em algumas aplicações, o anonimato pode ser usado por criminosos ou pessoas com más intenções. Exemplos são *spam*[10] de e-mail e lavagem de

[9] Esta expressão é atribuída a grandes quantidades de cabos, na qual trafegam os dados e mensagens.

[10] Esta é a denominação usada para os e-mails não solicitados, e que enchem a caixa postal de mensagens inúteis.

dinheiro[11]. Em muitas situações é necessário a revogação do anonimato a pedido de um juiz ou outra autoridade legal.

Assim como nos exemplos acima citados, outras aplicações necessitam de anonimato para comunicação, como um sistema de denúncia anônima ou entrega de uma proposta em uma concorrência pública, garantindo a imparcialidade no ato da entrega da proposta, através da Internet.

Existem hoje diversos produtos que proporcionam comunicações anônimas, por exemplo, o *Crowds* da AT&T, *Onion Routing* da US Navy, o *Freedom* da Zero-Knowledge Systems, o *Anonymizer* da Anonymizer Inc. e o *ProxyMate* da LPWA Bell-labs.

Este documento pretende discutir soluções e técnicas para prover comunicação anônima. Conforme será visto no decorrer deste documento, prover comunicação anônima na Internet não é tão trivial quanto se apresenta, pois a maior parte das aplicações revelam informações sobre a identidade do usuário.

Será utilizado como demonstração prática o contexto de denúncias anônimas através da Internet, que devem garantir os plenos direitos constitucionais e penais com base nas leis brasileiras.

1.2 Objetivos

1.2.1 Objetivo Geral

Propor um protocolo criptográfico para produzir uma mensagem com ocultação da identidade do emissor, sendo que qualquer subgrupo de um grupo de entidades autorizadas, poderá revelar a identidade deste emissor em um período de tempo específico no futuro, $t_1 \leq t \leq t_2$, onde t_1 é o início e t_2 o final do período. Caso a identidade não seja revelada neste período, esta identidade deverá ser destruída após t_2.

[11] Esta expressão é utilizada, quando o dinheiro proveniente de transações ilícitas, é colocado novamente em circulação no mercado legal.

1.2.2 Objetivos Específicos

São os objetivos específicos:

- Estudar como são utilizados os métodos de autenticação existentes hoje;
- Levantar as questões jurídicas que envolvem as comunicações anônimas;
- Diferenciar anonimato de privacidade, e como estes se correlacionam;
- Estudar aplicações que necessitam invariavelmente de comunicações anônimas;
- Estudar protocolos criptográficos que garantem comunicação anônima;
- Verificar como deve ocorrer uma comunicação entre um grupo fechado numa rede de comunicação de dados e, mesmo assim, garantir o anonimato das mensagens;
- Propor um protocolo criptográfico que atenda ao objetivo geral.
- Listar os requisitos de segurança necessários para que haja uma comunicação anônima com revelação da identidade no futuro;
- Implementar uma aplicação que se baseia em um dos protocolos propostos;

1.3 Materiais e Métodos

Para desenvolver este trabalho, utilizou-se a metodologia de primeiro identificar claramente o problema a qual se propõe resolver. Com a definição clara do que resolver, levanta-se uma série de requisitos que a solução deveria abranger para a solução do problema. Propõe-se alguns protocolos que visam resolver o problema.

Com o intuito de atingir melhor compreensão prática do comportamento do protocolo, foi implementada uma aplicação para denúncia anônima segura que se utiliza de um dos protocolos propostos.

Utilizara-se diversas ferramentas comerciais que auxiliaram na conclusão do trabalho, como livros, artigos, revistas e aplicativos, conforme descrito a seguir.

A pesquisa em livros clássicos da área como os de Bruce Schineier [SCH 96], Douglas Stinson [STI 95], Willian Stallings [STA 98] e A. Menezes [MEN 96], trouxeram a base teórica em segurança e criptografia para o entendimento dos conceitos trabalhados. Esta pesquisa deu-se através de leitura e resolução de exercícios dos livros.

A busca da atualidade do tema do trabalho deu-se através de revistas especializadas e artigos relacionados à segurança da informação.

1.4 Motivações para Tema

Uma das questões da humanidade é que toda pessoa tenha direito de expressão garantida, porém, segundo a nossa Constituição Federal [BRA 88] esta expressão não deve ser anônima.

Com o crescimento acentuado das comunicações de dados através da Internet, aumentam os casos de fraudes por pessoas mal intencionadas.

Mesmo que utilizemos técnicas para cifragem de dados e mensagens antes do envio à Internet, ainda assim, não se estará imune ao recolhimento de informações a nosso respeito. Para continuar incógnitos na Internet, devemos aplicar técnicas que garantem o anonimato das transmissões.

Em algumas aplicações, o anonimato é fundamental para garantia de imparcialidade, como é o caso de uma votação digital, em que deve existir segredo em relação ao seu conteúdo no momento da entrega do voto eletrônico, tornando-o anônimo após ser colocado na urna eletrônica.

Não faz muito tempo, na guerra de Kosovo, pessoas enviavam mensagens anônimas para o mundo, informavam como estava a situação naquele país, de maneira que poderiam se comunicar sem que suas identidades fossem reveladas, criando uma proteção para os informantes.

O que podemos notar é que a questão é mais abrangente do que parece. Como pode uma pessoa efetuar a denúncia de um crime ou fraude financeira, e mesmo assim ter o seu direito de não identificação garantido?

Convivemos com dois extremos na Internet hoje; um em que tudo sobre

nós e nossos hábitos são conhecidos, ou seja, temos uma identificação garantida, e outra onde pode-se navegar sem ser rastreado de qualquer forma caracterizados pelo anonimato absoluto, conforme a figura 1.1.

O que hoje se pretende dentro da Internet é uma navegação segura e privacidade em suas comunicações. Podemos notar que esta é uma questão difícil de resolver. Existe um meio termo em que estas duas situações convivem constantemente, onde o que mais se leva em conta é a confiança entre as partes.

Figura 1.1: Identificação X Anonimato. A figura ilustra um meio comum na questão existente entre o anonimato absoluto e a identificação garantida. Um lugar onde o que prevalece é a relação de confiança mútua entre as partes que estão comunicando entre si.

O tema do trabalho enfoca uma determinada aplicação de denúncia, pois os sistemas que se propõem a este fim não garantem o anonimato denunciante, apenas a sua privacidade. O Tribunal Regional Eleitoral de Roraima implantou o "*Sistema de Denúncia On Line e Disque Denúncia*"visando atender ao eleitor. O eleitor não necessita identificar-se ao fazer a denúncia. A Polícia Civil do Estado do Espírito Santo, também oferece um sistema de cadastramento de denúncia on-line. A página da Polícia Civil do Estado do Espírito Santo atenta ao fato que a falsa comunicação à Polícia é crime, mas na mesma página coloca que "*seu anonimato será preservado*". Esta afirmação se coloca contra a citação da Constituição Brasileira [BRA 88].

O Sistema proposto neste trabalho prevê que o usuário deve ser identificado ao fazer a denúncia, porém sua identificação deve permanecer em sigilo até um certo período no futuro.

O fato do usuário necessitar ser identificado deve-se ao fato da garantia de que ninguém e nenhum sistema poderão efetivar a denúncia sem a autorização do usuário.

1.5 Organização do Trabalho

Este trabalho está dividido em doze capítulos, procurou-se abordar os temas de cada capítulo de forma gradativa para o melhor entendimento.

No capítulo 1 introduz-se o tema e apresentam-se as motivações para o seu desenvolvimento.

No capítulo 2 apresenta-se a caracterização do problema. Também são feitas considerações comparativas com documento papel.

Uma breve revisão dos princípios de criptografia é apresentada no capítulo 3, visando fornecer a base de estudo para o restante do trabalho.

O capítulo 4 apresenta os métodos utilizados para a garantia de autenticação de um indivíduo ou entidade perante aos sistema computacionais.

O anonimato, diversas maneiras de defini-lo e a relação de técnicas que o garantem nas comunicações está colocado no capítulo 5.

Um estudo sobre criptografia temporal foi abordado no capítulo 6, visando ao seu entendimento e aplicações.

No capítulo 7, colocou-se um estudo sobre compartilhamento de segredo como técnica para dividir o controle de um determinado segredo.

A comunicação em grupo será abordada no capítulo 8, com o intuito de apresentar os conceitos e propriedades.

No capítulo 9 será apresentada a elaboração do protocolo que atende ao objetivo do trabalho, também apresentada-se a sua construção através de cinco protocolos criptográficos, os requisitos de segurança e suas análises comparativas entre todos.

No capítulo 10, apresentam-se algumas características e métodos para se fazer uma denúncia anônima. Apresentam-se também citações de leis e pareceres dados com relação ao tema de anonimato.

No capítulo 11, detalha-se o funcionamento da aplicação que está baseado no terceiro protocolo proposto.

Por fim, no capítulo 12 apresentam-se as considerações ao trabalho, sua aplicabilidade, contribuições e sugestão de trabalhos futuros.

Capítulo 2

Definição do Problema

2.1 Introdução

Um dos pilares da resistência na utilização da comunicação através de meios eletrônicos ainda é a dificuldade de transpor a realidade das comunicações efetuadas através de um documento em papel[1] aos padrões de funcionamento das comunicações por meios eletrônicos[2].

Neste capítulo apresenta-se a caracterização de uma situação cujo propósito é solucionar através de um protocolo criptográfico.

A seção 2.2 apresenta como é a característica do problema no mundo real, ou seja, uma comunicação através de documento em papel.

A seção 2.3 apresenta o problema do ponto de vista do mundo digital.

2.2 Caracterização com Documento em Papel

Suponha-se que, com documento em papel, uma pessoa deseje enviar uma correspondência anônima e que no futuro, o receptor da correspondência possa obter a identidade do emissor. Uma solução para este problema é esconder (encobrir) a parte

[1] A expressão documento em papel, indica as comunicações que são feitas utilizando papel como meio de armazenar a informação a ser comunicada.

[2] Maneira pela qual a informação é armazenada e transmitida ao destinatário.

da identidade da correspondência. Se, no futuro, o receptor quiser conhecer a identidade do emissor, basta descobrir a parte da identidade da correspondência.

Uma correspondência em papel normalmente é formada por duas partes. A parte do conteúdo da mensagem que contém a informação propriamente dita. A parte da identidade que contém assinatura do emissor da correspondência a sua identidade. Na figura 2.1 retirada do trabalho da Adriana Elissa Notoya [NOT 02], vê-se um modelo de correspondência de documento em papel, constando na parte da mensagem e na parte da identidade de quem enviou a correspondência.

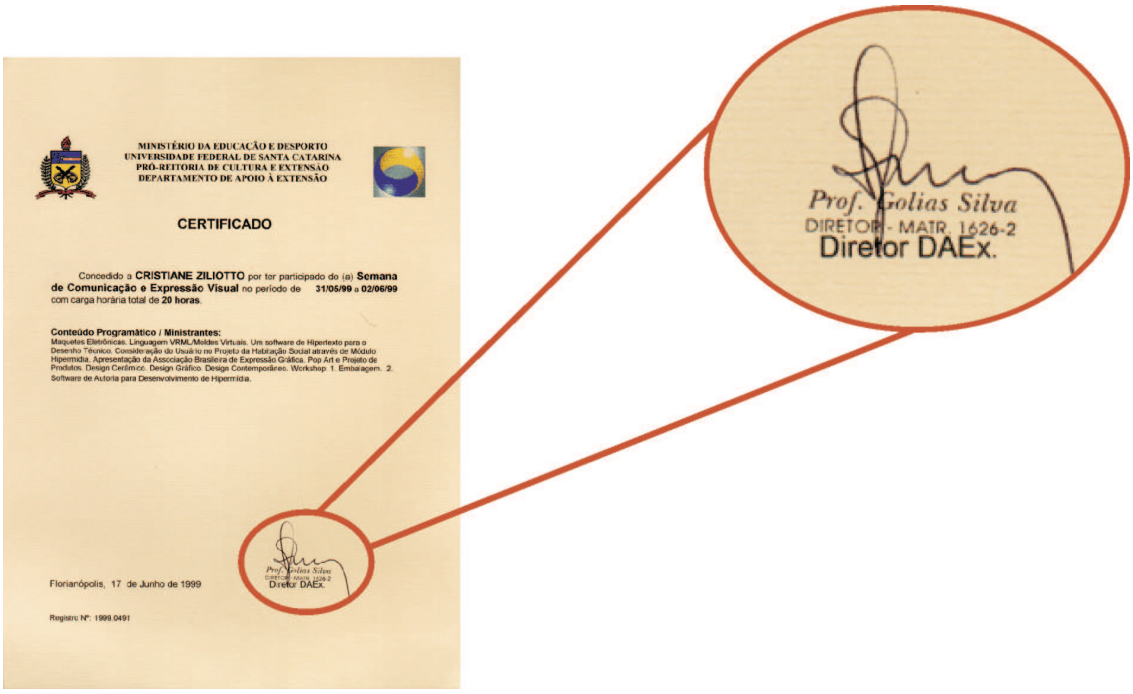

Figura 2.1: Documento em Papel: Exemplo de um documento em papel assinado manualmente. Conforme pode-se notar, existe uma divisão da informação com a identidade do autor da mensagem.

2.3 Caracterização no Documento Eletrônico

De forma semelhante ao documento em papel, as correspondências com documentos eletrônicos podem apresentar uma divisão em duas partes, a parte do conteúdo

da mensagem na sua forma eletrônica representado com codificação binária de 0 e 1, e a parte da identidade que apresenta a assinatura também na sua representação binária. Na figura 2.2 [NOT 02] apresenta-se a analogia do documento em papel para o documento eletrônico, destacando que neste caso a assinatura no formato digital também é uma seqüência de bits 0 e 1.

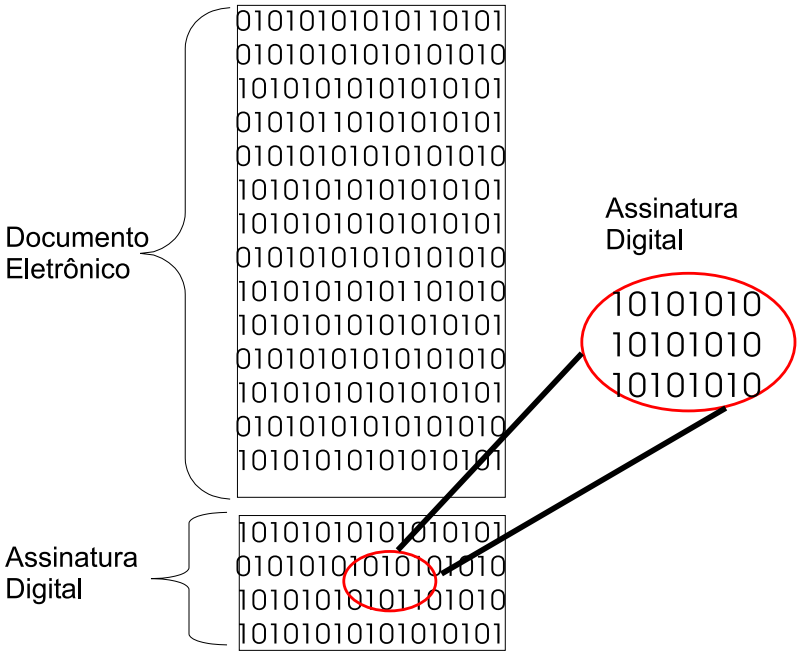

Figura 2.2: Documento Eletrônico: Apesar de formada por códigos binários, pode existir a separação do conteúdo da mensagem com a identidade do autor assinante da mensagem.

2.4 Protocolo com Documento em Papel

Imagine-se a situação de uma pessoa que necessita enviar uma mensagem para uma outra com as seguintes considerações:

- A mensagem deve conter a identificação de quem a escreveu;
- A identificação do escritor deve ser ocultada;

- A mensagem deve ser enviada e recebida sem revelar a identidade do emissor e do receptor;
- O receptor deve ter acesso à informação contida na mensagem, mesmo sem conhecer o autor;
- Deve ser possível a revelação do autor, caso necessário, num futuro determinado;
- Após o período de possível revelação do autor, a identificação deve ser perdida.

Estas condições caracterizam um problema a ser resolvido tanto com documento em papel como documento eletrônico. Técnicas podem ser aplicadas na tentativa de solucionar este problema. Dentre estas técnicas, existem várias que possuem analogia entre estas duas situações. A representação simples do envio de uma mensagem anônima com possível identificação numa janela de tempo futura é apresentada na figura 2.3 para melhor ilustrar.

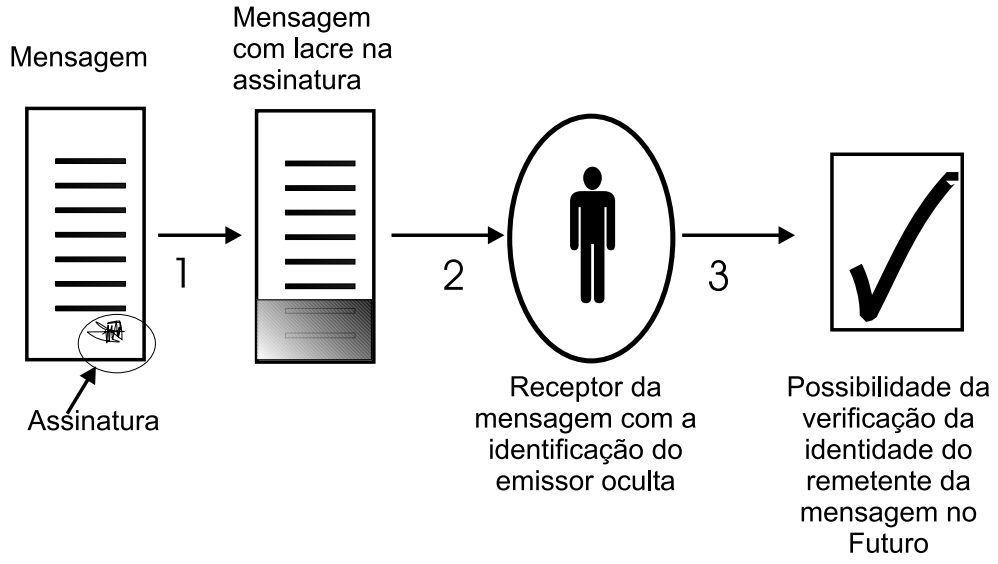

Figura 2.3: Envio de carta anônima com documento em papel: 1. Uma Mensagem foi escrita, assinada, e necessita ser enviada com identidade oculta; 2. Foi encoberto a assinatura da mensagem de forma a ocultar a identidade do emissor e enviada para o receptor; 3. Decorrido o período inicial de ocultação, o receptor tem a possibilidade de identificar o assinante da mensagem, dentro de uma janela de tempo no futuro.

2.5 Temporização do Processo

Para um melhor entendimento do processo no decorrer do tempo, a figura 2.4 demonstra a linha do tempo com marcações para análise dos acontecimentos. Observa-se nesta figura que existe um intervalo bem definido de ocultamento da identidade, um período (ou janela) para possível revelação da identidade e o período final onde a mensagem deve permanecer anônima.

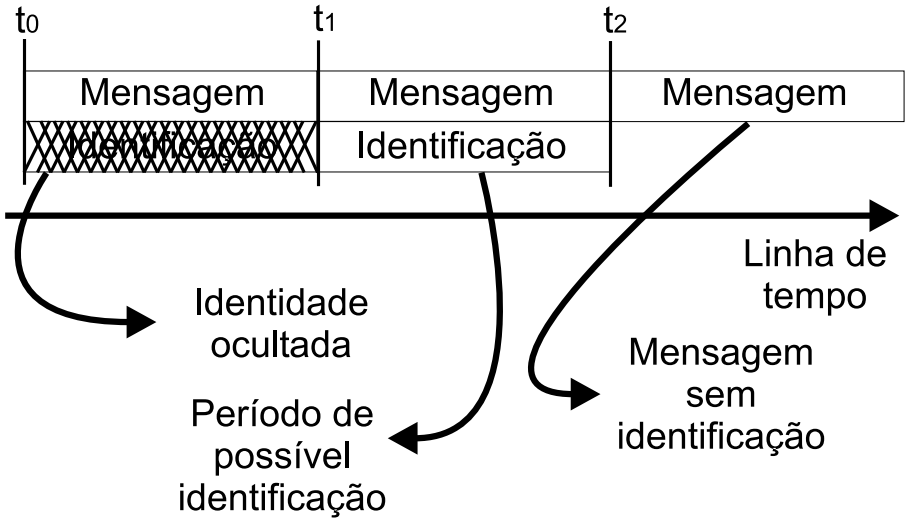

Figura 2.4: Linha de tempo: A marcação de ponto t_0 indica o momento de envio da mensagem com identidade oculta. Entre t_0 e t_1, prevalece a ocultação da identidade. O intervalo entre t_1 e t_2, é a janela ou período onde pode-se ou não revelar a identidade do emissor. Após t_2 a mensagem deve permanecer anônima, e a identidade do autor deve ser destruída.

Uma solução para o anonimato a partir de t_2 seria, por exemplo, a queima da parte do documento que contém a assinatura do autor. Isso com documento em papel. A questão é como fazê-lo no caso de documento eletrônico.

Tanto no formato digital como no formato papel, a composição do documento pode ser separada em mensagem, identificação e lacre após o seu envio. A mensagem corresponde à informação propriamente dita, a identificação utiliza-se dos métodos apresentados no capítulo 4 para identificar corretamente o autor e o lacre que pode representar desde uma tinta a ser passada em cima da assinatura no papel, como uma técnica

de criptografia para guardar informações apresentada no capítulo 3.

2.6 Aplicações para o Tema

O problema que foi apresentado visa resolver situações onde o usuário apesar de se identificar ao sistema, não precise necessáriamente exposto está identidade aos participantes, a menos que haja uma situação especial para isto.

Pode-se apresentar as seguintes situações em que o tema pode ajudar na sua resolução:

Denúncia Anônima O fato de um usuário poder efetuar um relato de um ocorrido sem que tenha a sua identidade tornada pública, este relato pode ser de um ato criminal;

Pesquisa de Satisfação de um Funcionários Cada funcionário tem o direito de fazer uma avaliação clara segura, sabendo que a sua identidade não será revelada ao público;

Avaliação de um Curso Ao final de um curso, faz-se uma avaliação onde o instrutor não necessáriamente pode identificar o seu avaliador.

2.7 Conclusão

Este capítulo teve a finalidade de apresentar a caracterização do problema e as situações a qual se propõem resolver. O problema com documento em papel foi apresentado no intuito de facilitar o entendimento. A caracterização no tempo se mostra como uma variável a ser resolvida.

Capítulo 3

Princípios da Criptografia

3.1 Introdução

Neste capítulo são apresentados conceitos e terminologias para o estudo da criptografia e suas técnicas. O conteúdo deste capítulo servirá de base para os outros, onde serão discutidas as questões relacionadas à autenticação e ao anonimato.

3.2 Princípios

A palavra criptografia pode ser definida como a arte e ciência de manter uma mensagem segura de intrusos segundo, Bruce Schneier [SCH 96]. No momento em que uma entidade emite um texto aberto[1], esta mensagem é submetida a um processo criptográfico, gerando um texto cifrado[2], este texto cifrado é, então, enviado para uma entidade receptora. Esta, por sua vez, submete-o a um processo criptográfico que retorna o texto à sua forma original.

Em um sistema criptográfico, pode-se definir alguns personagens usados como padrão para melhor ilustrar o que está sendo apresentado. Neste trabalho definiu-se que Alice é sempre uma entidade emissora de mensagens, Beto será sempre uma entidade receptora de mensagens enviadas por Alice e Eva é uma terceira pessoa que

[1] Mensagem legível.

[2] Mensagem incompreensível naturalmente.

faz o papel de intruso. A figura 3.1 mostra um sistema criptográfico.

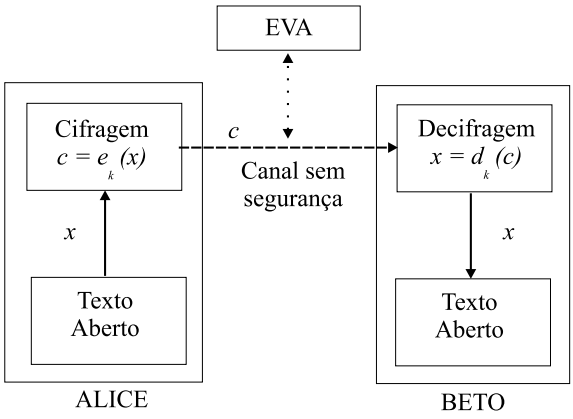

Figura 3.1: Crifrar e decifrar. Alice produz um texto aberto x, submete-o a um processo criptográfico e, produzindo assim o texto cifrado c. O texto cifrado c é enviado para Beto. Para Beto decifrar o texto c, basta fazer o processo inverso, submetendo-o ao processo criptográfico d, obtendo o texto aberto x. Eva representa um intruso que fica observando o canal de comunicação que não tem recursos de segurança de dados.

Segundo Bruce Schneier e William Stallings [SCH 96, STA 98], um sistema, para garantir que a informação trocada seja segura, deve fornecer os seguintes serviços de segurança:

1. **Confidencialidade:** Deve garantir que a informação seja secreta para pessoas não autorizadas;

2. **Autenticação:** Deve garantir com certeza que, quem enviou e quem recebeu a mensagem eram o emissor e receptor corretos;

3. **Integridade:** Deve garantir que a mensagem não foi alterada durante a sua transmissão, ou seja, uma terceira pessoa, intrusa, não pode ter condições de obter ou alterar a mensagem original;

4. **Não-Repúdio:** Deve garantir que o emissor não possa negar que a mensagem é de sua autoria, depois desta ser recebida pelo receptor e vice-versa;

5. **Controle de acesso:** Habilidade de limitar e controlar o acesso de sistemas e aplicações no canal de comunicação;

6. **Disponibilidade:** Garantir disponibilidade do sistema mediante vários ataques.

No exemplo citado, foi utilizada a expressão processo criptográfico tanto para transformar a mensagem em texto cifrado, quanto para retornar ao texto original. Estes dois processos são chamados respectivamente de cifrar e decifrar. Estes processos acontecem através de um algoritmo criptográfico. Quando este algoritmo contém o segredo é chamado de **restrito**, entretanto essa abordagem não é mais utilizada em larga escala, como coloca Bruce Schneier [SCH 96].

Hoje, nos processos modernos, o segredo não está mais no algoritmo, seguem o princípio de Dutchman A. Kerckhoff, onde se prega que a segurança deve estar nas chaves envolvidas nos processos, e não no algoritmo em si, como apresentado por Douglas Stinson [STI 95]. O algoritmo criptográfico acima referido é chamado de cifrador, segundo Bruce Schneier [SCH 96], e é usada uma função matemática, geralmente uma para cifrar e uma para decifrar.

De acordo com Douglas Stinson [STI 95] um criptossistema[3] é composto por cinco-tuplas $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, onde quatro condições devem ser satisfeitas:

1. $\mathcal{P}$ é um conjunto finito de possibilidades de textos abertos;

2. $\mathcal{C}$ é um conjunto finito de possibilidades de textos cifrados;

3. $\mathcal{K}$, o espaço de chaves, é um conjunto finito de possibilidades de chaves;

4. Para cada $k \in \mathcal{K}$, existe uma regra de cifragem;

$$e_k : \mathcal{P} \rightarrow \mathcal{C} \tag{3.1}$$

e

$$d_k : \mathcal{C} \rightarrow \mathcal{P} \tag{3.2}$$

são funções semelhantes que $d_k(e_k(x)) = x$ para todo texto aberto $x \in \mathcal{P}$.

A maneira como é usada a chave, determina o tipo de algoritmo que será usado, podendo ser *algoritmo simétrico ou assimétrico*.

[3]Segundo Bruce Schneier [SCH 96], um criptossistema é formado por um algoritmo, mais todas as possibilidades de textos abertos, textos cifrados e chaves.

3.3 Criptografia Simétrica

A criptografia de chave simétrica ou convencional utiliza a mesma chave para cifrar e decifrar uma mensagem. Pode-se então, dizer que a segurança do algoritmo simétrico está na chave utilizada. Caso esta seja descoberta, perde-se a segurança, pois, pode-se cifrar e decifrar a qualquer momento a mensagem. A figura 3.2 mostra o processo de cifrar e decifrar para o caso no qual Alice e Beto compartilham a mesma chave secreta k. Como a segurança da criptografia simétrica está na chave utilizada para cifrar, deve existir um canal seguro para a transmissão desta chave k para Beto.

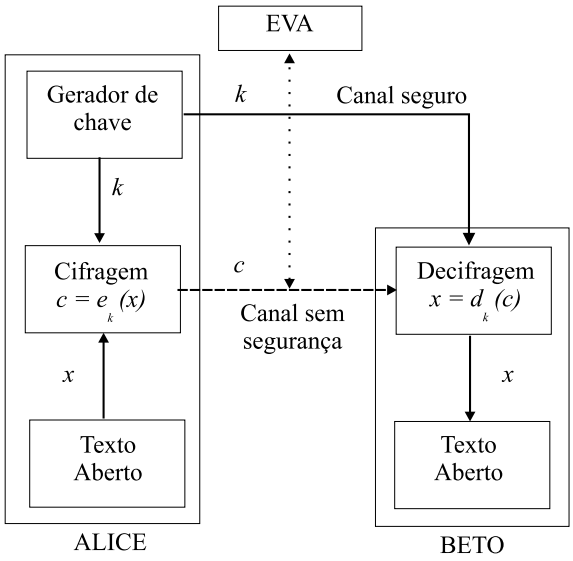

Figura 3.2: Criptografia Simétrica: Alice produz um texto aberto x, passa-o através de um algoritmo de cifragem e que utiliza uma chave secreta k. A mensagem cifrada c é enviada para Beto através de um meio de comunicação qualquer sem segurança. Note-se que a chave k também é enviada para Beto, porém, ela segue um caminho seguro, para garantir que somente Beto tenha acesso a ela. Quando Beto recebe o texto cifrado c, ele o decifra utilizando a chave secreta k enviada a ele por Alice, recuperando o texto aberto x. Eva é o intruso que fica observando a comunicação, porém não consegue compreender a mensagem que está sendo enviada, pois não possui a chave k para isto.

Analisando a figura 3.2, observa-se que os algoritmos de cifrar e decifrar são basicamente os mesmos, mudando apenas a forma como é utilizada a chave secreta k [STA 98].

3.4 Criptografia Assimétrica

Algoritmos assimétricos ou de chave pública utilizam duas chaves distintas, uma chave privada (secreta) e uma pública. Estas chaves possuem um relacionamento entre si, pois o que uma cifra, só a outra decifra. Com a criptografia assimétrica pode-se garantir a autenticação de quem enviou a mensagem, e também a confidencialidade para quem recebe uma mensagem. Abaixo apresentam-se dois exemplos que ilustram estas propriedades:

1. Alice quer enviar uma mensagem confidencial para Beto, ou seja, que só ele possa ter acesso. Neste caso, Alice deve cifrar a mensagem com a chave pública de Beto, e enviar o texto cifrado para ele. Beto, quando receber a mensagem, utiliza a sua chave privada, para decifrar a mensagem. Como somente Beto conhece a sua chave privada, Alice tem a garantia de que a mensagem manteve-se confidencial. A figura 3.3 mostra uma situação de processo de confidencialidade.

2. No outro caso, Alice necessita provar a autenticidade da sua mensagem perante Beto. Então, Alice deve cifrar a mensagem, com a sua chave privada, e enviar para Beto o texto cifrado, junto com a sua chave pública. Observa-se que Beto agora tem como provar que foi Alice que enviou a mensagem, pois somente a chave pública de Alice, pode decifrar a sua mensagem, porém este processo não garante a privacidade da mensagem, pois qualquer que possuir a chave pública da Alice, poderá ler a mensagem. A figura 3.4, ilustra como ocorre este processo. Este é o princípio da assinatura digital.

Para solucionar o problema de autenticação e confidencialidade, pode-se combinar estas duas abordagens, criando um processo híbrido. A Alice primeiro cifra a mensagem com sua chave privada, em seguida cifra novamente com a chave pública de Beto. Neste caso híbrido, pode-se garantir que somente Beto possa verificar a mensagem, pois só ele possui a chave privada para fazer o processo inverso. O algoritmo de chave pública necessita que o proprietário da chave privada a mantenha sempre segura, pois senão perde-se a segurança do sistema.

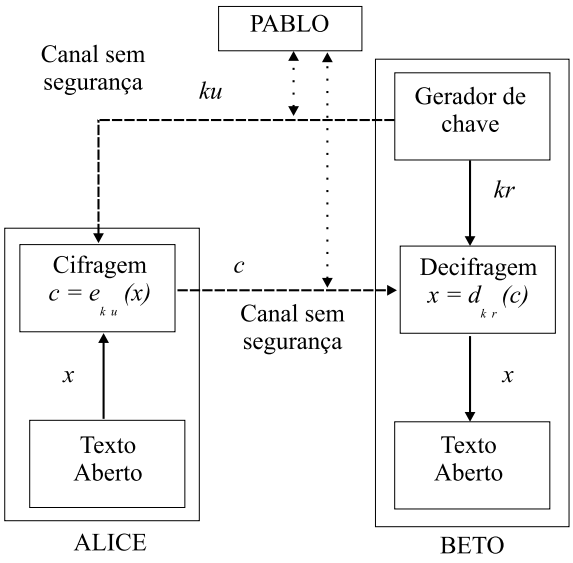

Figura 3.3: Cifragem com integridade da informação: Alice produz texto aberto x para enviar a Beto. Alice recebe de Beto a sua chave pública ku. Ela submete o texto aberto x a um algoritmo de criptografia assimétrica, utilizando a chave pública de Beto para cifrar, produzindo assim o texto cifrado c. O texto cifrado c é enviado a Beto através de um meio de comunicação qualquer. Beto, para ler a mensagem, utiliza sua chave privada kr junto com o algoritmo criptografia d e obtém o texto aberto x. Este processo utiliza duas chaves distintas. Pablo representa um intruso que tem acesso ao texto cifrado c e a chave pública ku, porém não consegue decifrar a mensagem para ler x.

3.5 Possíveis Ataques em uma Rede

Sempre que se conhece o oponente, como ele age, pode-se precaver contra ele. Em segurança de redes, existem alguns ataques que são previamente conhecidos. Apresenta-se a seguir, uma lista de ataques, segundo William Stallings [STA 98]:

1. **Divulgação**: Liberação do conteúdo de uma mensagem para qualquer pessoa ou processo, que não possua a chave criptográfica adequada;

2. **Análise de tráfego**: Retirada de uma amostra do tráfego entre dois pontos de uma comunicação;

3. **Mensagens mascaradas**: Inserção de mensagens dentro de uma rede, vindas de uma fonte fraudulenta;

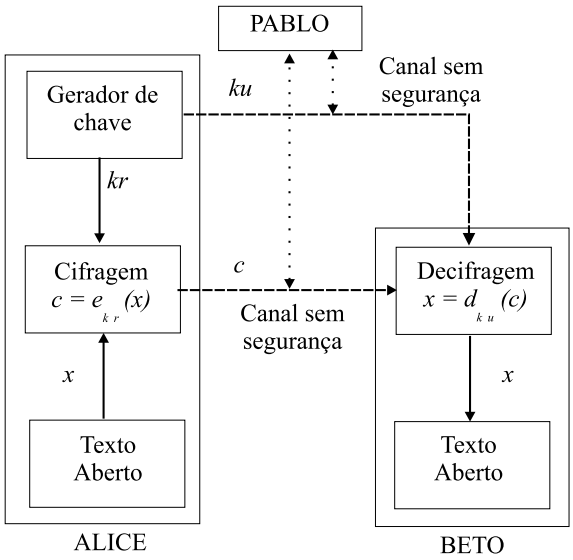

Figura 3.4: Autenticação de origem: Alice produz um texto aberto x e deseja enviar a Beto provando quem é. Alice submete o texto aberto x a um algoritmo de criptografia assimétrica e, que utiliza a chave privada kr de Alice para cifrar, produzindo assim o texto cifrado c. O texto é enviado a Beto. Beto, para ler a mensagem, utiliza a chave pública ku de Alice junto com algoritmo de criptografia assimétrica d e obtém o texto aberto x novamente. Pablo representa um intruso que tem acesso ao texto cifrado c e a chave pública ku, porém não consegue decifrar a mensagem para ler x. Este processo utiliza duas chaves distintas. É possível dizer que esta-se fazendo uma autenticação de origem, pois, como Alice utilizou sua chave privada kr para cifrar, somente a chave pública ku poderá decifrar a mensagem.

4. **Modificação de conteúdo**: Mudança no conteúdo das mensagens, estas mudanças são a inserção de partes, a exclusão do todo ou partes, a transposição, e a modificação;

5. **Modificação da seqüência**: Qualquer modificação de uma seqüência de mensagens entre as duas partes;

6. **Modificação da temporização**: Atraso ou reenvio de mensagens fora do tempo previsto para a transmissão da informação;

7. **Repúdio**: Negação de recebimento da mensagem pelo destinatário, ou negação de envio de mensagem pelo emissor.

Fazendo uma breve análise, pode-se dizer que, o item 1 e 2 da lista acima dizem respeito à confidência da mensagem. Os itens de 3 a 6 são resolvidos através de autenticação da mensagem e o item 7, através de assinatura digital, será resolvido.

3.6 Função Resumo

O termo *função resumo* ou *sumário da mensagem*, vem do inglês ”hash function”, ou também chamado de ”message digest”.

É uma função que recebe uma mensagem de qualquer tamanho, gerando uma saída de tamanho único, um resumo identificador único, que pode ser usado como comparação para a devida autenticação.

A função resumo tem a propriedade de ser uma função de ”mão única”, ou seja, depois de gerado o resumo, não é possível restabelecer a mensagem original a partir deste. Outra propriedade que está presente é a propriedade de espalhamento[4], que garante se a mensagem original foi alterada ou não. Pode-se sintetizar os requerimentos básicos para uma função resumo como segue, segundo William Stallings [STA 98]:

[4]O espalhamento é a propriedade de mudança que ocorre na saída, em uma mensagem cifrada, quando a mensagem aberta sofre uma alteração de qualquer bit ou bits.

1. **Tamanho da entrada**: Uma função resumo deve aceitar entradas de qualquer tamanho de blocos.

2. **Tamanho da saída**: O tamanho da saída desta função deve permanecer constante, independente do tamanho da entrada.

3. **Facilidade de execução**: O algoritmo deve ser relativamente fácil de executar, tanto em software, como em hardware.

4. **Computação inviável**: Deve ser, necessariamente, inviável a obtenção da mensagem aberta, partindo-se do resultado da função resumo, esta é a propriedade de função de mão única.

5. **Resistência à colisão fraca**: Dado um bloco x, deve ser computacionalmente inviável obter um $y \neq x$ com *h(y) = h(x)*.

6. **Resistência à colisão forte**: Deve ser computacionalmente inviável encontrar um par (x, y), tal que *h(x) = h(y)*.

Existem aplicações que se utilizam desta função, como um método de garantir a verificação de autenticidade dos dados por ela processados. A figura 3.5 a seguir apresenta um caso típico de aplicação desta função, a assinatura digital de um documento eletrônico.

Pode-se citar, como exemplos de algoritmos de função resumo:

- MD5 (*Message Digest*), desenvolvido por Ron Rivest no MIT, este algoritmo trabalha com saídas de 128 bits, a entrada é independente do tamanho, sendo sempre complementada para formar blocos de 512 bits;

- SHA (*Security Hash Algorithm*), desenvolvido pelo NIST e publicado como um padrão de processamento de informação federal (FIPS PUB 180) em 1993, revisado em 1995 passado a SHA-1. Trabalha com saídas de 160 bits e entradas variáveis complementadas para formar blocos de 512 bits;

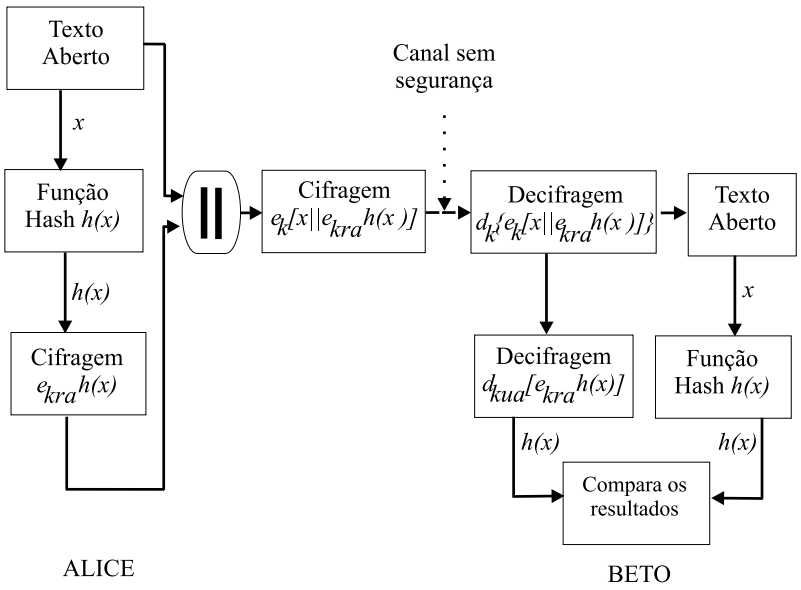

Figura 3.5: Aplicação de Função Resumo: Alice produz um texto aberto x, aplica uma função resumo h no texto x gerando um código único $h(x)$, este código é então cifrado com a chave privada de Alice kra. Faz-se a concatenação ($||$) do texto x e o código já cifrado $e_{kra}h(x)$, em seguida aplica-se um algoritmo simétrico e, com chave compartilhada k, obtendo $e_k[x||e_{kra}h(x)]$. Envia-se o resultado para Beto através de um canal sem segurança. Beto, ao receber esta mensagem cifrada, aplica o algoritmo simétrico com chave compartilhada d para decifrar, da seguinte forma $d_k\{e_k[x||e_{kra}h(x)]\}$. Beto separa o código cifrado $e_{kra}h(x)$ e decifra com a chave pública de Alice kua, obtendo a função resumo $h(x)$ novamente. No texto aberto x aplica-se a função resumo h para obter o resumo $h(x)$ para então comparar com o resumo $h(x)$ que foi obtido antes do envio, se forem iguais, a mensagem não foi alterada na comunicação. Nota-se que Alice fornece uma autenticação de origem, uma assinatura digital da mensagem e também confidencialidade.

- RIPEMD-160 (*RACE Integrity Primitives Evaluation*), este algoritmo também possui saída de 160 bits;
- HMAC (*Hash Message Authentication Code*), algoritmo baseado no uso de cifradores de bloco simétrico.

3.7 Assinatura Digital

Quando uma pessoa faz uma assinatura manuscrita em um documento, sabe-se que esta pertence à pessoa, porque leva intrínseca as características de aperto da caneta e continuidade na escrita da pessoa que o assinou, e também um cartório pode atestar a sua veracidade.

Nas comunicações eletrônicas digital, não se tem estas características para comparação, somente cadeias de bits. Assinar um documento digitalmente requer uma implementação que possa fornecer as propriedades equivalentes de uma assinatura manual. Estas propriedades segundo [STA 98] são:

- Capacidade de verificar o autor, a data e a validade da assinatura;
- Autenticar o conteúdo no prazo de validade da assinatura;
- A assinatura deve poder ser verificada por uma terceira pessoa (ou entidade), para resolver disputas, por exemplo, judiciais.

Segundo William Stallings [STA 98], para garantir que estas propriedades sejam satisfeitas, devem existir os seguintes quesitos para assinatura digital:

- A assinatura deve ser uma amostra de bits, que depende da mensagem que está sendo enviada;
- A assinatura deve usar alguma informação única do assinante, para prevenir contra falsificação e repúdio;
- Deve ser relativamente fácil a produção da assinatura digital;

- Deve ser relativamente fácil reconhecer e verificar a assinatura digital;
- Deve ser computacionalmente inviável falsificar a assinatura digital, em ambos os casos, de construção de uma nova mensagem para uma assinatura digital já existente, ou a geração de uma assinatura digital falsa, para uma dada mensagem;
- Deve ser prático, reter uma cópia da assinatura digital armazenada em meio digital;

A figura 3.5 é um exemplo de esquema de assinatura digital. Observa-se a necessidade de um esquema de assinatura digital consistindo de dois componentes básicos, conforme Douglas Stinson [STI 95]: *um algoritmo de assinatura e um algoritmo de verificação*. Uma descrição formal deste esquema é apresentada a seguir.

Um esquema de assinatura digital é formado por um conjunto de cinco tuplas (P, A, K, S, V), onde as seguintes condições são satisfeitas:

1. P é um conjunto finito de possibilidades de mensagens.
2. A é um conjunto finito de possibilidades de assinaturas.
3. K, o espaço de chaves, sendo um conjunto finito de possibilidades de chaves.
4. Para cada $k \in K$, existe um algoritmo $sig_k \in S$ e um correspondente algoritmo de verificação $ver_k \in V$. Cada $sig_k : P \rightarrow A$ e $ver_k : P \times A \rightarrow \{verdadeiro, falso\}$ são funções tais, que a seguinte equação é satisfeita para toda mensagem $x \in P$ e para toda assinatura $y \in A$:

$$ver(x, y) = \begin{cases} \text{verdadeiro} & \text{se} \quad y = sig(x) \\ \text{falso} & \text{se} \quad y \neq sig(x) \end{cases}$$

Segundo Douglas Stinson [STI 95] pode-se usar um criptossistema de chave pública RSA[5] [RIV 78], por exemplo, para prover assinatura digital, conforme o seguinte esquema.

[5] A primeira realização de um sistema de chave pública foi em 1978 por Rivest, Shamir e Adleman. Este sistema é baseado na dificuldade de fatoração de números inteiros grandes.

Pega-se $n = pq$, onde p e q são números primos. Pega-se $P = A = \mathbb{Z}_n$, e define-se

$$K = (n, p, q, a, b) : n = pq, \quad p, \quad q \ \ primos, \quad ab \equiv 1 \pmod{\phi(n)}.$$

Os valores de n e b são públicos, e os valores de p, q, a são secretos. Para $k = (n, p, q, a, b)$, define-se

$$sig_k(x) = x^a \mod n$$

e

$$ver_k(x, y) = verdadeiro \Leftrightarrow x \equiv y^b \mod n$$

$$(x, y \in \mathbb{Z}_n).$$

3.8 Problema do Logaritmo Discreto

A segurança de alguns criptossistemas baseia-se na dificuldade computacional do Problema de Logaritmo Discreto, que pode ser descrito como a dificuldade de se determinar a, mesmo tendo o conhecimento de y, α e p na equação abaixo:

$$y = \alpha^a \mod p \tag{3.3}$$

Sendo que p é um número primo longo (*geralmente com 512 bits*) e os números inteiros α e t, onde $0 < \alpha$ e $t < p$. Usa-se então esta dificuldade como um fator de segurança num esquema criptográfico.

3.8.1 Dificuldade Computacional

A dificuldade computacional vem da análise da complexidade do algoritmo segundo Routo Terada [TER 00]. Por exemplo:

1. Quando se aplicar uma quantidade de bits de dados de entrada n a um algoritmo qualquer A, este algoritmo será de *tempo polinomial* se a função $f(n)$ do tempo de execução, no pior caso, do algoritmo A, for $f(n) = O(n^k)$, onde k é uma constante.

2. Em outro caso, um algoritmo A, também com entrada n e função de tempo $f(n) = O(n^k)$, terá um *tempo exponencial*, se não for possível determinar k como um valor constante.

Em termos computacionais, um algoritmo de tempo polinomial é *computacionalmente eficiente (viável)*, e os de tempo exponencial são *computacionalmente ineficientes (difíceis).*

3.9 Algoritmo de ElGamal

Segundo Bruce Schneier [SCH 96] e T. ElGamal [ELG 85], o algoritmo de ElGamal tanto pode ser usado para assinatura digital como para cifragem de mensagens. A sua segurança é dada, conforme a seção anterior, na dificuldade de calcular o logaritmo discreto.

Para a geração de um par de chaves, primeiro escolhe-se um número primo, p, e dois números randômicos, α e a, tal que ambos, α e a, sejam menores que p. Então calcula-se:

$$y = \alpha^a \mod p \tag{3.4}$$

A chave pública obtida é composta por y, α, e p. Ambos α e p podem ser compartilhados com um grupo de usuários. A chave privada é a.

3.9.1 Cifras com ElGamal

Conforme visto, o algoritmo de ElGamal pode ser aplicado na cifragem de mensagens que serão enviadas. A tabela 3.1 [STI 95, SCH 96, ELG 85] apresenta de maneira resumida, os passos e os requisitos necessários para o envio de mensagens utilizando o algoritmo de ElGamal.

Apresenta-se a seguir um exemplo para ilustrar melhor a aplicação do algoritmo de ElGamal para cifrar mensagens.

Tabela 3.1: Cifrando com ElGamal

Chave Pública	
p	número primo (pode ser compartilhado com um grupo de usuários)
α	$< p$ (pode ser compartilhado com um grupo de usuários), sendo base geradora de p
β	$= \alpha^a \mod p$
Chave Privada	
a	$< p$, tal que $1 \leq a \leq (p-2)$
Cifragem	
k	aleatório escolhido, relativamente primo a $p-1$
y_1 (texto cifrado)	$= \alpha^k \mod p$
y_2 (texto cifrado)	$= x\beta^k \mod p$
Decifragem	
x (texto aberto)	$= y_2/y_1^a \mod p$

3.9.2 Exemplo de Cifra com ElGamal

Suponha-se que Alice necessita enviar uma mensagem para Beto e deseja utilizar, para isto, o algoritmo de ElGamal.

Conforme a tabela 3.1, escolhemos $p = 13$, $\alpha = 6$, e a chave privada $a = 9$, calcula-se:

$$\beta = \alpha^a \, mod \, p = 2^9 \, mod \, 13 = 5$$

a chave pública é composta por $\beta = 5$, $\alpha = 6$, e $p = 13$. A mensagens que se quer enviar é $x = 6$, para isto, conforme a tabela, escolhe-se $k = 11$, e calcula-se o texto cifrado:

$$y_1 = \alpha^k \, mod \, p = 6^{11} \, mod \, 13 = 11$$

e

$$y_2 = x\beta^k \, mod \, p = 6 \times 5^{11} \, mod \, 13 = 9$$

o par formado por (y_1, y_2), ou seja, $(11, 9)$, é o texto cifrado.

Para decifrar o texto, utiliza-se a equação abaixo, juntamente com o algoritmo de Euclides estendido para encontrar o multiplicativo inverso, como segue:

$$x = y_2 / y_1^a \, mod \, p = 9/11^9 \, mod \, 13 = 6$$

3.9.3 Assinatura com ElGamal

Uma das aplicações mais usadas é a assinatura digital. O algoritmo de ElGamal também pode ser utilizado para este fim, conforme a tabela 3.2 [STI 95, SCH 96, ELG 85], vemos também os requerimentos necessários.

3.9.4 Exemplo de Assinatura com ElGamal

Suponha-se que Alice necessita agora, assinar uma mensagem para enviar a Beto, e deseja utilizar, para isto, o algoritmo de ElGamal.

Conforme a tabela 3.2, escolhemos $p = 13$, $\alpha = 7$, e a chave privada $a = 10$, calcula-se:

$$\beta = \alpha^a \, mod \, p = 7^{10} \, mod \, 13 = 4$$

Tabela 3.2: Assinatura com ElGamal

Chave Pública	
p	número primo (pode ser compartilhado com um grupo de usuários)
α	$< p$ (pode ser compartilhado com um grupo de usuários), sendo base geradora de p
β	$= \alpha^a \mod p$
Chave Privada	
a	$< p$, tal que $1 \leq a \leq (p-2)$
Assinatura	
k	aleatório escolhido, relativamente primo à $p-1$
y_1 (assinatura)	$= \alpha^k \mod p$
y_2 (assinatura)	tal que $x = (y_1 a + k y_2) \mod (p-1)$
Verificação	
aceita como válido se	$\beta^{y_1} y_1^{y_2} \mod p = \alpha^x \mod p$

a chave pública é composta por $\beta = 4$, $\alpha = 7$, e $p = 13$. A mensagem que se quer assinar é $x = 6$, para isto, conforme a tabela escolhe-se $k = 11$, e calculamos o texto cifrado:

$$y_1 = \alpha^k \, mod \, p = 7^{11} \, mod \, 13 = 2$$

e utilizando o algoritmo de Euclides estendido para encontrar y_2 tal que:

$$x = (y_1 a + k y_2) \;\; mod \;\; (p-1)$$

$$6 = (2 \times 10 + 11 y_2) \, mod \, (12)$$

$$y_2 = 2$$

o par formado por (y_1, y_2), ou seja, $(2, 2)$, é a assinatura.

Para verificar a assinatura do texto, utiliza-se a equação abaixo,

$$\beta^{y_1} y_1^{y_2} \, mod \, p = \alpha^x mod \, p$$

$$4^2 2^2 \, mod \, 13 = 7^6 mod \, 13$$

$$12 = 12$$

que confirma a assinatura do texto.

3.10 Certificados Digitais

Imagine-se que Alice quer identificar-se para Beto, porém Beto não confia em palavras, quer uma prova que tenha algum respaldo confiável. Neste caso, basta que Alice apresente a sua carteira de identidade emitida por algum órgão de emissão legal (por exemplo, a Secretaria de Estado de Segurança Pública). Como Beto e Alice confiam neste órgão, ele reconhece Alice.

Nas comunicações eletrônicas, existe equivalente a esta situação, que é o certificado digital. Alice deve solicitar um certificado digital a uma autoridade certificadora digital (AC), podendo ser uma entidade pública ou privada que faz o papel de cartório para verificar a autenticidade de Alice. Beto, neste caso verifica a legitimidade de Alice através da assinatura digital da AC, pois Beto confia na AC.

Nos dois casos acima, a prova da identidade de Alice é importante. No mundo real usa-se a "*Carteira de Identidade*"para este fim, e no mundo digital o "*Certificado Digital*". Pode-se dizer que o certificado digital equivale à carteira de identidade pessoal, e assim como este não pode ser transferido para outra pessoa.

O padrão X.509 v3 é o adotado como padrão para a indústria, pois, permitem a inserção de outras informações para atender casos específicos de empresas [GAR 99].

Segundo Simson Garfinkel e Gene Spafford [GAR 99, IT 00], o certificado deve conter ao menos um número de versão, um número serial para o certificado, a chave pública e nome do usuário, a identificação da AC que emitiu o certificado, a assinatura digital da AC e assinada com a chave secreta da AC.

3.11 Conclusão

Neste capítulo, revisamos os conceitos fundamentais que envolvem a criptografia de dados e que serão utilizados neste trabalho. Outras técnicas podem ser aplicadas para garantir a autenticidade e confidencialidade da mensagem. No capítulo seguinte ver-se-ão quais são e como são classificados os métodos de autenticação de um usuário, quando envia uma mensagem, quando pretende fazer uma compra via internet, ou obter acesso a uma base de dados de uma empresa.

Capítulo 4

Autenticação

4.1 Introdução

Nos dias atuais, uma pessoa convive diariamente com situações em que deve identificar-se para outras pessoas, ou sistemas de informação, ou o próprio sistema deve identificar-se para ela. Então, como pode-se provar que uma pessoa é quem ela diz que é realmente? Muitos avanços têm acontecido nas últimas décadas com relação a este problema.

Este capítulo apresenta a base para o entendimento de métodos de autenticação baseados em criptografia e níveis de conhecimento do que se quer autenticar. Ele será de grande valia para a correta identificação dentro do protocolo criptográfico proposto no capítulo 9.

Segundo o dicionário [dSB 72], autenticar é o ato de tornar autêntico, reconhecer como verdadeiro. Existem várias denominações para que um documento ou usuário seja denominado autêntico. Autêntico é sinônimo de verdadeiro, genuíno, certo, verás e legalizado. Este, constitui-se em um problema para o caso de uma comunicação através da internet. Quando se tem de reconhecer como verdadeiro (autenticar) um documento em papel, coloca-se nossa assinatura e faz-se um reconhecimento da mesma em cartório, pode-se até estar com as partes envolvidas presentes, ou seja, tem-se o papel (documento) e a pessoa que vai assinar o documento no cartório.

No caso da internet, torna-se mais difícil a reunião das partes, portanto devem-se utilizar alguns métodos que auxiliam este processo de reconhecimento. No nosso dia-a-dia, as pessoas vivem-se autenticando para o mundo, no trabalho, através de crachás, para o nosso cachorro, através do cheiro, etc. A autenticação de um usuário é o principal aspecto para a segurança da informação. Uma frase resume bem a função de autenticação: *"Se não for possível identificar uma pessoa que esteja tentando entrar em um sistema, então como pode-se garantir a segurança?"*[D'A 00].

4.2 Fatores de Autenticação

Quando se fala em guardar uma informação, logo se pensa que se usará um sistema sem falhas para garantir que uma informação não seja alterada, ou roubada. Na realidade, os sistemas possuem falhas que podem ser exploradas, portanto, como se pode ter um sistema que garanta o mínimo de sigilo na informação? Existem também casos em que a quantia investida para gerar segurança é maior que o valor do que se quer guardar.

A questão do valor da informação deve ser levada em consideração no momento da escolha do método que será usado para protegê-la. Apresenta-se a seguir, alguns fatores que constituem a base para a composição de sistemas de autenticação.

O grau de segurança de que realmente se necessita para a garantia de autenticação é fruto da combinação dos fatores abaixo, segundo [LIU 01, MEN 96, JF 99], conforme segue:

1. Informação Compartilhada;
2. Algo que se Possui;
3. Medidas Biométricas;
4. Localização Espacial;
5. Localização Temporal;

6. Testemunhas.

Apresenta-se a seguir uma breve descrição dos métodos para garantir esta autenticação.

4.2.1 Autenticação por Informação Compartilhada

Em sistemas de computador, o primeiro mecanismo que se tem conhecimento de senha, foi implementado por Fernando J. Corbató, em 1963 no MIT[1], num sistema compatível de *Time Sharing*[2] em ambiente acadêmico, segundo Richard E. Smith [SMI 02].

A autenticação por conhecimento é o método mais comum, segundo Bruce Schneier [SCH 01]. Este também é conhecido por identificação positiva, que em sua grande maioria inclui: padrões de senhas, PIN (*Personal Identification Number*) - número de identificação pessoal, data de nascimento ou número de CPF[3], segundo Edgar Roberto Pacheco D'Andréa [D'A 00].

Exemplos de utilizações deste método incluem conectar-se a um sistema de computador, digitando um usuário e uma senha, como uma conexão de internet via uma linha discada, uma ligação telefônica usando um cartão de chamada, ou uma transação efetuada por telefone.

Segundo o NIST[4] [NIS 97], identificação é o mecanismo pelo qual um usuário fornece uma identidade pedida para o sistema, e autenticação é o mecanismo de estabelecimento da validação deste pedido. Nota-se então que este é um processo de duas etapas, onde a primeira chama-se identificação, consiste em informar o usuário, ou seja, quem é você, e a segunda, chama-se autenticação, informar a senha, provar que você é quem está afirmando ser [SCH 01]. Naturalmente, como todo método de autenticação, existem alguns riscos associados que devemos levar em conta, segundo Edgar Roberto

[1]Massachusetts Institute of Technology

[2]Um mecanismo pelo qual vários usuários de computador podem usar o mesmo computador interagindo ao mesmo tempo.

[3]Cadastro Pessoa Física

[4]National Institute of Standards and Technology - Department of Commerce

Pacheco D'Andréa [D'A 00]. Apresenta-se a seguir, alguns destes riscos:

- *divulgação externa*: um estranho pode obter uma senha por meio de papéis onde ela foi escrita;
- *adivinhação*: um intruso tenta diversas combinações de senhas até obter sucesso no processo de autenticação;
- *interceptação de comunicação*: um intruso monitorando o canal de comunicação pode interceptar e visualizar a senha caso ela não seja transmitida cifrada;
- *retransmissão*: um intruso pode registrar a senha de um usuário, mesmo cifrada, e retransmiti-la simulando um usuário válido;
- *comprometimento da máquina*: um intruso pode obter acesso ao equipamento com a base de dados de senhas.

Uma fragilidade encontrada neste método, é o fato da não garantia de que o conhecimento de uma pessoa estará seguro, pois se for compartilhado com outra, esta também poderá autenticar-se.

4.2.2 Autenticação por Algo que se Possui

Segundo Bruce Schneier [SCH 01], essa é uma forma antiga de controle de acesso, através da posse de um *token*[5], uma chave física, pode-se obter acesso a um cofre, uma sala, um prédio. Na Idade Média, a possessão do selo ou anel de um rei autorizava alguém a atuar como seu representante. Atualmente, este método continua tendo a mesma idéia básica, pois, o possuidor do *token*, pode autenticar-se e ter acesso a algo como se fosse um *passaporte*.

Um exemplo é a tira de cartão magnético de banco, *smartcard* (cartão de plástico do mesmo tamanho do cartão de crédito e que contém um microprocessador ou circuito integrado) e *token* (SecurID card, um gerador de senhas aleatórias que muda a senha com o tempo).

[5]Do inglês, significa um sinal, uma prova.

O problema mais sério com esse método é que os *tokens* podem ser roubados e duplicados facilmente. Para amenizar este problema, também é associada uma senha ao *token*, fazendo assim, com que o sistema passe a ter duas etapas a serem cumpridas, uma, onde prova-se possuir o *token*, e outra, provando que se conhece uma senha compartilhada para este *token*.

4.2.3 Autenticação por Medidas Biométricas

Este tem o conceito mais simples dos métodos de autenticação, pois baseia-se em você ser o seu próprio autenticador nos sistemas. Esta categoria envolve as características físicas da pessoa, chamadas biométricas. Alguns exemplos são a impressão digital, a geometria da mão, o reconhecimento da retina, da íris, da face, da assinatura, da voz e também as características da digitação no teclado. Na tabela 4.1 [LIU 01], pode-se observar uma comparação entre as diferentes tecnologias.

Segundo Simon Liu e Mark Silverman [LIU 01], as diferentes tecnologias de autenticações biométricas podem ser adaptadas a diferentes aplicações, dependendo da compreensão do perfil do usuário, as necessidades de interface com outros sistemas, o banco de dados e as condições ambientais. Atualmente, um dos mais interessantes usos de biometria envolve combinação de biometria com *smart cards*. No entanto, o maior problema tem sido encontrar uma maneira de como e onde armazenar os dados referentes a cada usuário na rede.

4.2.4 Autenticação por Localização Espacial

É um método de autenticação que aborda o lugar onde o usuário se encontra. Este método tem despertado interesse da comunidade empresarial, pois tem como premissa, que não basta ter uma identificação correta, é preciso estar em um local específico, para que o sistema possa autenticá-lo.

Este método pode usar um dispositivo de localização de posição, como o GPS[6], para que o sistema aceite a sua autenticação.

[6] *Sistema Global de Posicionamento baseado num conjunto de satélites de órbita polar dos EUA.*

Tabela 4.1: Comparações Biométricas

Características	Impressão Digital	Geometria das Mãos	Retina	Íris	Reconhecimento da face	Assinatura	Reconhecimento da Voz	Digitação do Teclado
Facilidade de uso	alto	alto	baixo	médio	médio	alto	alto	alto
Incidência de erro	secura, sujeira, idade	mão ferida, idade	óculos	iluminação pobre	iluminação, idade, óculos, cabelos	mudanças de assinatura	barulho, resfri-ado, tempo	mão ferida, idade
Precisão	alto	alto	muito alto	muito alto	alto	alto	alto	alto
Custo	*	*	*	*	*	*	*	*
Aprovação dos usuários	médio	médio	médio	médio	médio	muito alto	alto	médio
Nível requerido de segurança	alto	médio	alto	muito alto	médio	médio	médio	médio
Estabilidade a longo termo	alto	médio	alto	alto	médio	médio	médio	médio

* O grande número de fatores envolvidos fazem uma simples comparação de custo impraticável.

4.2.5 Autenticação por Localização Temporal

Determinados sistemas podem necessariamente considerar o momento da autenticação, o horário permitido, por exemplo, um sistema que prevê nos *profiles*[7] as permissões de cada usuário com horários pré-definidos para autenticação. Também pode-se adotar este método para o caso de um cofre de banco, com horários marcados para fazer a autenticação e abrir.

4.2.6 Autenticação por Testemunhas

Em um contrato há um local para que um ou dois terceiros assinem junto ao documento, atestando sua autenticidade. Este terceiro, chamado de testemunha, é uma pessoa confiável perante as partes envolvidas.

4.3 Conclusão

Neste capítulo, foi feito um relato sucinto sobre autenticação e seus principais métodos de utilização. Espera-se com isto apresentar conceitos fundamentais necessários para as questões relacionadas ao tema deste trabalho. Nota-se que o ato de garantir que pessoas ou sistemas sejam quem realmente dizem ser um para o outro é difícil e, por outro lado, não autenticar também é difícil.

[7] Lista de direitos de cada usuário nos sistema em questão, podendo ser um sistema operacional ou sistema empresarial.

Capítulo 5

Anonimato

5.1 Introdução

A liberdade de expressão e anonimato sempre foram assuntos importantes no mundo real[1]. Estes assuntos crescem em importância com o aumento da procura por trocas de informações através do mundo digital, iniciando assim, uma procura maior por privacidade nas comunicações, chegando até ao caso extremo de não querer se identificar, permanecendo no anonimato.

Este capítulo apresenta questões relacionadas ao anonimato, como técnicas de geração de anonimato, questões jurídicas e aplicações onde a utilização do anonimato é de grande valia.

As questões governamentais, políticas e sociais decorrentes do anonimato, levam a afirmar que, como quase todas as coisas, o anonimato tanto pode ser legal, quando se fala de anonimato médico, como também ilegal, quando se fala de uma atividade criminosa.

No Brasil, o anonimato, como uma forma de expressão, é considerada ilegal segundo as leis. Apesar disto, existem algumas aplicações onde o anonimato é fundamental, para a garantia de uma democracia justa. Existem situações na prática, onde é necessário o anonimato, por exemplo, o voto eletrônico, que no momento da entrega não

[1] A expressão “mundo real”, é colocada neste contexto como sendo o mundo em que se vive dia a dia.

deve ser ligado a quem votou, e compras com dinheiro digital, admitido que o comprador não queira ser identificado.

5.2 Definindo Anonimato

Existem várias formas de definir anonimato, algumas simples e outras que necessitam de um apoio matemático envolvido. Todas estas formas têm um objetivo comum, que é apresentar de maneira clara e objetiva a idéia de uma comunicação anônima na Internet.

Apresenta-se a seguir de maneira sucinta algumas formas para ilustrar melhor o anonimato nas comunicações e suas implicações.

5.2.1 Definição Clássica

Ao fazer uma busca nos dicionários da Língua Portuguesa Michaelis e Aurélio [LTD 00, dHF 89], encontra-se a definição de anonimato, conforme apresentado seguinte:

- **Anonimato**: *sm*, 1. Qualidade de anônimo. 2. Sistema de escrever sem assinar.

O Anonimato vem do grego **Anónymos** e significa sem nome ou que o não declara, e portanto define-se também esta expressão:

- **Anônimo**: *adj*, 1. Sem nome ou assinatura do autor. *sm*, 2. Pessoa que oculta seu nome.

Analisando estes dois verbetes, conclui-se que o anonimato é não querer se identificar, ou transmitir idéias sem se identificar.

5.2.2 Terminologias

Sentindo a necessidade de padronização, Andreas Pfitzmann e Marit Köhntopp em [AP 01] vêm elaborando desde julho de 2000 um documento que possa

servir de base para pesquisas e descrições em trabalhos científicos. Cabe lembrar que este documento está de acordo com a ISO99, que trata da segurança em comunicações de dados. Algumas das terminologias apresentadas neste documento são anonimato, não-ligação e não-observação:

- "**Anonimato é o estado de não ser identificado dentro de um conjunto de sujeitos, o conjunto de anonimato.**"

A palavra "sujeitos" neste contexto representa todos os emissores e receptores possíveis, podendo ser um usuário, um recurso ou um serviço, sem que a sua identidade seja revelada. Assim como a expressão "conjunto de anonimato", denota um conjunto de todos os possíveis sujeitos. Com relação aos autores[2], o conjunto de anonimato consiste nos sujeitos que podem causar uma ação. Com respeito ao endereçamento, o conjunto de anonimato consiste dos sujeitos que podem ser endereçados. Isto denota que existe um conjunto global de possíveis emissores e receptores que tem de ser analisados para garantir o anonimato do usuário e/ou sistema.

Uma outra terminologia aplicável no contexto de comunicações anônimas é a não-ligação, que se apresenta a seguir:

- "**A não-ligação de dois ou mais itens (por exemplo, sujeitos, mensagens, eventos, ações...) examinada dentro deste sistema, é dita se estes itens não estão mais e nem menos relacionados entre si, que o seu conhecimento prévio.**"

Em outras palavras, isto significa que a probabilidade de duas mensagens serem enviadas por um mesmo remetente e/ou recebidas por um mesmo receptor deve ser a mesma que se tinha através de um conhecimento prévio.

Esta é uma propriedade muito importante e deve ser considerada quando se trata de sistemas que trabalham com anonimato.

Apresenta-se o anonimato como um estado de pertencer, ou não, a um conjunto de anonimato possível, mas se é possível saber onde está este conjunto, é

[2]Considera-se autor, qualquer usuário, recurso ou serviço que inicia uma comunicação.

possível determinar um conjunto limitado de possíveis sujeitos. Para esta situação existe o conceito de não-observação conforme apresentado:

- "**A não-observação garante que nenhuma terceira parte é hábil para determinar se uma mensagem foi transmitida (enviada ou recebida) dentro de um sistema, ou não. A não-observação implica em anonimato em relação a terceiros.**"

A distinção entre o anonimato e a não-observação em sistemas de segurança está no fato que na não-observação um terceiro não tem habilidade para reconhecer quando e quem está comunicando dentro de um sistema, ainda que esteja acontecendo a comunicação e que estas partes estejam autenticadas entre si. No caso do anonimato, os sistemas estabelecem comunicação sem que sejam reveladas as sua identificações entre o emissor ou receptor (ou até ambos).

Esta também é uma propriedade importante das comunicações anônimas, pois apresenta perda referencial de onde a mensagem se originou. O total anonimato deve ser o resultado da "não-observação".

5.3 Nível de Anonimato

O fato de existir uma comunicação, leva a concluir que existem no mínimo três componentes envolvidos no processo, que são o iniciador da comunicação, o meio físico através da qual ocorre a comunicação propriamente dita e o receptor da mensagem.

Através do artigo [AP 87], A. Pfitzmann e M. Waidner apresentam dois aspectos relacionados ao anonimato. O primeiro está relacionado com o tipo de anonimato e é composto por três componentes das comunicações anônimas como: *remetente anônimo, receptor anônimo,* e a *não-ligação* entre o remetente e o receptor. Declaramos com mais detalhes cada componente:

- **Remetente anônimo**: o mecanismo que esconde a identidade do emissor da mensagem;

- **Receptor anônimo**: mecanismo que identifica quem recebe a mensagem é escondido;

- **Não-ligação**: propriedade que garante a não identificação das partes que estão comunicando entre si, mesmo que possam ser identificadas como participantes isolados de uma comunicação.

Um segundo aspecto introduzido por Pfitzmann e Waidner [AP 87], é o anonimato contra o atacante, pois este pode estar espiando a todo tempo quem envia e quem recebe.

Seguindo este conceito, Michael K. Reiter e Aviel D. Rubin [MKR 98], introduziram um terceiro aspecto para o anonimato chamado grau de anonimato.

O grau de anonimato apresenta diversas situações para classificar na verdade qual é o grau de exposição a que uma comunicação está sujeita. A figura 5.1 apresenta as diversas situações possíveis. A descrição de cada estado é apresentada a seguir.

- **Privacidade absoluta**: esta é a condição de anonimato absoluto contra qualquer atacante. O atacante não tem meios de identificar quem, com quem e quando acontece a comunicação.

- **Além da suspeita**: neste caso o atacante tem uma idéia de um conjunto possível de remetentes, porém não consegue distinguir dentro deste conjunto qual é o real remetente.

- **Inocência provável**: do ponto de vista do atacante, um remetente pode aparecer como não sendo mais a origem da mensagem.

- **Possível inocência**: do ponto de vista do atacante, há uma probabilidade não comum que o real remetente é outra pessoa.

- **Exposto**: o atacante neste caso, do ponto de vista dele, tem uma alta probabilidade de saber quem é o remetente.

- **Exposição provável**: exposição completa, ou seja, o atacante identifica o remetente, e pode provar esta identidade para quaisquer outros.

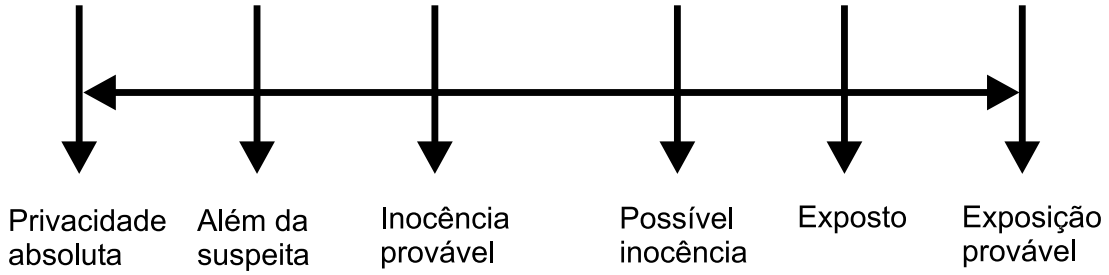

Figura 5.1: Grau de anonimato. Faixa que vai da privacidade absoluta (anonimato), onde o atacante não tem como presenciar a comunicação, para provável exposição onde o atacante pode conhecer o remetente, o receptor, e os relacionamentos entre estes.

5.4 Definindo Privacidade

De acordo com o dicionário Michaelis [LTD 00], a palavra *privacidade*, no setor de informática, pode aparecer num contexto conforme segue:

- **Privacidade** é o direito de um indivíduo em limitar o conhecimento e o controle de acesso aos dados armazenados sobre ele;

- **Privacidade de dados** é a regra que estabelece que o dado é secreto e não deve ser acessado por usuários que não foram autorizados;

- **Privacidade de informação** é a regra que estabelece que usuários não-autorizados não podem acessar bancos de dados para obter informações confidenciais de pessoas, ou que cada pessoa tem o direito de saber que informação está sendo mantida a seu respeito em um banco de dados.

De posse do apresentado, pode-se notar uma clara diferença entre *privacidade* e *anonimato*. Enquanto o anonimato prevê que em nenhum momento a identidade

do usuário será revelada, a privacidade prevê que a identidade do usuário, quando armazenada em um banco de dados, não será utilizada para outros fins, senão os que foram propostos no ato de cadastramento.

5.5 Questões Legais sobre o Anonimato no Brasil

Verifica-se na Constituição da República Federativa do Brasil [BRA 88], no Título II, Dos Direitos e Garantias Fundamentais, Capítulo I, Dos Direitos e Deveres Individuais e Coletivos, Art 5º, parágrafo IV:

> *"IV - é livre a manifestação do pensamento, sendo vedado o anonimato;´´* [BRA 88].

Esta ainda é uma questão polêmica na maioria dos países. No Brasil, como a própria Constituição coloca, isto é vedado. Vê-se que é permitida a expressão do pensamento, desde que haja uma pessoa responsável pelas conseqüências desta expressão, conforme é colocado no parágrafo V, deste mesmo capítulo;

> *"V - é assegurado o direito de resposta, proporcional ao agravo além da indenização por dano material, moral ou à imagem;´´* [BRA 88].

Outra questão levantada é o sigilo do voto, no item b), do parágrafo XXXVIII, deste mesmo texto como segue;

> *"V - XXXVIII - é reconhecida a instituição do júri, com a organização que lhe der a lei, assegurados:*
>
> *a) a plenitude de defesa;*
>
> *b) o sigilo das votações;*
>
> *c) a soberania dos veredictos;*
>
> *d) a competência para o julgamento dos crimes dolosos contra a vida;*
>
> *XXXIX - não há crime sem lei anterior que o defina, nem pena sem prévia cominação legal;´´* [BRA 88].

Nota-se que o voto tem de ser sigiloso, porém, quando se trabalha com votação eletrônica, existem meios de identificar quem votou e em quem foi votado. Isto violaria a nossa Constituição e conseqüentemente a lei.

5.6 Trabalhos Correlatos

Existem vários trabalhos que envolvem a área de geração de anonimato. O primeiro trabalho foi realizado por David L. Chaum [CHA 81] em 1981, que introduziu o conceito de rede de misturadores e sistemas de pseudônimos digitais. A idéia principal deste modelo de rede de misturadores é a perda de relação entre as mensagens de entrada e as mensagens de saída. Apresentar-se-á com mais detalhes este modelo na seção 5.7.1.

O trabalho de David L. Chaum [CHA 81] constitui-se, até o momento, uma referência para trabalhar com aplicações como geradores de correio eletrônico anônimo, *Browser de WEB*[3] anônimo, eleições seguras, como também para aplicações de sistema de pagamento anônimo, como o desenvolvido por Markus Jakobsson [MJ 99].

O modelo inicial de David L. Chaum possui o problema de não ser robusto[4], ou seja, se um servidor falhar, ou existir de um servidor não confiável, a segurança do sistema estará comprometida.

A primeira proposta para resolver o problema surgiu através de uma rede de misturadores robusta, que foi apresentada por Ogata, Kurosawa, Skao e Takatani [WO 97] em 1997. Recentemente o trabalho de M. Ohkubo[MO 00] demonstrou que é possível construir uma rede de misturadores com ambas as propriedades de robustez e eficiência no uso de cifragem através de chave simétrica.

Vários trabalhos sobre o tema robustez podem ser relacionados como, o trabalho de Markus Jakobsson, A. Juels e R. L. Rivest [MJ 02].

Frank Stajano e Roos Anderson [FS 99], apresentaram a proposta do protocolo “Cocaine”para leilões anônimos. Eles apresentaram o conceito de *anonymous*

[3]Aplicativo utilizado para pesquisa na Internet.

[4]Sistema que pode retornar ao trabalho depois de uma falha; Este item também é chamado de tolerância a falhas.

broadcast, na qual as comunicações anônimas entre o leiloeiro e os licitantes, permanecem anônimas, e os concorrentes não são revelados quando do fracasso. Esta técnica não é aplicável para uma grande área, pode-se trabalhar somente localmente.

5.7 Técnicas para Geração de Anonimato

Apresentam-se a seguir algumas técnicas que, juntas, formam a base para entender como deve-se garantir o anonimato nas comunicações através da *rede*.

5.7.1 Redes de Misturadores

Em 1981 David Chaum [CHA 81] introduziu um novo conceito para comunicações eletrônicas. Esta técnica promove o anonimato entre o remetente e o receptor da mensagem, através de uma entidade[5] denominado misturador. A proposta de um misturador, segundo [CHA 81], é esconder a correspondência entre os ítens na entrada e na saída do misturador.

Apresentar-se-á o funcionamento básico de uma rede de misturadores conforme descrito por David Chaum.

5.7.1.1 Funcionamento Básico

O modelo de rede de misturador prevê que uma mensagem cifrada, passando através de um conjunto de servidores, sofra o processo de decifragem na entrada, permutação e cifragem na saída, com isto, obtém-se a garantia de que não existe uma relação entre a mensagem de entrada e a de saída.

O sistema proposto utiliza o conceito de criptografia de chave pública como o de Taher ElGamal [ELG 85] ou de Rivest, Shamir e Adleman [RIV 78]. Através da geração de uma chave pública e outra privada, pode-se cifrar com a chave pública e decifrar com a chave privada a mensagem dentro do misturador.

[5]Neste caso entidade, pode ser entendida como um componente lógico, ou seja, um software da camada de aplicação [CG 96].

5.7.1.2 Sistema de Correio

Uma mensagem x necessita ser enviada por Alice de forma anônima para Beto, para isto será usado um gerador de anonimato chamado misturador. Uma seqüência de bits aleatórios R, será incluída a cada seqüência de endereços, garantindo assim sua unicidade. Alice conhece a chave pública de Beto k_{uB}, o seu endereço A_B e a chave pública do misturador k_{um1}.

Alice cifra com a chave k_{uB} a mensagem x, juntamente com a seqüência de bits aleatórios R_0 e associa com o endereço de Beto A_B, este conjunto é novamente adicionado R_1 e cifrado com a chave pública do misturador k_{um1}. A expressão resultante é apresentada como:

$$k_{um1}(R_1, k_{uB}(R_0, x), A_B) \longrightarrow k_{uB}(R_0, x), A_B$$

A expressão acima demonstra a saída do misturador após a decifragem com k_{rm1}. Nota-se que a seqüência de bits R_1 foi descartada pelo misturador e que resta agora enviar para Beto a mensagem, conforme mostrado:

$$k_{uB}(R_0, x), A_B \longrightarrow x$$

5.7.1.3 Endereço de Retorno

Imagine agora que Beto deseja retornar a mensagem anônima que recebeu. Neste caso, Alice, quando preparou a mensagem, deveria incluir um endereço de retorno anônimo conforme coloca David Chaum [CHA 81]. O conjunto contém uma chave pública k_{uret} para cifrar a mensagem x e a seqüência R_0, e um endereço de retorno A_{ret}.

$$k_{um1}(R_1, A_{ret}), k_{uret}(R_0, x) \longrightarrow A_{ret}, R_1(k_{uret}(R_0, x))$$

Note-se que a seqüência R_1 servirá neste caso como uma chave para cifrar o resultado após ser decifrado por k_{rm1}.

5.7.1.4 Cascata de Misturadores

O uso de uma *cascata*, ou série de misturadores enfileirados, oferece algumas vantagens que qualquer rede de misturadores única não provê. Esta idéia provê mais garantia de não-ligação entre as mensagens de entrada da série e as mensagens da saída. A única exigência feita é que a primeira rede de misturador seja confiável. Para uma mensagem que entra em uma série de misturadores, o processo é igual como se fosse um único misturador. Abaixo apresenta-se a entrada e a saída de uma série de misturadores, onde:

$$k_{umn}(R_n, k_{umn-1}(R_{n-1}, ..., k_{um2}(R_2, k_{um1}(R_1, k_{uB}(R_0, x), A_B))...)) \longrightarrow$$

é a entrada do misturador

$$k_{umn-1}(R_{n-1}, ..., k_{um2}(R_2, k_{um1}(R_1, k_{uB}(R_0, x), A_B))...) \longrightarrow$$

é a saída do primeiro misturador e a entrada do segundo e

$$\longrightarrow k_{uB}(R_0, x), A_B$$

é a saída do último misturador.

A figura 5.2 ilustra o funcionamento básico de uma rede de misturadores, várias mensagens entram na rede; na saída não deve ser capaz de determinar as entradas correspondentes.

Através de uma rede de misturadores, pode-se idealizar uma comunicação em grupo na internet, conforme mostrado na figura 5.3.

Escolheu-se implementar uma proposta baseada no artigo de Markus Jakobsson e Ari Juels [MJ 99]. Uma aplicação que poderia usar uma rede de misturadores é a votação eletrônica. Na votação eletrônica, os votos devem ser depositados em uma urna e não deverão ser de forma alguma relacionados com o votante. Isto pode ser conseguido com o uso de uma rede de misturadores.

A rede sugerida por [MJ 99] utiliza os seguintes dispositivos: rede de ordenação, comparador e o algoritmo de cifragem de ElGamal.

No apêndice A encontra-se a implementação destes mecanismos em ANSI C++.

Figura 5.2: Básico de uma rede de misturadores. Um conjunto de mensagem entra em uma rede de misturadores e não há ligação entre estes e a sua saída correspondente.

5.7.2 Sistemas de Pseudônimos

Segundo o dicionário Michaelis [LTD 00], pseudônimo é um *nome falso ou fictício*, são nomes que autenticam um usuário, porém mantém a sua identidade secreta.

A sua utilização foi feita por David Chaum [CHA 81], como um sistema de chave pública capaz de verificar as assinaturas feitas por um anônimo correspondente da chave privada.

A técnica de sistemas de pseudônimos também foi aprimorada por David Chaum [CHA 85], em 1985, como um caminho que permitia um usuário trabalhar com várias organizações, só que anonimamente. O sistema sugere que uma organização possa conhecer um usuário por um pseudônimo, e este ser diferente do pseudônimo utilizado por outra organização para o mesmo usuário. Se estas duas organizações combinarem seus bancos de dados, não poderão levantar dados em comum para um determinado usuário.

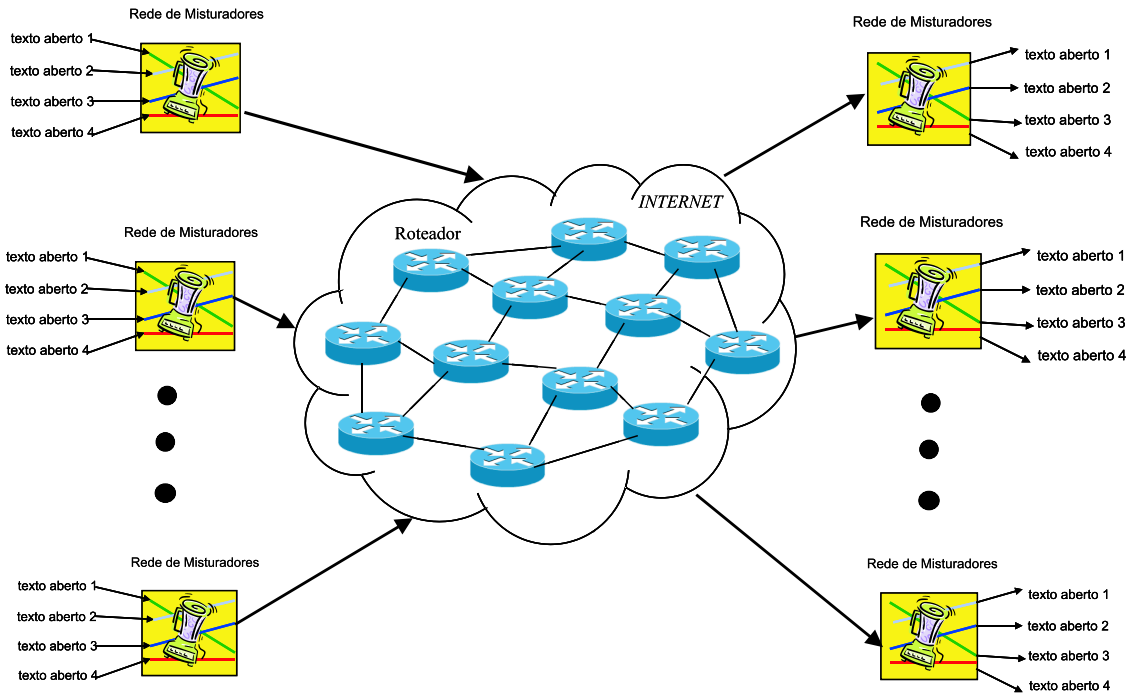

Figura 5.3: Comunicação na internet através de redes de misturadores. Observa-se que as mensagens atravessam uma rede de misturadores na entrada e na saída de uma comunicação na internet.

5.8 Conclusão

Apresentou-se neste capítulo a conceituação básica sobre o anonimato, privacidade e técnicas de geração dos mesmos. Estes conceitos são fundamentais para o desenvolvimento deste trabalho, pois estas técnicas serão utilizadas na sua elaboração. Nota-se que o anonimato pleno e absoluto não existe na Internet. Entretanto, existem diversos meios que dificultam as ações dos bisbilhoteiros digitais.

Capítulo 6

Criptografia Temporal

6.1 Introdução

A criptografia temporal possibilita para que novas aplicações possam serem realizadas no mundo digital. A ação de cifrar um documento, que também não pode ser lido antes de um determinado tempo, é chamado *enviando uma informação para ser lida no futuro* ou criptografia temporal, segundo Timothy C. May [MAY 93].

O tema deste trabalho prevê que uma informação só poderá ser revelada num futuro pré-determinado. Diante desta situação, existe a necessidade de compreender os conceitos da criptografia temporal apresentados neste capítulo.

Na seção 6.2, apresenta-se um breve histórico do tema e também alguns conceitos. Na seção 6.3, fala-se de algumas aplicações possíveis para o tema. A construção do quebra-cabeça que permite o efeito temporal será apresentado na seção 6.5 e a resolução deste, na seção 6.6.

6.2 Breve Histórico e Conceitos

Em 1993, Timothy C. May [MAY 93], mandou uma mensagem para a lista de discussão *cypherpunk@toad.com*, expressando uma idéia preliminar sobre o que chamou de “Protocolos criptográficos de liberação com tempo determinado”. Ele

apresentou na ocasião argumentos e exemplos de aplicações para este novo método de enviar uma mensagem cifrada para ser aberta no futuro.

Uma grande contribuição às pesquisas de criptografia temporal foi apresentada por Ronald L. Rivest, Adi Shamir e David A. Wagner [RLR 96], apresentando aplicações e dois possíveis métodos de implementação da criptografia temporal:

- **Quebra-cabeça temporal computacional**: consiste na criação de um problema computacional com a informação que somente será resolvido se o computador ficar processando-o continuamente por um período de tempo pré-determinado;

- **Agentes confiáveis**: consiste em uma entidade ou estrutura confiável que tem como compromisso guardar uma informação e não revelá-la antes de uma determinada data.

Este período de tempo pré-determinado é chamado de *período de ocultação*, segundo Fernando Carlos Pereira [PER 03], que também aborda no seu trabalho de mestrado o tema de criptografia temporal.

Num caso usual, para proteger uma informação, deve-se utilizar a criptografia clássica para cifrar com uma chave k, e construir o enigma temporal em cima da chave k para decifrar, garantindo que será revelada somente no futuro.

6.3 Aplicações da Criptografia Temporal

Timothy C. May [MAY 93] apresentou exemplos de aplicação como o envio de dinheiro eletrônico no futuro, admitindo que você pode estar em suspensão criobiológica e quer receber no futuro.

Ronald L. Rivest, Adi Shamir e David A. Wagner [RLR 96] e Fernando Carlos Pereira [PER 03] também apresentaram exemplo relacionados como:

- **Licitações Públicas**: em um processo de licitação, a proposta deve ser selada para a abertura numa data futura na presença de todos os participantes;

- **Pagamento programado de prestações**: no caso de um financiamento, pode-se programar a abertura das quantias de dinheiro eletrônicos referentes a cada prestação futura;

- **Diário pessoal**: alguém que possua seu diário no formato digital, programando a revelação para cinqüenta anos depois;

- **Custódia de chaves**: por exemplo, em negociações internacionais com outros países de cunho secreto, a chave pode ser confiada à custódia do governo, que poderá saber dos detalhes da negociações, mas somente no futuro;

- **Provas de vestibular**: podendo ser elaboradas para serem liberadas somente nas datas específicas de reprodução, dias antes da aplicação, ou no formato digital no momento da aplicação;

- **Denúncia de delitos**: permitir que a identidade do delator seja mantida em sigilo até uma determinada data futura, para evitar repreensões pelo feito;

- **Herança para menores**: podendo programar a liberação da quantia em dinheiro de herança, somente após o jovem completar dezoito anos.

Como se pode notar, existem diversas aplicações para o tema, tornado justificável o seu estudo.

6.4 Considerações ao Quebra-cabeça Temporal

Segundo estudos de Ronald L. Rivest, Adi Shamir e David A. Wagner [RLR 96], o quebra-cabeça baseado em complexidade computacional tem um sério problema, que é relacionado ao tempo real de processamento, pois com os recursos existentes hoje poder-se-ia montar um sistema de computação paralela que tornaria a resolução do quebra-cabeça viável muito antes do prazo pré-determinado.

Visando resolver este problema, eles propuseram a construção de um quebra-cabeça que tivesse as características de funcionamento seqüencial, ou seja, mesmo

que fosse resolvido por cluster[1] de computadores, teriam praticamente o mesmo tempo de resolução que executando em um computador individual. Atente a formação de um bebê, duas mulheres não conseguem se reunir para produzir um filho em quatro meses e meio.

As definições para a construção do quebra-cabeça ainda merece algumas considerações como:

- A utilização de uma chave k convencional pode ser um problema devido ao ataque de força bruta que, pois utilizando-se o paralelismo de computadores, pode-se descobrir a chave k;
- A capacidade de cada computador de processar cada etapa do quebra-cabeça, que na verdade trabalha com valores aproximados de T segundos para a resolução do quebra-cabeça.

6.5 Construção do Quebra-Cabeça Temporal

Imagine que Alice deseja enviar uma mensagem M para Beto, porém ele só poderá conhecer a mensagem no tempo futuro de T segundos. Para criar o quebra-cabeça, ela deve seguir os seguintes passos, segundo Ronald L. Rivest, Adi Shamir e David A. Wagner [RLR 96], baseados na repetição de exponenciações:

1. Ela gera um número $\eta = pq$, divisor de uma redução modular como sendo o produto de dois números primos p e q grandes escolhidos aleatoriamente,

$$\phi(\eta) = (p - 1)(q - 1) \tag{6.1}$$

2. Então ela calcula o fator t como $t = T\,S$ onde o valor S representa o número de exponenciações em módulo η por segundo podem serem resolvidas pelo computador responsável por resolver o quebra-cabeça, e T o tempo total de ocultação da informação;

[1] Paralelismo ou reunião de um certo número de terminais, estações, dispositivos ou posições de memória agrupadas em um local para resolução computacional com mais velocidade que um único computador.

3. Ela gera um chave K aleatória grande ($\geq\ 160\,bits$) utilizando um criptossistema convencional, como o RC5 [RIV 95]. Esta será usada para cifrar o segredo M. Convém alertar que poderia ser utilizada um criptossistema de chave pública como o ElGamal [ELG 85].

4. Utilizando a chave K, ela cifra a mensagem M, obtendo o texto cifrado

$$C_M\ =\ RC5(K\,,\,M) \tag{6.2}$$

5. Agora ela escolhe um número aleatório α módulo η, tal que satisfaça $(1\ <\alpha\ <\ \eta)$, e cifra a chave K com este valor da seguinte forma:

$$C_K\ =\ K\ +\ \alpha^{2^t}\,(mod\,\eta) \tag{6.3}$$

   Para fazer isto de forma eficiente, ela, em primeiro lugar, resolve

$$e\ =\ 2^t\,(mod\,\phi(\eta)) \tag{6.4}$$

   e então resolve

$$b\ =\ \alpha^e\,(mod\,\eta). \tag{6.5}$$

6. O resultado destas expressões é a composição do quebra-cabeça computacional $(\eta,\,\alpha,\,t,\,C_K,\,C_M)$.

Todos os outros parâmetros utilizados são apagados para não deixar vestígios.

6.6 Resolução do Quebra-Cabeça Temporal

Como a mensagem M está cifrada com a chave K, e adivinhar esta chave RC5 diretamente é inviável, deve-se resolver K em função de C_K, como

$$K\ =\ C_K\ -\ \alpha^{2^t}\,(mod\,\eta) \tag{6.6}$$

para isto basta encontrar o valor de b

$$b\ =\ \alpha^{2^t}\,(mod\,\eta) \tag{6.7}$$

Alice, para encontrar o valor correto de b, deve resolver a equação na quantidade de t exponenciações modulares quadráticas seqüenciais utilizando sempre o valor quadrático encontrado anteriormente.

Em seguida, basta encontrar K pela equação 6.6, e então novamente aplica K na equação 6.2 para revelar a mensagem M.

Para melhor entendimento do processo, apresenta-se um exemplo numérico:

6.6.1 Para Construir o Quebra-cabeça

1. Escolhe-se dois números primos $p = 13$ e $q = 19$, calcula-se $\eta = pq = 247$ e $\phi(\eta) = (p-1)(q-1) = 216$;

2. Escolhe-se um tempo de ocultação de $T = 4\,unidades\,de\,tempo$ e a capacidade do computador de resolução $S = 2\,partes/unidade\,de\,tempo$, calcula-se $t = TS = 8$ como quantidade de exponenciações para a resolução do quebra-cabeça;

3. Utilizando a cifra de ElGamal da seção 3.9.1 para o par de chaves pública e privada. Obtém-se $ku = (13,\, 6,\, 5)$ como chave pública e $kr = 9$ chave privada. Executa-se a cifragem de uma mensagem $M = 6$, utilizando a chave pública ku, obtendo

$$C_M = ElGamal(ku,\, 6) = (11,\, 9)$$

4. Escolhe-se um número aleatório α módulo η igual a 2 para cifrar a chave privada kr, primeiro encontra-se o valor de e

$$e = 2^8\,(mod\,216) = 40$$

em seguida o valor de b

$$b = 2^{40}\,(mod\,247) = 16$$

e por fim C_{kr}

$$C_{kr} = 9 + 16\,(mod\,247) = 25\,(mod\,247)$$

5. Obtém-se então a composição do quebra-cabeça como

$$quebra - cabeça \ = \ [247,\ 2,\ 8,\ 25,\ (11,\ 9)]$$

6.6.2 Resolução do Quebra-cabeça

Para a resolução do $quebra - cabeça \ = \ [247,\ 2,\ 8,\ 25,\ (11,\ 9)]$, é preciso executar $t \ = \ 8$ exponenciações quadráticas modulares, depois aplica-se na equação 6.6 para se obter a chave privada kr que será usada para a decifragem da mensagem M.

1. Primeiro deve-se encontrar b

$$b \ = \ 2^{2^8} \, (mod \, 247)$$

conforme a tabela 6.1, na oitava linha o resultado de $b \ = \ 16 \, (mod \, 247)$. Fica simples agora resolver a equação 6.6

$$kr \ = \ 25 \ - \ 16 \, (mod 247) \ = \ 9$$

Tabela 6.1: Resolução de Exponenciações Quadráticas Modulares

Interações	Intermediário	Resultados
2^{2^1}	2^2	$4 \, (mod \, 247)$
2^{2^2}	4^2	$16 \, (mod \, 247)$
2^{2^3}	16^2	$256 \, (mod \, 247)$
2^{2^4}	256^2	$81 \, (mod \, 247)$
2^{2^5}	81^2	$139 \, (mod \, 247)$
2^{2^6}	139^2	$55 \, (mod \, 247)$
2^{2^7}	55^2	$61 \, (mod \, 247)$
2^{2^8}	61^2	$16 \, (mod \, 247)$

2. O passo seguinte é utilizar a chave privada $kr \ = \ 9$ para decifrar a mensagem M conforme a seção 3.9.1.

6.6.3 Cápsula do Tempo LCS35 do MIT

Um aplicação que está funcionando desde 4 de abril de 1999, é o quebra-cabeça computacional do Laboratório de Ciência da Computação (LCS) do *Massachusetts Institute of Technology* (MIT), em comemoração ao aniversário da sua fundação.

Este quebra-cabeça foi escolhido para se resolver em aproximadamente 35 anos e utiliza a idéia apresentada nas seções 6.5 e 6.6 segundo Ronald Rivest [RIV 99].

A cápsula de nome *LCS35 Time Capsule Crypto-Puzzle* deverá processar continuamente durante 35 anos e em sua implementação foi utilizada a linguagem de programação Java. Um fato interessante é que para determinarem o crescimento de poder de resolução que os computadores iriam ter, foi utilizada a "Lei de Moore".

6.6.4 Uso de Agentes Confiáveis

Duas maneiras de utilização de agentes confiáveis foram apresentadas por Ronald L. Rivest, Adi Shamir e David A. Wagner [RLR 96], uma baseada na confiança total da chave ou dispositivo que impeça o acesso à informação secreta a uma entidade terceira confiável. Esta deverá reter a liberação da chave de acesso até que se cumpra o prazo pré-determinado de segredo.

Na outra sugestão, a idéia é que se crie um grupo de pares de chaves pública e privada, sendo que se tornaria públicas a chaves públicas para que usuários ou entidades possam cifrar seus segredos. As chaves privadas seriam guardadas para uma liberação somente no futuro.

6.7 Conclusão

Não é difícil de imaginar, a quantidade de aplicações que poderão beneficiar-se das técnicas relacionadas com a criptografia temporal.

Este capítulo apresentou a técnica de criptografia que permitem armazenar uma informação com revelação futura. Esta técnica será utilizada na elaboração deste trabalho, visando a garantia de se ter a real revelação num tempo futuro determinado.

Capítulo 7

Divisão e Compartilhamento de Segredo

7.1 Introdução

O ato de dividir e compartilhar um segredo ou chave apresenta-se como um mecanismo fundamental em situações onde não existe a confiança mútua entre as entidades envolvidas, ou quando existe problema na revelação podendo ser prejudicial a outrem.

Convive-se a todo momento com algum tipo de segredo, sejam segredos pessoais, políticos, comerciais ou científicos. Em determinados momentos, estes segredos podem ser compartilhados com pessoas que devem atuar e tomar certas decisões.

A utilização destes exerce um grande papel na sociedade atual e em diversas aplicações comerciais. Eis alguns exemplos:

- **Caso 1:** Dentro de uma empresa, o processo de assinatura de um cheque requer a participação de no mínimo dois integrantes da diretoria: Neste caso, a divisão é de uma chave (assinatura válida) que libera o cheque;
- **Caso 2:** Em um conselho fiscal de uma grande empresa, todos os conselheiros têm poder de voto igual, sendo que as decisões são tomadas pela maioria absoluta de votos a favor ou contra. Neste caso, cada conselheiro poderia ter um cartão magnético que lhe permitiria votar, como se fosse uma chave de acesso: Só que para o voto ter validade, a maioria absoluta dos conselheiros deveria estar presente;

- **Caso 3:** Em um banco, o cofre possui mecanismos para abertura somente através de chaves (senhas) de um número mínimo das pessoas autorizadas para este serviço, além de verificar em alguns casos também o horário para esta abertura. Esta é uma aplicação típica de divisão de segredo para executar uma determinada tarefa.

- **Caso 4:** Um outro exemplo é a divisão do segredo que permitia à Rússia o lançamento de uma bomba nuclear, que era dividida em três partes, o Presidente, o Ministro da Defesa e o Ministério de Defesa[1], sendo necessária a presença de pelo menos dois para efetuar o lançamento.

- **Caso 5:** Uma das maneiras de proteger as cópias de dados de uma empresa, é colocar uma senha que será dividida entre um certo número de pessoas autorizadas a retorná-lo. Fazendo isto, visa-se o caso do acontecimento de um roubo das cópias, onde seria necessária a composição de vários pedaços de senha para a composição final da informação.

Como não existe confiança mútua entre as entidades envolvidas, o compartilhamento de um segredo denota uma idéia de distribuição de responsabilidade, ou seja, para que alguém faça algo, outras pessoas devem querer também, é a divisão de poder entre várias pessoas. Este conceito é chamado de *confiança distribuída*.

Este capítulo objetiva apresentar os conceitos de como se deve compartilhar uma informação (o segredo) com segurança, permitindo que esta, possa ser revelada caso necessário, *a posteriori*.

No protocolo criptográfico proposto no capítulo 9, utiliza-se os conceitos apresentados neste capítulo para o compartilhamento da identificação do emissor da mensagem.

7.2 Conceitos e Definições

Os esquemas de compartilhamento de segredo estão relacionados com protocolos de estabelecimento de chave de múltiplas partes, segundo A. Menezes [MEN 96].

[1] Time Magazine, 4 de Maio, 1992, pág 13.

Segundo Adi Shamir e A. Menezes [SHA 79, MEN 96], a idéia de compartilhamento de segredo inicia-se com um segredo S, dividindo-o em fragmentos, chamados de partes como ($S_1, S_2, ..., S_n$), os quais são distribuídos entre um grupo de usuários específico que tem a permissão de formar subgrupos para a reconstrução do segredo original S.

Os objetivos propostos neste esquema por Adi Shamir [SHA 79] são:

1. O conhecimento de quaisquer m partes, ou mais, de partes de segredos S_n, torna a recuperação do segredo S facilmente calculável;

2. O conhecimento de quaisquer $m-1$ partes, ou menos, de partes de segredo S_n, torna a recuperação do segredo S indeterminável (considera-se que todas as partes da divisão têm pesos iguais e são apropriadas para uso).

O esquema que acaba de ser descrito é chamado de *Esquema Limiar*[2] *(m, n)* . Este será visto com mais detalhes na seção 7.5.

Um esquema de compartilhamento de um segredo pode servir como um esquema de divisão de responsabilidade sobre um determinado assunto (decisão crítica). Apresentamos a seguir algumas definições e convenções segundo Douglas Stinson e A. Menezes [STI 95, MEN 96] que serão úteis para o entendimento do restante do capítulo.

- Conjunto P: é o grupo formado por todos os participantes que vão receber as partes do compartilhamento;

- Participantes ($P_1, ..., P_n$): representa o conteúdo do conjunto P formado por n participantes;

- Valor mínimo m: representa a quantidade mínima de participantes para que possa compor o segredo novamente;

- Subconjunto B: é um subconjunto de participantes do conjunto P que se reúne para compor o segredo;

[2]O termo limiar significa: Ponto a partir do qual um efeito ou fenômeno começa a produzir-se.

- Teodoro ou T: Terceiro confiável responsável por fazer a preparação, divisão e distribuição das partes com segurança.

Segundo Douglas Stinson e A. Menezes [STI 95, MEN 96], o processo de compartilhamento pode ser dividido em três fases, sendo:

1. **Fase de inicialização**: aqui é feito o levantamento básico e a divisão para o compartilhamento, ou seja, levanta-se o segredo, o número de participantes total e como vai ser a estrutura de acesso. De posse destes parâmetros, Teodoro pode decompor o segredo em partes;

2. **Fase de distribuição das partes**: neste ponto é fundamental que o método utilizado para distribuir as partes seja seguro e livre de ataques. Uma estrutura de distribuição através de chave pública pode ser uma solução.

3. **Fase de restauração**: acontece a restauração do segredo através da união de um subconjunto de participantes autorizados.

7.3 Estrutura de Acesso

Na seção 7.2, apresenta-se o conjunto B como sendo um subconjunto de participantes autorizados a revelar o segredo. Este subconjunto B é formado por uma combinação com número de integrantes mínimo quaisquer entre todos os membros do conjunto P. A reunião de todas as possíveis combinações de subconjuntos B é um conjunto chamado de *Estrutura de Acesso*, segundo Douglas Stinson [STI 95], que se denotará por A.

No caso específico de um conjunto $P = (P_1, P_2, P_3, P_4)$, com quatro participantes e com número mínimo de subconjunto B para restaurar o segredo de três participantes, a estrutura de acesso fica:

$$A = \{(P_1, P_2, P_3), (P_1, P_2, P_4), (P_1, P_3, P_4), (P_2, P_3, P_4)\}$$

Esta estrutura pode ser representada para o esquema limiar como sendo uma combinação de três participantes em um grupo de quatro possíveis, ou seja, um esquema $(3, 4)$.

Segundo Douglas Stinson [STI 95], um esquema de compartilhamento de segredo realizado com a estrutura de acesso A é perfeito se atende a duas propriedades:

1. Se um subconjunto de participantes autorizados B, inserido no conjunto total de participantes P se reunir, ou seja $B \subseteq P$, então este subconjunto B pode determinar o valor do segredo S.

2. Se um subconjunto de participantes não-autorizados B, inserido no conjunto total de participantes P se reunir, ou seja $B \subseteq P$, então este subconjunto B não pode determinar nada sobre o valor do segredo S.

Admitindo-se a hipótese de um subconjunto C querer determinar o segredo S, sendo, o subconjunto autorizado $B \in A$ e $B \subseteq C \subseteq P$, então pode-se afirmar que o subconjunto C reunido também pode revelar o segredo S. Um superconjunto C de um conjunto autorizado P também é um conjunto autorizado, e é dito que atende às especificações de propriedades do circuito monótônico[3].

Se $B \in A$ e $B \subseteq C \subseteq P$, então $C \in A$

7.4 Divisão de Segredo do Tipo [n,n]

Descreve-se a seguir uma situação real para a divisão de segredo simples:

- Um grupo de três cientistas acabou de descobrir a fórmula para a cura do vírus da AIDS[4] e desejam guardá-la em segredo até um momento oportuno.

[3] São circuitos com construção uniforme.

[4] s. f. Acrônimo de *Acquired Immunological Deficiency Syndrome*: Síndrome da Deficiência Imunológica Adquirida. Med. Doença de origem viral, de elevada incidência e índice letal absoluto.

O problema descrito acima é que eles não confiam entre si para guardar a fórmula por completo. Eles chegaram à conclusão de que irão solicitar a um terceiro confiável para que divida a fórmula em partes iguais, de forma que uma parte isolada não revelará a fórmula completa e que somente a reunião de todas as partes pode revelar a fórmula completa. Desta forma, cada cientista fica encarregado de guardar a sua parte sabendo que ela é extremamente necessária na composição do segredo.

Este exemplo contempla uma aplicação prática no mundo real: Transportando este conceito para o mundo digital, segundo Bruce Schneier e Daniel Balparda de Carvalho [SCH 96, dC 00], esta implementação apresenta-se como tendo *n* entidades envolvidas por *n* entidades que terão que se juntar para compor a mensagem ou seja do tipo [n,n]. Ela pode ser chamada de divisão do tipo *one-time pad*, pois utiliza operação booleana *XOR*.

7.4.1 Composição da Divisão de Segredo do Tipo [n,n]

1. Teodoro (o terceiro confiável), recebe a mensagem completa para a correta divisão. Ele gera duas seqüências de bits aleatórios R e U do mesmo tamanho da mensagem S.

   - $S = 11111000$
   - $R = 10100110$
   - $U = 10000111$

2. Teodoro efetua uma operação booleana *XOR* entre a mensagem S e as seqüências aleatórias R e U, gerando uma nova seqüência de mesmo tamanho T.

   - $T = S \bigoplus R \bigoplus U$
   - $T = 11011001$

3. Teodoro distribui as seqüências R, U e T para os cientistas Alice, Beto e Carol respectivamente.

A figura 7.1 apresenta o processo de composição da divisão para melhor visualização.

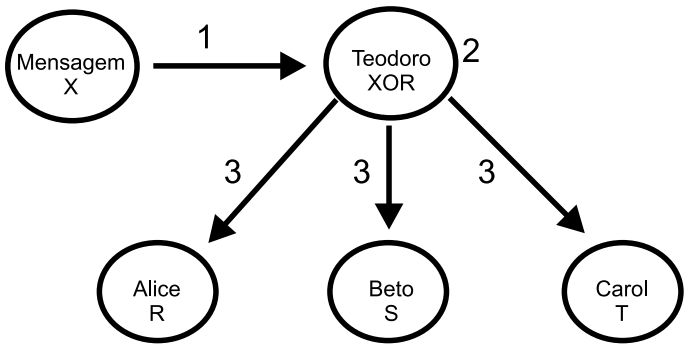

Figura 7.1: Divisão de segredo do tipo [n,n]: 1. A Mensagem S é enviada a Teodoro (a entidade confiável) para que possa ser dividida entre os participantes; 2. Teodoro gera dois números aleatórios R e U e executa a operação *XOR* ($T = S \bigoplus R \bigoplus U$) para obtenção do número T; 3. Teodoro envia para Alice, Beto e Carol os valores R, U e T respectivamente.

Para a recuperação da mensagem S original, é obrigatória a presença das três partes R, U e T conforme mostrado a seguir e na figura 7.2.

1. Para a recuperação da mensagem S, é necessário que Alice, Beto e Carol forneçam as suas partes do segredo.

2. De posse das três partes, basta que seja efetuada a operação booleana *XOR* entre elas para a obtenção da mensagem original S.

   - $S = T \bigoplus R \bigoplus U$
   - $S = 11111000$

Convém salientar que este tipo de protocolo apresenta algumas desvantagens quanto ao seu funcionamento, pois ele necessita que a seqüência de bits aleatórios seja do mesmo tamanho da mensagem que se quer compartilhar, tornando-se em determinadas situações num empecilho para a sua utilização. Outra desvantagem incorre na possível perda de uma das partes, o que impossibilitaria a recuperação da mensagem original.

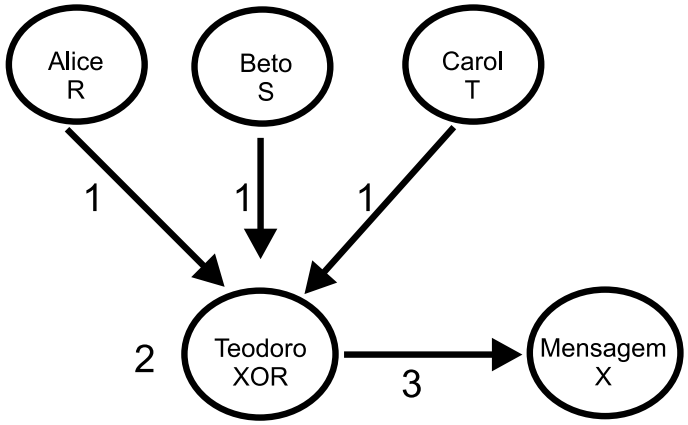

Figura 7.2: Composição de segredo do tipo [n,n]: 1. Alice, Beto e Carol enviam suas partes para a entidade confiável Teodoro; 2. Teodoro pega as partes R, U e T e executa a operação *XOR* ($S = R \bigoplus U \bigoplus T$) para obtenção da mensagem S novamente; 3. Teodoro disponibiliza a mensagem S.

7.5 Divisão de Segredo do Tipo [m,n]

Conforme visto na seção anterior, apresenta-se o caso da divisão de segredo onde o número m é necessário para reconstrução do segredo, que deve ter o número n total de entidades envolvidas.

Nesta seção, mostrar-se-á o método de construção de um esquema limiar (m, n), chamado *Esquema Limiar de Shamir* [SHA 79]. Este esquema é baseado em interpolação polinomial no plano bi-dimensional, ou seja, dado um número de pontos m no plano (x, y), como $(x_1, y_1), ..., (x_m, y_m)$ e sendo todos os x_i distintos, existe um e somente um polinômio $f(x)$ de grau $m - i$ tal que $f(x_i) = y_i$ para todo e qualquer i.

7.5.1 Fase de Inicialização

Teodoro, na fase inicial recebe o segredo $S > 0$, o número de participantes n e o valor do limiar m. Visando obter uma maior precisão, será usada a aritmética modular. O conjunto dos números inteiros módulo p, ou seja $\mathbb{Z}_p$, forma o campo no qual a interpolação é possível. Escolhe-se um valor p primo maior que o valor de S e n. O grau do polinômio será $m - 1$, então o polinômio pode ser escrito na forma:

$$y = f(x) = (a_0 + a_1 * x + a_2 * x^2 + ... + a_{m-1} * x^{m-1}) \ \ mod \ \ p \tag{7.1}$$

Na equação 7.1, o coeficiente a_0 é o segredo S que se quer dividir e os demais coeficientes são escolhidos de forma aleatória dentro do conjunto de inteiros $\mathbb{Z}_p$, e então calculam-se as várias partes do segredo:

$$y_1 = s_1 = f(1), y_2 = s_2 = f(2), ..., y_i = s_i = f(i), ..., y_n = s_n = f(n)$$

Os valores $x_1, ..., x_n$, que representam a quantidade de participantes são declarados públicos e os valores obtidos de $y_1, ..., y_n$ secretos são enviados a cada participante individualmente por método seguro.

Pode-se reescrever a equação 7.1 de forma genérica como:

$$s_i = f(x_i) = S + \sum_{i=1}^{m-1} a_i x^i \, mod \, p \tag{7.2}$$

7.5.2 Fase de Distribuição das Partes

Na fase de distribuição, os valores s_i são distribuídos de forma segura através de recursos criptográficos a seus respectivos participantres P_i.

7.5.3 Fase de Restauração do Segredo

Para a restauração do segredo S, um subconjunto B_i de participantes autorizados devem unir-se e revelar o seus segredos s_i, utilizando a **fórmula de interpolação de Lagrange**, para obtenção dos coeficientes da equação, segundo A. Menezes [MEN 96].

$$f(x) = \sum_{i=1}^{m} y_i \prod_{1 \leq j \leq m, i \neq j} \frac{x - x_j}{x_i - x_j} \tag{7.3}$$

Como o interesse é o coeficiente $f(0) = a_0 = S$, a fórmula pode ser simplificada para a obtenção de S:

$$S = \sum_{i=1}^{m} c_i y_i, \ onde \ c_i = \prod_{1 \leq j \leq m, i \neq j} \frac{x_j}{x_j - x_i} \tag{7.4}$$

Uma condição, segundo Adi Shamir [SHA 79], que denota robustez do esquema limiar (m, n), é a utilização de um valor mínimo para número m composto pela parte inteira da divisão por 2 do valor n quando acrescido de 1, conforme segue:

$$m = |\frac{n+1}{2}| \tag{7.5}$$

Seguindo esta regra, se ocorrerem perdas de partes do segredo S_i na ordem de $m - 1$, pode-se garantir a correta recuperação do segredo S.

Adi Shamir e A. Menezes [SHA 79, MEN 96] colocam que o esquema limiar (m, n) possui algumas propriedades:

1. *Perfeito*: se qualquer subconjunto de participantes autorizados B, menor ou igual que $m - 1$, tentar reconstruir o segredos, não será possível para este subconjunto levantar qualquer informação correta sobre este segredo;

2. *Ideal*: se o tamanho de cada parte s_i é igual ao segredo S;

3. *Extensível a novos participantes*: se novas partes podem ser calculadas e distribuídas sem afetar as partes anteriores já distribuídas;

4. *Possibilidade de variação no nível de controle*: através do envio de mais de uma parte s_i para um mesmo participante.

7.6 Conclusão

Neste capítulo, apresentaram-se algumas técnicas de compartilhamento de segredo com o intuito do entendimento de sua aplicação no contexto do trabalho. Não caracterizava nesta abordagem esgotar tal assunto, pois sua bibliografia é muito ampla.

Estas técnicas possuem aplicações específicas que devem ser levadas em consideração no momento de sua utilização.

Capítulo 8

Comunicação em Grupo

8.1 Introdução

Este capítulo apresenta técnicas para a comunicação em grupo baseada nos esquemas de assinatura de grupo.

Este tema foi abordado recentemente por Paulo Sérgio Ribeiro [RIB 03] no seu trabalho de mestrado para a construção de *Um Protocolo Criptográfico para Comunicação Anônima Segura em Grupo*.

Uma aplicação de esquemas de assinatura em grupo ocorre quando uma empresa necessita de um identidade corporativa. No momento da assinatura de um contrato comercial, deve conter um número mínimo de assinaturas de responsáveis legais da empresa, independente de quem for o responsável.

Este capítulo está dividido da seguinte forma: na seção 8.2 apresentam-se as propriedades que uma assinatura de grupo deve conter; na seção 8.3, as suposições feitas por Chaum e Heyst na montagem das propostas; nas seções 8.4, 8.5, 8.6 e 8.7 uma breve descrição dos protocolos proposto por Chaum e Heyst; e na seção 8.8, considerações e trabalhos correlatos.

8.2 Propriedades

David Chaum e Eugène van Heyst [CHA 91], introduziram o conceito de assinatura de grupo. Uma assinatura de grupo tem as seguintes propriedades:

1. Somente membros de um grupo pré-definido podem assinar mensagens;
2. Qualquer membro do grupo pode verificar a validade da assinatura, mas nenhum está habilitado a identificar qual membro do grupo assinou;
3. Em caso de disputas, a assinatura pode ser aberta (com ou sem a ajuda de membros do grupo) para revelar a identidade do membro do grupo que a assinou.

Lidong Chen e Torben Pryds Pedersen [CHE 95] estenderam esta idéia para fornecer ao grupo de assinaturas funções adicionais.

8.3 Suposições

Para apresentar as quatro propostas de assinatura de grupo, David Chaum e Eugène van Heyst [CHA 91] fizeram as seguintes suposições:

Suposição 1. *Para cada entidade ou pessoa, é impraticável resolver a raiz do RSA (ou seja, é impraticável para partes de números que são produtos de um mesmo número primo grande encontrar a solução; e é impraticável a solução modular de logaritmos discretos para números grande compostos*).

Suposição 2. *Para cada entidade ou pessoa, é impraticável resolver o módulo do logaritmo discreto de um número primo grande.*

8.4 Primeiro Esquema de Grupo de Assinaturas

Primeiro, a autoridade confiável e gerenciador da lista T escolhe um sistema de chave pública, o sistema de T. ElGamal [ELG 85], por exemplo. Em seguida

ele distribui a lista de chaves privadas para cada membro do grupo (todas as listas são para cada membro do grupo) e publica a lista completa das chaves públicas em um diretório público confiável para o grupo (esta publicação se dá de maneira aleatória). Convém salientar que só a autoridade confiável T possui o conhecimento total de quem está com tal chave secreta.

Para uma entidade enviar uma mensagem, basta que ela assine com uma de suas chaves secretas. O grupo ao receber a mensagem, pode verificar através do diretório público confiável a veracidade da participação do assinante no grupo, mas não pode saber a sua identidade.

Um problema que ocorre com esta proposta, é que a autoridade T tem o conhecimento de todas as chaves secretas, podendo criar novas assinaturas e também falsificá-las. Para resolver este problema, David Chaum e Eugène van Heyst [CHA 91] sugeriram uma implementação utilizando *Chaves Públicas Cegas*.

8.5 Segundo Esquema de Grupo de Assinaturas

A autoridade T escolhe dois números primos grandes diferentes. T dá a cada entidade do grupo uma chave secreta, que é um número primo grande e aleatório. Ele constrói um esquema com estes valores de forma que as entidades possam verificar que a assinatura é válida sem ter de revelar a identidade da entidade.

Em casos de disputa futura, o receptor pode executar o protocolo de verificação com cada entidade do grupo sem a ajuda da autoridade T.

Neste protocolo, a autoridade T deve fazer parte do grupo, pois se todas as entidades do grupo conspirarem, exceto um, a chave secreta deste um poderá ser revelada.

8.6 Terceiro Esquema de Grupo de Assinaturas

Neste protocolo utiliza-se a **Suposição 1** e admite-se que exista um *Diretório Público Confiável* no qual cada entidade participante possua seu módulo RSA

listado. As chaves secretas de cada membro do grupo serão a decomposição dos seus próprios módulos RSA. A autoridade T escolhe um módulo RSA N, independente dos demais, como chave privada.

8.7 Quarto Esquema de Grupo de Assinaturas

Neste protocolo utiliza-se a **Suposição 2**, tal qual foi apresentada na seção 3.9. Se uma entidade deseja assinar uma mensagem, ela seleciona um conjunto aleatório de entidades e ela mesma para a assinatura.

8.8 Considerações sobre Esquema de Grupo de Assinaturas

Jan Camenish [CAM 97], Lidong Chen e Torben Pryds Pedersen em [CHE 95] propuseram dois novos esquemas que *proporcionavam anonimato teórico da informação e anonimato computacional*. Estes esquemas permitiam a adição de novos membros no grupo, mesmo depois da inicialização e da distribuição das funções da autoridade confiável.

Jan Camenish [CAM 97] propôs um esquema de grupo de assinaturas onde a autoridade confiável T não pode falsa ou erroneamente acusar qualquer que seja a entidade do grupo, pois ela também faz parte do grupo. Eles apresentaram os itens a serem verificados para a eficiência de um esquema: a quantidade de processamento dos algoritmos de *inicialização, assinatura, verificação, e revelação*; o tamanho do grupo de chave pública e o tamanho da assinatura.

Lidong Chen e Torben P. Pedersen [LC 95] analisaram a necessidade de esquemas terem uma nova assinatura a cada mensagem para prover anonimato e concluíram ser possível colocar um limite de tamanho da chave secreta de cada membros do grupo e das informações auxiliares de dependência da autoridade no número permitido que cada assinatura pode realizar e o número de membros do grupo.

Chuan-Kun Wu e Vijay Varadharajan [EDW 99] apresentaram duas classes de algoritmos criptográficos muitos-para-um, e os aplicaram então para o desenvolvimento de esquemas de assinatura de grupo. Os grupos de chaves públicas são estabelecidos e fixados antes que as chaves privadas dos membros do grupo sejam atribuídas. Além disso, o tamanho de cada grupo de chave publica é independente do tamanho do grupo de membros.

Yuh-Min Tseng e Jinn-Ke Jan [YMT 99] apresentaram *Um Novo Esquema de Assinatura de Grupo Baseado na Identidade*, o qual resolve o problema levantado no esquema proposto por Park et al. [SP 97], pela inclusão ou exclusão de um membro no grupo, este não reconhecia mais o que já havia assinado. A segurança deste esquema é baseada na **Suposição 2.**.

8.9 Conclusão

Este capítulo teve como objetivo apresentar uma visão geral do tema de assinatura de grupo. Este assunto é de fundamental importância para as aplicações da solução apresentada por este trabalho, no sentido de garantir a correta participação de uma entidade ou indivíduo em um grupo.

Capítulo 9

Protocolos Criptográficos Propostos

9.1 Introdução

A importância do estabelecimento de um protocolo deriva da necessidade de ordenação de ações através de um determinado evento, segundo Bruce Schneier [SCH 96].

Neste capítulo apresentam-se os conceitos sobre protocolos criptográficos, os protocolos propostos e seus requisitos de segurança necessários. Ele está dividido da seguinte forma: na seção 9.2 apresentam-se as necessidades de um protocolo criptográfico; na seção 9.3, os requisitos de segurança exigidos; na seção 9.4, a notação que será utilizada no desenvolver do capítulo; nas seções 9.5, 9.6, 9.7, 9.8, 9.9 e 9.10, a descrição dos protocolos proposto juntamente com as respectivas análises; e na seção 9.11, as considerações sobre o capítulo.

9.2 Definição e Necessidades dos Protocolos Criptográficos

É importante ter uma definição clara do que é, e como deve comportar-se um protocolo criptográfico, pois só assim é possível analisá-lo em toda sua amplitude.

A definição apresentada por Bruce Schnneier e A. Menezes [SCH 96,

MEN 96], dão uma visão clara do que é um protocolo.

> *“ Um protocolo é uma série de etapas, envolvendo duas ou mais partes, projetadas para executar uma tarefa específica.´´*

Desta definição podem-se tirar algumas conclusões: se é uma *série de etapas*, então estas etapas devem estar ordenadas para o correto funcionamento; se envolve *duas ou mais partes*, fica claro que uma entidade sozinha não caracteriza um protocolo; e se são *para executar uma tarefa específica*, deve existir um protocolo para a resolução de cada tarefa específica. Bruce Schneier [SCH 96] também define como *protocolo criptográfico* o protocolo que se utiliza de criptografia.

Todo protocolo deve conter as seguintes características como satisfatórias segundo Bruce Schneier e A. Menezes [SCH 96, MEN 96]:

1. Deve existir uma seqüência do protocolo do início ao fim;

2. Deve conter duas ou mais entidades envolvidas no funcionamento do protocolo;

3. Cada um que está envolvido no protocolo deve conhecer o protocolo e todos as suas etapas para segui-lo;

4. Todos que estão envolvidos no protocolo devem concordar com o seu funcionamento;

5. O protocolo não deve ser ambíguo, ou seja, cada etapa deve ser bem definida e não permitir uma má compreensão;

6. O protocolo deve ser completo, ou seja, deverá ser especificada uma ação para cada situação possível;

7. Não deve ser possível fazer mais ou aprender mais do que está especificado no protocolo.

9.3 Requisitos de Segurança do Protocolo

Os requisitos de segurança têm grande importância na definição de um protocolo, pois dão uma visão do que ele pode fazer.

Todo sistema deve atender a alguns requisitos básicos para a garantia de funcionamento correto e com segurança.

Apresenta-se agora a lista de alguns requisitos de segurança que protocolos criptográficos propostos devem atender no decorrer de seu funcionamento.

1. **Anonimato no envio**: Deve ser possível ao emissor enviar uma mensagem de forma anônima;

    - O emissor deve ter à sua disposição meios para que possa enviar uma mensagem sem se identificar.

2. **Confiança distribuída**: A identidade do emissor da mensagem pode ser conhecida por qualquer dos subgrupos formados de um grupo de entidades previamente autorizadas;

    - Dentro de um grupo de entidades envolvidas, que tem autorização para conhecer o emissor, deve-se garantir que qualquer dos subgrupos, com quantidade mínima de entidades que recebeu a autorização, possa conhecer a identidade do emissor;

3. **Anonimato temporal (imparcialidade)**: A mensagem deve permanecer anônima desde o seu envio pelo emissor até um determinado tempo t_1 no futuro;

    - Deve haver meios de inserir um contador de tempo na cifra da mensagem para garantir que esta somente possa ser revelada após o tempo t_1 no futuro.

4. **Cifra temporal**: A identidade do emissor da mensagem poderá ser conhecida no período de tempo entre t_1 e $t_2 > t_1$;

    - Deve-se incluir junto à cifra da identidade do emissor um contador de tempo para liberar somente entre t_1 e t_2 sendo $t_1 < t_2$.

5. **Destruição da identidade**: A mensagem deverá permanecer anônima para qualquer $t > t_2$;

    - Após este tempo, a identidade deve ser destruída para sempre.

6. **Aviso ao emissor**: O emissor deve saber que a sua identidade foi revelada;

    - No período entre t_1 e t_2, se a identidade do emissor for revelada ele deve receber uma mensagem padrão de notificação do ocorrido.

7. **Autenticação**: A toda mensagem anônima deve ser possível identificar seu emissor no período de tempo específico, sendo que a identidade do emissor deve estar incluída corretamente;

    - Deve haver mecanismos que garantam a correta identificação do emissor no ato da assinatura da mensagem, ainda que esta possa não ser revelada.

8. **Prova (não-coação)**: O emissor de uma mensagem anônima não pode provar que foi ele que a emitiu;

    - Deve-se garantir a não-ligação entre o emissor e a mensagem emitida de forma anônima.

9. **Autonomia**: O emissor da mensagem não deve precisar confiar em qualquer entidade, a menos que ele queira;

    - Por questões de autonomia, a entidade emissora deve ter total conhecimento para produzir os quesitos acima citados sem que necessite de um terceiro confiável. Este é um problema difícil de resolver.

9.4 Notação Usada

Para facilitar o entendimento dos protocolos criptográficos propostos, serão usadas as seguintes notações:

Alice Usuário ou entidade que emite uma mensagem;

Beto Usuário ou entidade que recebe uma mensagem anônima;

Carol Usuário ou entidade que recebe uma mensagem anônima;

Davi Usuário ou entidade que recebe uma mensagem anônima;

Elisa Usuário ou entidade que recebe uma mensagem anônima;

Teodoro Usuário ou entidade terceira confiável do sistema;

Mário Usuário ou entidade maliciosa do sistema, que tenta descobrir a identidade do autor;

Fernando Bloco responsável pela divisão de segredo através do esquema limiar de Shamir;

Pedro Bloco responsável por aplicar o quebra-cabeça temporal e duplicar os segredos;

Roberto Bloco da rede de misturadores, responsável por criar o anonimato de Alice;

Fernando^{-1} Bloco chamado de Shamir inverso[1], responsável por resolver os quebra-cabeças e destruir quando necessário;

AD Autoridade de Datação ou Protocoladora Digital de Documentos Eletrônicos (PDDE);

AC Autoridade Certificadora, é o ponto de mútua confiança, pois esta tem uma relação de confiança com cada entidade / usuário envolvido no processo.

A figura 2.4 mostra com detalhes como fica a comunicação entre duas entidades, as fases de ocultação e de possível revelação da identificação da mensagem, bem como o que deve acontecer após decorrido o tempo limite t_2 para esta identificação na linha de tempo, para efeitos de notação utilizada.

[1]É dito como sendo Bloco Shamir Inverso devido ao fato de reunir novamente todas as partes para a reconstrução do segredo.

9.5 Protocolo Básico

Este protocolo apresenta o modelo mais simples de comunicação, conforme sua descrição e apresentado na figura 9.1.

Funcionamento:

1. Alice cifra a mensagem x com uma chave simétrica k de posse somente dela. Aplica uma função resumo h no texto cifrado concatenado com a sua identidade Id_a. Em seguida ela concatena a função resumo resultante com a sua identidade e a cifra simétrica do texto x, $e_k(x)$. Novamente ela cifra com a sua chave privada kr_a e, em seguida, com a chave pública de Beto ku_b. Envia o resultante para Beto, conforme mostrado:

$$A \rightarrow B: e_{ku_b}[e_{kr_a}(Id_a \parallel e_k(x) \parallel h(Id_a \parallel e_k(x)))].$$

2. Beto decifra com a sua chave privada kr_b. Em seguida, com a chave pública de Alice ku_a. Beto agora sabe quem enviou pela Id_a de Alice e possui a mensagem cifrada $e_k(x)$. Beto tem de esperar até Alice enviar a chave simétrica k.

3. Alice, no tempo determinado, envia para Beto a chave simétrica k para que possa decifrar e conhecer a mensagem original, como segue:

$$A \rightarrow B: e_{ku_b}[e_{kr_a}(Id_a \parallel k \parallel h(Id_a \parallel k))].$$

Análise:

O protocolo não atende ao problema proposto, pois falha ao fornecer os requisitos de segurança propostos inicialmente.

1. **Anonimato de envio**: Alice não tem como enviar uma mensagem anônima para Beto. Ele sabe quem enviou, mas não sabe o conteúdo da mensagem, isto é o contrário do problema proposto;

2. **Confiança distribuída**: Não atende, pois só existem duas entidades envolvidas;

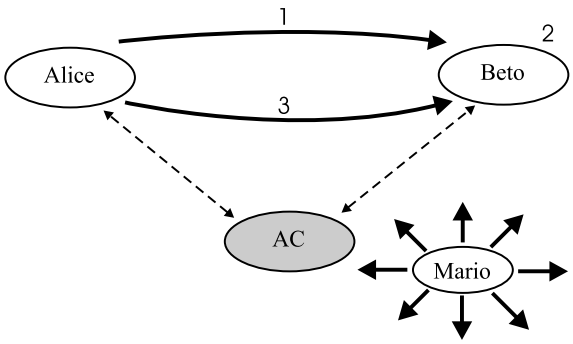

Figura 9.1: Protocolo Básico: 1. Alice envia para Beto uma mensagem x cifrada com chave simétrica k, assinada digitalmente e cifrada com a chave pública ku_b de Beto. 2. Beto decifra, obtém a mensagem cifrada $e_k(x)$ e aguarda o envio da chave simétrica por Alice. 3. Alice envia a chave simétrica k para que Beto possa decifrar e ter acesso à mensagem x. Note-se que Alice e Beto possuem uma relação de confiança com **AC**, garantindo que Beto confie na assinatura de Alice.

3. **Anonimato temporal**: Não atende, pois já é conhecido o emissor no recebimento;

4. **Cifra temporal**: Não atende, a identidade é conhecida;

5. **Destruição da identidade**: Não atende;

6. **Aviso ao emissor**: Não atende, o emissor já revelou sua identidade;

7. **Autenticação**: Atende, pois Beto sabe que foi Alice quem enviou a mensagem;

8. **Prova**: Não atende, pois Alice possui a chave privada kr_a correspondente à chave pública ku_a da cifra da mensagem, provando que foi ela quem a emitiu;

9. **Autonomia**: Não atende, necessitam da Autoridade Certificadora para provar suas identidades.

A tabela 9.1 apresenta um resumo da análise do protocolo com base nestes requisitos apresentados na seção 9.3.

Neste caso, observa-se que o protocolo só atente a um dos requisitos propostos.

Tabela 9.1: Análise do Protocolo Básico

Requisitos de Segurança	Análise para o protocolo
1. Anonimato de envio	não atende
2. Confiança distribuída	não atende
3. Anonimato temporal	não atende
4. Cifra temporal	não atende
5. Destruição da identidade	não atende
6. Aviso ao emissor	não atende
7. Autenticação	atende
8. Prova (não coação)	não atende
9. Autonomia	não atende

9.6 Protocolo com Terceiro Intermediário

Visando aumentar a satisfação dos requisitos, introduz-se uma terceira entidade confiável chamada Teodoro, responsável por receber a mensagem, enviar e somente liberar a identidade no futuro. A figura 9.2 representa esta situação.

Funcionamento:

1. Alice envia a mensagem x para Teodoro, concatenada com a sua identidade Id_a e cifrada com a chave privada kr_a de Alice e a chave pública ku_t de Teodoro, respectivamente. Visando aumentar a garantia de integridade da mensagem, pode-se usar uma função resumo h aplicada à concatenação da mensagem x com a identidade Id_a de Alice. Envia o resultante para Teodoro conforme mostrado:

$$A \rightarrow T: \; e_{ku_t}[e_{kr_a}(\, Id_a \,\|\, x \,\|\, h(Id_a \,\|\, x))].$$

2. Teodoro decifra e compara as funções resumo. Caso esteja correto, ele solicita junto ao órgão de datação (PDDE) a hora e data legal. Envia a função resumo da concatenação da mensagem x com a identidade de Alice Id_a cifrada com a sua

chave privada kr_t e novamente com a chave pública do PDDE ku_{AD}. O órgão de datação recebe, adiciona ao que foi enviado a hora e data, cifra com sua chave privada kr_{AD}, com a chave pública de Teodoro ku_t e devolve para Teodoro como segue:

$$T \rightarrow AD: e_{ku_{AD}}[e_{kr_t}(h(Id_a \parallel x))].$$

$$AD \rightarrow T: e_{ku_t}[e_{kr_{AD}}(e_{kr_t}[h(x \parallel Id_a)] \parallel data \parallel hora)].$$

3. Teodoro guarda esta informação datada e envia para Beto somente a mensagem x assinada digitalmente por ele e cifrada com a chave pública ku_b de Beto, como segue:

$$T \rightarrow B: e_{ku_b}[e_{kr_t}(x \parallel Id_t \parallel h(Id_t \parallel x))]$$

Beto recebe esta mensagem, decifra e toma conhecimento da mensagem x. Note-se que ele só conhece Teodoro.

4. Beto necessita conhecer o verdadeiro autor da mensagem x, portanto envia a Teodoro uma requisição req de identidade contendo a mensagem x anexada:

$$B \rightarrow T: e_{ku_t}[e_{kr_b}(req \parallel Id_b \parallel x \parallel h(Id_b \parallel req \parallel x))]$$

5. Teodoro recebe a requisição e verifica se já decorreu o período de ocultação de identidade para Alice $t < t_1$ e se ainda está dentro do período de possível liberação $t_1 < t < t_2$. Caso a requisição esteja compreendida dentro deste período, Teodoro enviará para Beto a identidade de Alice e a notificará da revelação de sua identidade:

$$T \rightarrow B: e_{ku_b}[e_{kr_t}(Id_a \parallel Id_t \parallel h(Id_a \parallel Id_t))]$$

$$T \rightarrow A: e_{ku_a}[e_{kr_t}(req \parallel Id_t \parallel h(Id_t \parallel req))]$$

Análise:

Esta solução apresenta vantagens em relação à apresentada na seção anterior, porém ainda não atende a todos os requisitos de segurança da seção 9.3.

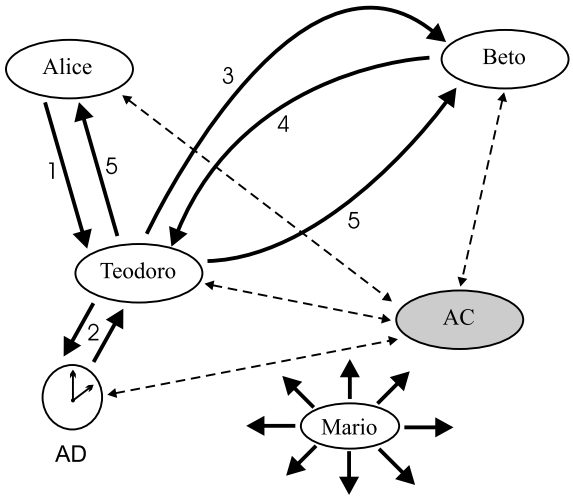

Figura 9.2: Protocolo com Terceiro Intermediário: 1. Alice envia a mensagem x e a sua identidade Id_a para Teodoro; 2. Teodoro solicita a datação da mensagem x e identidade de Alice Id_a através do envio da função resumo; 3. Teodoro separa a mensagem x da identidade de Alice e envia a mensagem para Beto com a sua identidade Id_t. 4. Beto solicita a Teodoro o envio da identidade do autor verdadeiro da mensagem x; 5. Teodoro recebe a requisição req, analisa se a data e hora permitem a revelação, caso permitir, retorna a identidade de Alice Id_a para Beto. No mesmo instante Teodoro envia para Alice uma notificação de que sua identidade foi revelada. Caso não seja solicitada a revelação dentro do prazo correto, Teodoro deverá destruir a identidade de Alice que está associada à mensagem, tornando-a anônima.

1. **Anonimato de envio**: Atende, Alice passa a ter o anonimato em relação a Beto, pois se utiliza de Teodoro como intermediário;

2. **Confiança distribuída**: Não atende, pois Alice confia em Teodoro a sua identidade;

3. **Anonimato temporal**: Atende, pois utiliza Teodoro para controlar o tempo de revelação da identidade;

4. **Cifra temporal**: Não atende, este papel está sendo feito por Teodoro;

5. **Destruição da identidade**: Atende, Teodoro sabe fazer isto;

6. **Aviso ao emissor**: Atende, Alice recebe notificação quando sua identidade é revelada;

7. **Autenticação**: Atende, desde que Teodoro forneça a identidade de Alice corretamente;

8. **Prova**: Atende, Teodoro é quem controla a revelação. Não é possível provar com certeza sem o auxílio de Teodoro;

9. **Autonomia**: Não atende, tudo é dependente de Teodoro.

A tabela 9.2 apresenta um resumo da análise do protocolo com base nestes requisitos apresentados na seção 9.3.

Apesar de atender grande parte dos requisitos propostos, ocorre uma situação de dependência do terceiro,que é Teodoro. O anonimato de envio ainda é condicional a Teodoro.

9.7 Protocolo com Rede de Misturadores

Introduziu-se uma rede de misturadores para o anonimato de envio, tirando da responsabilidade de Teodoro. A figura 9.3 representa esta situação.

Funcionamento:

Tabela 9.2: Análise do Protocolo com Terceiro Intermediário

Requisitos de Segurança	Análise para o protocolo
1. Anonimato de envio	atende
2. Confiança distribuída	não atende
3. Anonimato temporal	atende
4. Cifra temporal	não atende
5. Destruição da identidade	atende
6. Aviso ao emissor	atende
7. Autenticação	atende
8. Prova (não coação)	atende
9. Autonomia	não atende

1. Alice envia a mensagem x através de uma rede de misturadores R para gerar o anonimato no envio da mensagem a Beto, utilizando o endereço físico de Beto A_b da seguinte forma:

$$A \rightarrow R: e_{ku_R}[A_b \parallel h(A_b) \parallel e_{ku_b}(h(x) \parallel x)].$$

   Como uma rede de misturadores tem como característica a não relação entre os itens de entrada e os de saída, Beto não sabe quem lhe enviou a mensagem;

2. Alice, junto com o envio para Roberto, envia para Teodoro a mensagem assinada com sua identidade para que ele possa guardar e revelar no futuro, se necessário.

$$A \rightarrow T: e_{ku_t}[e_{kr_a}(Id_a \parallel h(Id_a \parallel x) \parallel x)]$$

3. Roberto recebe a mensagem de Alice, decifra com a sua chave privada kr_R, identifica para quem deve ser enviado a mensagem. Roberto encaminha para Beto cifrando com a sua chave privada kr_R como segue:

$$R \rightarrow B: e_{kr_R}[e_{ku_b}(\, x \parallel h(x))\,]$$

   Beto recebe, decifra, compara as funções resumo e pode ler a mensagem x;

4. Teodoro solicita à base de tempo legal AD que forneça a hora e data para o resumo da mensagem x concatenado com a identidade Id_a:

$$T \rightarrow AD: e_{ku_{AD}}[e_{kr_t}(h(Id_a \parallel x))].$$

$$AD \rightarrow T: e_{ku_t}[e_{kr_{AD}}(e_{kr_t}[h(x \parallel Id_a)] \parallel data \parallel hora)].$$

5. Beto necessita conhecer a identidade do autor da mensagem x. Para isto envia a Teodoro uma requisição req e a mensagem para comparação.

$$B \rightarrow T: e_{ku_t}[e_{kr_b}(req \parallel Id_b \parallel x \parallel h(Id_b \parallel req \parallel x))]$$

6. Teodoro recebe a requisição e verifica se já decorreu o período de ocultação de identidade para Alice $t < t_1$ e se ainda está dentro do período de possível liberação $t_1 < t < t_2$. Caso a requisição esteja compreendida dentro deste período, Teodoro enviará para Beto a identidade de Alice:

$$T \rightarrow B: e_{ku_b}[e_{kr_t}(Id_a \parallel x \parallel req \parallel h(Id_a \parallel req \parallel x))]$$

7. Caso esteja satisfeito o passo anterior, Teodoro envia uma notificação a Alice de que sua identidade foi revelada a Beto:

$$T \rightarrow A: e_{ku_a}[e_{kr_t}(req \parallel Id_t \parallel h(Id_t \parallel req))]$$

Análise:

Notam-se mais vantagens em relação à apresentada nas seções anteriores, porém ainda não atende a todos os requisitos de segurança da seção 9.3.

1. **Anonimato de envio**: Alice agora tem o anonimato em relação a Beto, pois se utiliza Roberto, que é uma rede de misturadores;

2. **Confiança distribuída**: Não atende, pois Alice confia a Teodoro a sua identidade;

3. **Anonimato temporal**: Atende, pois utiliza Teodoro para controlar o tempo de revelação da identidade;

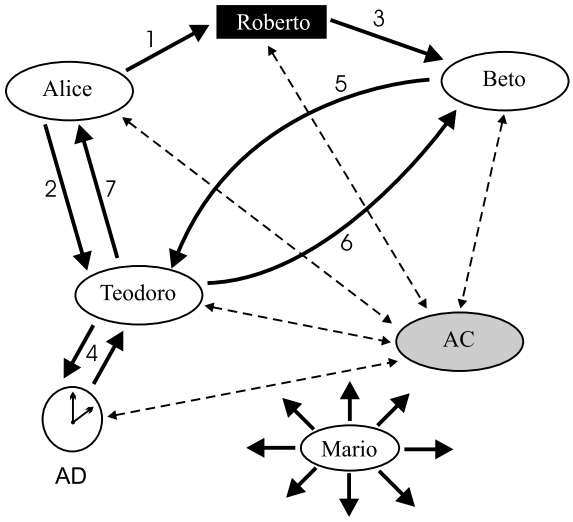

Figura 9.3: Protocolo com Rede de Misturadores: 1. Alice envia uma mensagem x para Beto através da rede de misturadores Roberto; 2. Alice envia para Teodoro a mesma mensagem x com a sua identidade Id_a; 3. Roberto envia a Beto a mensagem x sem a identidade do autor, após passar a Rede; 4. Teodoro solicita a datação da mensagem x e a identidade de Alice Id_a, e guarda para possível revelação após o tempo determinado; 5. Beto solicita a Teodoro a identidade correta do emissor da mensagem x; 6. Teodoro verifica se o prazo para revelação está correto, caso esteja, envia para Beto a identidade de Alice Id_a; 7. Teodoro, caso tenha revelado a identidade de Alice, envia-lhe uma notificação, caso contrário ele destrói a identidade de Alice Id_a relacionada com a mensagem x.

4. **Cifra temporal**: Não atende, este papel está sendo feito por Teodoro;
5. **Destruição da identidade**: Atende, Teodoro sabe fazer isto;
6. **Aviso ao emissor**: Atende, Alice recebe notificação quando sua identidade é revelada;
7. **Autenticação**: Atende, desde que Teodoro forneça a identidade de Alice corretamente;
8. **Prova**: Atende, Teodoro é quem controla a revelação e Roberto não revela sua origem. Não é possível provar com certeza sem o auxílio de Teodoro;
9. **Autonomia**: Não atende, ainda existe dependência de Teodoro.

A tabela 9.3 apresenta um resumo da análise do protocolo com base nestes requisitos apresentados na seção 9.3.

Tabela 9.3: Análise do Protocolo com Rede de Misturadores

Requisitos de Segurança	**Análise para o protocolo**
1. Anonimato de envio	atende
2. Confiança distribuída	não atende
3. Anonimato temporal	atende
4. Cifra temporal	não atende
5. Destruição da identidade	atende
6. Aviso ao emissor	atende
7. Autenticação	atende
8. Prova (não coação)	atende
9. Autonomia	não atende

A introdução da rede de misturadores fez com que o protocolo tivesse o fator anonimato bem claro. Porém ainda existe a necessidade de Teodoro como confiável para a identidade, fazendo com que o fator de atendimento de requisitos seja o mesmo ao protocolo da seção 9.6.

9.8 Protocolo com Rede de Misturadores e Quebra-Cabeça Temporal

Introduziu-se um quebra-cabeça temporal ao protocolo da seção 9.7 para a aquisição de um método de revelação de identidade no futuro, tirando da responsabilidade de Teodoro. A figura 9.4 representa esta situação.

Funcionamento:

1. Alice envia a mensagem x concatenada com a sua identidade Id_a e o endereço para onde será enviada a mensagem A_b para Pedro.

$$A \rightarrow P: e_{ku_p}[e_{kr_a}(A_b \parallel x \parallel Id_a \parallel h(A_b \parallel x \parallel Id_a))]$$

2. Pedro, que é o responsável por gerar o quebra-cabeça temporal, recebe de Alice a mensagem x, a sua identidade Ida e o endereço de Beto A_b. Ele os separa e solicita a datação de tempo legal para AD, então aplica um quebra-cabeça temporal na identidade de Alice, conforme é apresentado no na seção 6.5, gerando P_{Id_a}.

$$P \rightarrow AD: e_{ku_{AD}}[e_{kr_p}(h(Id_a \parallel x))].$$

$$AD \rightarrow P: e_{ku_p}[e_{kr_{AD}}(e_{kr_p}[h(x \parallel Id_a)] \parallel data \parallel hora)].$$

3. Pedro envia para Roberto a mensagem x concatenada com o quebra-cabeça temporal e o endereço físico A_b de Beto

$$P \rightarrow R: e_{ku_R}[e_{kr_p}(A_b \parallel x \parallel P_{Id_a} \parallel h(A_b \parallel x \parallel P_{Id_a}))].$$

   onde

$$P_{Id_a} = (\eta, \alpha, \beta, C_K, C_M)$$

   é o quebra-cabeça criado segundo Rivest, Shamir e Wagner [RLR 96].

4. Roberto decifra a mensagem e envia através da rede de misturadores R, no endereço físico de Beto A_b, a mensagem x concatenada com o quebra-cabeça P_{Id_a}.

$$R \rightarrow B: e_{kr_R}[e_{ku_b}(x \parallel P_{Id_a} \parallel h(x \parallel P_{Id_a}))]$$

Como a rede de misturadores tem como característica a não relação entre os itens de entrada e os de saída, Beto continua sem saber quem lhe enviou a mensagem;

5. Beto agora conhece a mensagem, mas não conhece a identidade do autor. Como recebeu o quebra-cabeça temporal P_{Id_a}, ele pode resolver este quebra-cabeça, conforme apresentado na seção 6.6, e no determinado tempo futuro, poderá saber quem é o autor da mensagem x. Quando a mensagem for revelada, Beto deve enviar uma notificação para Alice, de revelação de identidade.

$$B \rightarrow A: e_{ku_a}[e_{kr_b}(Id_a \parallel Id_b \parallel x \parallel h(Id_a \parallel Id_b \parallel x))]$$

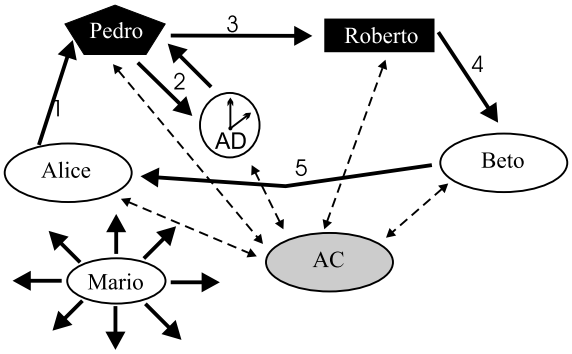

Figura 9.4: Protocolo com Rede de Misturadores e Quebra-cabeça Temporal: 1. Alice envia uma mensagem x, a sua identidade Id_a e o endereço do destinatário Beto A_b para Pedro; 2. Pedro solicita a data e hora para AD na mensagem x e a identidade Id_a; 3. Pedro pega a identidade de Alice Id_a e faz um quebra-cabeça temporal P_{Id_a}. Pedro agora concatena este quebra-cabeça, o endereço de Beto A_b com mensagem x e envia para Roberto; 4. Roberto envia esta informação para o endereço físico A_b de Beto através da rede de misturadores; 5. Beto decifra e conhece a mensagem x e o quebra-cabeça P_{Id_a}. Para saber o autor da mensagem, Beto resolve o quebra-cabeça, e depois de revelar, ele envia uma notificação ao autor (Alice), informando sobre a revelação.

Análise:

Este protocolo atende mais que os anteriores, porém ainda não atende a todos os requisitos de segurança da seção 9.3.

1. **Anonimato de envio**: Alice agora tem o anonimato em relação a Beto, pois utiliza Roberto que é uma rede de misturadores;

2. **Confiança distribuída**: Não atende, pois Beto só depende dele para revelar a identidade de Alice;

3. **Anonimato temporal**: Atende, pois só será revelada a identidade após resolver o quebra-cabeça temporal;

4. **Cifra temporal**: Não atende, pois Beto pode revelar a identidade após o tempo t_2, mesmo Pedro gerando a cifra com quebra-cabeça temporal a ser resolvido;

5. **Destruição da identidade**: Não atende, não existe nenhuma garantia de que Beto não vá revelar a identidade e sim destruí-la;

6. **Aviso ao emissor**: Atende, Alice recebe notificação de Beto quando sua identidade é revelada;

7. **Autenticação**: Atende, a identidade contida na cifra do quebra-cabeça temporal realmente é a de Alice;

8. **Prova**: Atende, Beto não tem como saber quem emitiu a mensagem, a menos que ele resolva o quebra-cabeça temporal;

9. **Autonomia**: Atende, Alice agora sabe o caminho para enviar uma mensagem anônima com possível revelação no futuro.

A tabela 9.4 apresenta um resumo da análise do protocolo com base nestes requisitos apresentados na seção 9.3.

Com a construção do quebra-cabeça, obteve-se um ganho grande para o protocolo, principalmente a saída de Teodoro em relação ao protocolo da seção 9.7.

9.9 Protocolo com Rede de Misturadores, Quebra-Cabeça Temporal e Compartilhamento de Segredo

Introduziu-se agora o compartilhamento de segredo com o intuito de dividir o controle da revelação da identidade do autor da mensagem anônima. Utiliza-se

Tabela 9.4: Análise do Protocolo com Rede de Misturadores e Quebra-cabeça Temporal

Requisitos de Segurança	**Análise para o protocolo**
1. Anonimato de envio	atende
2. Confiança distribuída	não atende
3. Anonimato temporal	atende
4. Cifra temporal	não atende
5. Destruição da identidade	não atende
6. Aviso ao emissor	atende
7. Autenticação	atende
8. Prova (não coação)	atende
9. Autonomia	atende

para isto o esquema limiar de Shamir apresentado na seção 7.5. A figura 9.5 representa esta situação.

Funcionamento:

1. Alice envia para Fernando a mensagem x concatenada com a sua identidade Id_a e o endereço para onde será enviada a mensagem A_g.

$$A \rightarrow F: e_{ku_f}[e_{kr_a}(A_g \parallel x \parallel Id_a \parallel h(A_g \parallel x \parallel Id_a))]$$

2. Fernando, que é o responsável por preparar o compartilhamento do segredo S com a identidade de Alice Id_a, sabe que o esquema limiar é do tipo $(m,\, n)$. Fernando constrói o compartilhamento conforme apresentado na seção 7.5, obtendo as partes s_i para cada membro do grupo autorizado A_g. Fernando agora junta a cada parte do segredo s_i, a mensagem x e o endereço de cada destinatário do grupo $A_g = (A_b,\, A_c,\, A_d,\, A_e)$. Envia estas mensagens para Pedro incluir à cifra temporal.

$$F \rightarrow P: e_{ku_p}[e_{kr_f}(A_b \parallel x \parallel s_1 \parallel h(A_b \parallel x \parallel s_1))]$$

$$F \rightarrow P: e_{ku_p}[e_{kr_f}(A_c \parallel x \parallel s_2 \parallel h(A_c \parallel x \parallel s_2))]$$

$$F \rightarrow P: e_{ku_p}[e_{kr_f}(A_d \parallel x \parallel s_3 \parallel h(A_d \parallel x \parallel s_3))]$$

$$F \rightarrow P: e_{ku_p}[e_{kr_f}(A_e \parallel x \parallel s_4 \parallel h(A_e \parallel x \parallel s_4))]$$

3. Pedro, que é o responsável por gerar o quebra-cabeça temporal, recebe de Fernando a mensagem x, uma parte do segredo, por exemplo $s1$ e o endereço do destinatário, por exemplo A_b, para cada parte do segredo $S = Id_a$. Ele os separa e solicita a datação de tempo legal para AD em cada uma das partes do segredo s_i.

   Para s_1, fica da seguinte forma:

$$P \rightarrow AD: e_{ku_{AD}}[e_{kr_p}(h(s_1 \parallel x))].$$

$$AD \rightarrow P: e_{ku_p}[e_{kr_{AD}}(e_{kr_p}[h(x \parallel s_1)] \parallel data \parallel hora)].$$

   A datação das demais partes são semelhantes à apresentada acima.

4. Em seguida, ele aplica um quebra-cabeça temporal em cada um dos segredos s_i, sendo $(i = 0, 1, 2, 3, 4)$, conforme é apresentado na seção 6.5, gerando os P_{s_i}. Pedro envia para Roberto a mensagem x concatenada com o quebra-cabeça temporal P_{s_i} e o endereço físico A de cada membro do grupo A_g.

   Para o caso de Beto fica:

$$P \rightarrow R: e_{ku_R}[e_{kr_p}(A_b \parallel x \parallel P_{s_1} \parallel h(A_b \parallel x \parallel P_{s_1}))].$$

   onde

$$P_{s_1} = (\eta, \alpha, \beta, C_K, C_{M_1})$$

   é o quebra-cabeça criado segundo Rivest, Shamir e Wagner [RLR 96]. Convém salientar que os parâmetros utilizados para a construção do quebra-cabeça, são os mesmos para todas as partes do segredo $S = Id_a$ inicial. Para as demais partes o processo é o mesmo.

5. Roberto, que é uma rede de misturadores R, envia para cada um dos membros do grupo A_g, as suas partes correspondentes de forma anônima. Isto é feito para o endereço físico de cada membro, por exemplo de Beto A_b, concatenado com a mensagem x e concatenada com o quebra-cabeça P_{s_1}, por exemplo.

Para cada membro do grupo fica:

$$R \rightarrow B: e_{kr_R}[e_{ku_b}(x \parallel P_{s_1} \parallel h(x \parallel P_{s_1}))]$$

$$R \rightarrow C: e_{kr_R}[e_{ku_c}(x \parallel P_{s_2} \parallel h(x \parallel P_{s_2}))]$$

$$R \rightarrow D: e_{kr_R}[e_{ku_d}(x \parallel P_{s_3} \parallel h(x \parallel P_{s_3}))]$$

$$R \rightarrow E: e_{kr_R}[e_{ku_e}(x \parallel P_{s_4} \parallel h(x \parallel P_{s_4}))]$$

Como a rede de misturadores tem como característica a não relação entre os itens de entrada e os de saída, Beto e qualquer membro do grupo continuam sem saber quem enviou a mensagem para ele;

6. O grupo agora conhece a mensagem, mas não conhece a identidade do autor. Como cada membro recebeu um quebra-cabeça temporal P_{s_i}, onde $s_i = 1, 2, 3, 4$, eles podem resolver seus quebra-cabeças conforme apresentado na seção 6.6 e, no determinado intervalo de tempo futuro, poderão saber quem é o autor da mensagem x. Quando a mensagem for revelada, cada membros que revelou a identidade deve enviar uma notificação de revelação de identidade para Alice.

   No caso de Beto fazer parte deste subgrupo que revelou a identidade, a notificação fica:

$$B \rightarrow A: e_{ku_a}[e_{kr_b}(Id_a \parallel Id_b \parallel x \parallel h(Id_a \parallel Id_b \parallel x))]$$

Análise:

Este protocolo atende mais requisitos que os anteriores, porém ainda não atende a todos os requisitos de segurança da seção 9.3.

1. **Anonimato de envio**: Alice agora tem o anonimato em relação ao grupo autorizado, pois utiliza Roberto que é uma rede de misturadores;

2. **Confiança distribuída**: Atende, pois Fernando faz a divisão da sua identidade Id_a em partes para cada participante do grupo;

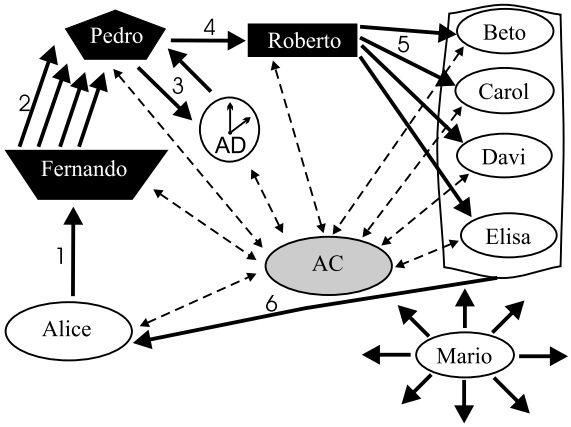

Figura 9.5: Protocolo com Rede de Misturadores, Quebra-cabeça Temporal e Compartilhamento de Segredo: 1. Alice envia uma mensagem x, a sua identidade Id_a e o endereço do grupo dos destinatários A_g para Fernando; 2. Fernando sabe quem são os destinatários do grupo de participantes autorizados no esquema limiar $(m,\, n)$, prepara a divisão da identidade de Alice como o segredo $S = Id_a$ e envia cada uma das partes com seus respectivos destinatários para Pedro; 3. Pedro solicita a data e hora para AD, para cada um dos participantes, composta pela mensagem x e o pedaço do segredo s_i sendo $s_i = 1, 2, 3, 4$; 4. Pedro pega cada pedaço do segredo s_i e faz um quebra-cabeça temporal P_{s_i}. Pedro agora concatena este quebra-cabeça temporal, o endereço respectivo de cada membro do grupo A_g com mensagem x e envia para Roberto; 5. Roberto envia esta informação para o endereço físico de cada membro, de Beto por exemplo A_b, através da rede de misturadores; 6. Cada participante decifra e conhece a mensagem x e o quebra-cabeça P_{s_i} correspondente. Para saber o autor da mensagem, um subgrupo pode se reunir, após resolver o quebra-cabeça temporal, e revelar o autor da mensagem x. Após a revelação, cada participante deste subgrupo deve enviar uma notificação ao autor (Alice), informando sobre a revelação.

3. **Anonimato temporal**: Atende, pois só será revelada a identidade após resolver o quebra-cabeça temporal;

4. **Cifra temporal**: Não atende, pois nada impede que um subgrupo seja formado após o tempo $t > t_2$, mesmo Pedro gerando a cifra com quebra-cabeça temporal a ser resolvido;

5. **Destruição da identidade**: Não atende, não existe nenhuma garantia de que os participantes não vão revelar a identidade e sim destruí-la;

6. **Aviso ao emissor**: Atende, Alice recebe notificação do subgrupo quando sua identidade é revelada;

7. **Autenticação**: Atende, a identidade contida na cifra do quebra-cabeça temporal realmente é a de Alice;

8. **Prova**: Atende, o grupo não tem como saber quem emitiu a mensagem, a menos que resolvam o quebra-cabeça temporal e o compartilhamento;

9. **Autonomia**: Atende, Alice agora sabe o caminho para enviar uma mensagem anônima com possível revelação no futuro, desde que um subgrupo de um grupo se reúna para isto.

A tabela 9.5 apresenta um resumo da análise do protocolo com base nestes requisitos apresentados na seção 9.3.

Com o compartilhamento do segredo, teve-se mais um requisito de segurança satisfeito em relação ao protocolo da seção 9.8.

Tabela 9.5: Análise do Protocolo com Rede de Misturadores, Quebra-cabeça Temporal e Compartilhamento de Segredo

Requisitos de Segurança	Análise para o protocolo
1. Anonimato de envio	atende
2. Confiança distribuída	atende
3. Anonimato temporal	atende
4. Cifra temporal	não atende
5. Destruição da identidade	não atende
6. Aviso ao emissor	atende
7. Autenticação	atende
8. Prova (não coação)	atende
9. Autonomia	atende

9.10 Protocolo com Rede de Misturadores, Quebra-Cabeça Temporal, Compartilhamento de Segredo e Bloco Inverso

Mesmo com a introdução do compartilhamento de segredo, o protocolo ainda não atende a todos os requisitos de segurança previstos no na seção 9.3. Isto porque não é satisfeito o tempo t_2 limite para revelação da identidade, que em todos os casos foi considerado como infinito (∞).

Com a introdução de um bloco de compartilhamento de Shamir inverso, associado a quebra-cabeças com tempo variável de solução, pode-se resolver o restante dos requisitos. A figura 9.6 representa esta situação.

Funcionamento:

1. Alice envia para Fernando a mensagem x concatenada com a sua identidade Id_a e

o endereço para onde será enviada a mensagem A_{f1}.

$$A \rightarrow F: e_{ku_f}[e_{kr_a}(A_{f1} \parallel x \parallel Id_a \parallel h(A_{f1} \parallel x \parallel Id_a))]$$

2. Fernando, que é o responsável por preparar o compartilhamento do segredo S com a identidade de Alice Id_a, sabe que o esquema limiar é do tipo (m, n), ou $(3, 4)$ do exemplo. Fernando constrói o compartilhamento conforme apresentado na seção 7.5, obtendo as partes s_i para cada membro do grupo autorizado A_g. Fernando agora junta para cada parte do segredo s_i, a mensagem x e o endereço do destinatário A_{f1}. Envia estas mensagens para Pedro incluir a cifra temporal.

$$F \rightarrow P: e_{ku_p}[e_{kr_f}(A_{f1} \parallel x \parallel s_1 \parallel h(A_{f1} \parallel x \parallel s_1))]$$

$$F \rightarrow P: e_{ku_p}[e_{kr_f}(A_{f1} \parallel x \parallel s_2 \parallel h(A_{f1} \parallel x \parallel s_2))]$$

$$F \rightarrow P: e_{ku_p}[e_{kr_f}(A_{f1} \parallel x \parallel s_3 \parallel h(A_{f1} \parallel x \parallel s_3))]$$

$$F \rightarrow P: e_{ku_p}[e_{kr_f}(A_{f1} \parallel x \parallel s_4 \parallel h(A_{f1} \parallel x \parallel s_4))]$$

Note que todas as partes foram encaminhadas para $Fernando^{-1}$, que ficará responsável por distribuir as partes para o subgrupo de participantes diferentes, quando for solicitado.

3. Pedro, que é o responsável por gerar o quebra-cabeça temporal, recebe de Fernando a mensagem x, as parte, do segredo, por exemplo s_1 e o endereço do destinatário A_{f1}, para cada parte do segredo $S = Id_a$. Ele gera uma cópia idêntica de cada parte do segredo, formando um novo conjunto de partes $S' = s'_1, s'_2, s'_3, s'_4$. Para estes dois conjuntos ele solicita a datação de tempo legal para AD em cada uma das partes do segredo s_i e s'_i.

   Para s_1 e s'_1 respectivamente, fica da seguinte forma:

$$P \rightarrow AD: e_{ku_{AD}}[e_{kr_p}(h(s_1 \parallel x))]$$

$$AD \rightarrow P: e_{ku_p}[e_{kr_{AD}}(e_{kr_p}[h(x \parallel s_1)] \parallel data \parallel hora)]$$

$$P \rightarrow AD: e_{ku_{AD}}[e_{kr_p}(h(s'_1 \parallel x))]$$

$$AD \rightarrow P: e_{ku_p}[e_{kr_{AD}}(e_{kr_p}[h(x \parallel s'_1)] \parallel data \parallel hora)]$$

A datação das demais partes são semelhantes à apresentada acima.

4. Em seguida, ele aplica um quebra-cabeça temporal que deve conter tempos de resolução diferente para os conjuntos S e S'. Para o conjunto S, a resolução do quebra-cabeça deve acontecer no instante de tempo t_1 da proposta inicial do problema apresentado na seção 2.5. Já para o conjunto S', os tempos de resolução do quebra-cabeça devem aumentar para cada parte do segredo s'_i, a partir do mesmo ponto $t \geq t_1$ e respeitar o tempo final igual a t_2, onde terá de haver uma quantidade mínima de s'_i igual ao número limiar m do esquema de Shamir. A implementação para cada parte de segredo s_i, sendo $(i = 0, 1, 2, 3, 4)$, se dá conforme é apresentado no na seção 6.5, gerando os P_{s_i} e utilizando parâmetros de configuração iguais. Para a implementação de cada parte de segredo s'_i, sendo $(i = 0, 1, 2, 3, 4)$, basta que Pedro selecione m (limiar) partes quaisquer de segredo s'_i e aplique a implementação limiar de Shamir para resolução no limite de tempo t_2. Sucessivamente Pedro deve pegar as partes restantes de segredo s'_i e aplicar tempos de resolução menores, decrementando, mas nunca abaixo do tempo inicial t_1. Isto pode ser feito através dos parâmetros da geração do quebra-cabeça, mais precisamente no período T em segundos de cifra com quebra-cabeça temporal.

   Pedro envia para Roberto a mensagem x concatenada com os quebra-cabeças temporais P_{s_i} e $P_{s'_i}$, o endereço físico A_{f1} do $Fernando^{-1}$.

   Para os s_1 e s'_1, ficam respectivamente:

$$P \rightarrow R: e_{ku_R}[e_{kr_p}(A_{f1} \parallel x \parallel P_{s_1} \parallel h(A_{f1} \parallel x \parallel P_{s_1}))].$$

$$P_{s_1} = (\eta, \alpha, \beta, C_K, C_{M_1})$$

   e

$$P \rightarrow R: e_{ku_R}[e_{kr_p}(A_{f1} \parallel x \parallel P_{s'_1} \parallel h(A_{f1} \parallel x \parallel P_{s'_1}))].$$

$$P_{s'_1} = (\eta, \alpha, \beta, C'_K, C'_{M_1})$$

   onde P_{s_1} e $P_{s'_1}$ são quebra-cabeças criados segundo Rivest, Shamir e Wagner [RLR 96]. Para as demais partes, o processo é o mesmo.

5. Roberto, que é uma rede de misturadores R, envia para $Fernando^{-1}$ cada uma das partes de forma anônima. Isto é feito para o endereço físico de $Fernando^{-1}$, concatenado com a mensagem x e concatenada com os quebra-cabeças P_{s_1} e $P_{s'_1}$, por exemplo.

   Para cada parte fica:

$$R \rightarrow F1 : e_{kr_R}[e_{ku_{f1}}(x \parallel P_{s_1} \parallel h(x \parallel P_{s_1}))]$$

$$R \rightarrow F1 : e_{kr_R}[e_{ku_{f1}}(x \parallel P_{s'_1} \parallel h(x \parallel P_{s'_1}))]$$

   Como a rede de misturadores tem como característica a não relação entre os itens de entrada e os de saída, $Fernando^{-1}$ continua sem saber quem enviou a mensagem. Neste momento é feita a entrega da mensagem x ao grupo;

6. $Fernando^{-1}$, será chamado de "**Bloco Inverso de Shamir**", responsável por resolver os quebra-cabeças para revelação ou comparação e destruição. Ele resolve todos os quebra-cabeças P_{s_i} e $P_{s'_i}$. Fazendo isto em t_1 os quebra-cabeças de P_{s_i} resolvidos e com possibilidade de revelação, caso um subgrupo qualquer do grupo autorizado a conhecer o autor, solicite a identidade. Já os $P_{s'_i}$ vão resolvendo aos poucos, e à medida que isto acontece, $Fernando^{-1}$ compara o resultado s'_i com os s_i existentes, se for encontrada a igualdade, então ele destrói as duas partes s'_i e s_i. Note-se que à medida que o tempo passa, ou seja t tende a t_2, a quantidade partes para a revelação diminui. Finalmente em t_2, todos os s'_i resolvidos, e para qualquer $t > t_2$, não existe mais partes para revelar o segredo S.

   Convém salientar que entre os tempos t_1 e t_2, a qualquer momento pode ser solicitada a revelação da identidade do autor.

   No caso de algum subgrupo solicitar a identidade do autor, cada um receberá uma mensagem como segue, no caso de Beto fazer parte deste subgrupo:

$$F1 \rightarrow B : e_{ku_b}[e_{kr_{f1}}(Id_a \parallel Id_{f1} \parallel x \parallel h(Id_a \parallel Id_{f1} \parallel x))]$$

7. Quando a mensagem for revelada, $Fernando^{-1}$ enviará uma notificação de revelação de identidade para Alice.

$$F1 \rightarrow A : e_{ku_a}[e_{kr_{f1}}(Id_a \parallel Id_{f1} \parallel x \parallel h(Id_a \parallel Id_{f1} \parallel x))]$$

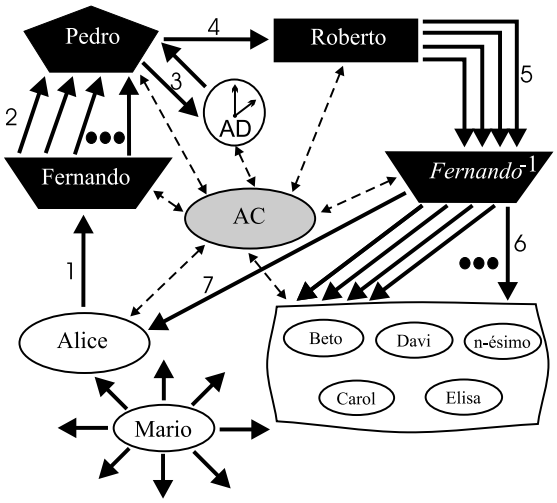

Figura 9.6: Protocolo com Rede de Misturadores, Quebra-cabeça Temporal, Compartilhamento de Segredo e Inverso: 1. Alice envia uma mensagem x, a sua identidade Id_a e o endereço do grupo dos destinatários A_g para Fernando; 2. Fernando sabe quem são os destinatários do grupo de participantes autorizados no esquema limiar (m, n), prepara a divisão da identidade de Alice como o segredo $S = Id_a$ e envia todas as partes com o endereço A_{f1} para Pedro; 3. Pedro duplica o conjunto s_i para s_i' idêntico e solicita a data e hora para AD, para os dois conjuntos 4. Pedro pega o conjunto s_i e aplica quebra-cabeça temporal com parâmetros de configuração iguais, para o conjunto s_i' é aplicado o quebra-cabeça com parâmetros de revelação diferentes. Pedro agora concatena estes quebra-cabeças temporais, o endereço A_{f1} com mensagem x e envia através de Roberto; 5. Roberto envia estas informações para o endereço físico A_{f1}, através da rede de misturadores, e este distribui a mensagem x ao grupo; 6. $Fernando^{-1}$ resolve todos os quebra-cabeças, sendo que em t_1, os quebra-cabeças s_i estão resolvidos, e que os quebra-cabeças s_i' vão se resolvendo entre t_1 e t_2. À medida que os s_i' são resolvidos, faz-se uma comparação de igualdade com os s_i já resolvidos, se forem iguais, $Fernando^{-1}$ os destrói. Caso seja solicitada a identidade do autor dentro do período permitido, então $Fernando^{-1}$ lhes enviará; 7. $Fernando^{-1}$ envia uma notificação ao autor (Alice) de que a sua identidade foi revelada.

Análise:

Este protocolo atende mais que os anteriores, porém ainda não atende a todos os requisitos de segurança da seção 9.3.

1. **Anonimato de envio**: Alice agora tem o anonimato em relação ao grupo autorizado, pois se utiliza Roberto, que é uma rede de misturadores;

2. **Confiança distribuída**: Atende, pois Fernando faz a divisão da sua identidade Id_a em partes;

3. **Anonimato temporal**: Atende, pois só será revelada a identidade após resolver o quebra-cabeça temporal;

4. **Cifra temporal**: Atende, pois a identidade após o tempo $t > t_2$, não terá como ser revelada, visto que não existem mais partes para isto;

5. **Destruição da identidade**: Atende, não existe mais nenhuma parte do segredo para montá-lo novamente;

6. **Aviso ao emissor**: Atende, Alice recebe notificação do $Fernando^{-1}$ quando sua identidade é revelada;

7. **Autenticação**: Atende, a identidade contida na cifra do quebra-cabeça temporal realmente é a de Alice;

8. **Prova**: Atende, o grupo não tem como saber quem emitiu a mensagem, a menos que resolvam o quebra-cabeça temporal e o compartilhamento;

9. **Autonomia**: Atende, Alice agora sabe o caminho para enviar uma mensagem anônima com possível revelação no futuro, desde que um subgrupo de um grupo se reúna para isto.

A tabela 9.6 apresenta um resumo da análise do protocolo com base nestes requisitos apresentados na seção 9.3.

Com o compartilhamento do segredo, satisfez-se mais um requisito de segurança em relação ao protocolo da seção 9.9.

Tabela 9.6: Análise do Protocolo com Rede de Misturadores, Quebra-cabeça Temporal, Compartilhamento de Segredo e Inverso

Requisitos de Segurança	Análise para o protocolo
1. Anonimato de envio	atende
2. Confiança distribuída	atende
3. Anonimato temporal	atende
4. Cifra temporal	atende
5. Destruição da identidade	atende
6. Aviso ao emissor	atende
7. Autenticação	atende
8. Prova (não coação)	atende
9. Autonomia	atende

A tabela 9.7 apresenta uma comparação que resume os requisitos que cada protocolo atendeu, comparando cada um entre eles.

9.11 Conclusão

Este capítulo apresentou em seqüência de atendimento dos requisitos de segurança e os protocolos criptográficos propostos.

Apesar de somente o sexto protocolo atender ao objetivo geral e requisitos de segurança, não se deve tirar os méritos e as funcionalidades dos demais, que perfeitamente podem ser utilizados para outras aplicações.

O Bloco Inverso de Shamir se coloca como o ponto base para o funcionamento do sexto protocolo, pois sem este fica difícil o correto cumprimento dos requisitos de segurança apresentados.

Convém salientar, que a utilização de qualquer dos protocolos em aplicações que não exigem maior rigor nos quesitos de segurança é possível. Desta forma há um aumento na quantidade de aplicações que podem ser atendidas com os protocolos.

Tabela 9.7: Comparação entre os Protocolos Criptográficos com relação aos Requisitos de Segurança

Protocolos Propostos	1	2	3	4	5	6	7	8	9
Básico							X		
Terceiro Intermediário	X		X		X	X	X	X	
Rede de Misturadores	X		X		X	X	X	X	
Rede de Misturadores e Quebra-cabeça Temporal	X		X			X	X	X	X
Rede de Misturadores, Quebra-cabeça Temporal e Compartilhamento de Segredo	X	X	X			X	X	X	X
Rede de Misturadores, Quebra-cabeça Temporal, Compartilhamento de Segredo e Inverso	X	X	X	X	X	X	X	X	X

1. Anonimato de envio; 2. Confiança distribuída; 3. Anonimato temporal; 4. Cifra temporal; 5. Destruição da identidade; 6. Aviso ao emissor; 7. Autenticação; 8. Prova(não coação); 9. Autonomia.

Capítulo 10

Denúncia

10.1 Introdução

Este capítulo apresenta as formas de se efetuar uma denúncia. Também serão abordadas questões sobre a legislação de denúncia.

Segundo o Dicionário Michaelis [MGR 98],

- **Denúncia**: 1. Ato ou efeito de denunciar. 2. Delação.

ou seja, contar algo que aconteceu ou que vai acontecer, certo ou errado.

10.2 Métodos Utilizados para Denúncia

Existem várias formas de fazer uma denúncia tradicionalmente, são elas:

Verbal: Simplesmente fala-se a outrem o acontecido, não relatando em papel ou outro meio de armazenamento de voz;

Verbal / Escrita: O denunciante fala o ocorrido e uma segunda pessoa escuta e anota tudo no momento da abertura da ocorrência. Fazer um relato na delegacia de polícia, de um assalto, por exemplo;

Escrita: O denunciante escreve uma carta ou utiliza um papel próprio para o ato, relatando de forma narrativa os fatos ocorridos ou que estão por vir;

Verbal / Telefone: O denunciante liga para um número de telefone, que geralmente é próprio para denúncia e também relata o ocorrido. Note-se que na maioria dos casos também pode-se utilizar deste meio para reclamação de produtos ou serviços;

Escrita / Internet: Uma forma que está começando a tomar forma para todas as pessoas, porém com um público considerável. Neste caso o denunciante deve estar conectado à Internet, entrar no sítio especial para denúncias e relatar os acontecimentos.

Todos estes métodos aplicam-se a entidades *Públicas* e *Privadas*. Alguns problemas entram em evidência, quando se necessita provar algo denunciado à justiça. Por exemplo, o método simplesmente **Verbal** passa a valer uma palavra contra a outra, colocando a decisão do caso na interpretação e decisão de um Juiz.

10.3 Questões Jurídicas

Empresas e usuários, na maioria dos casos não estão preocupados com as questões de cunho jurídico, que poderia resolver qualquer impasse rapidamente.

No ato da denúncia, é importante que o denunciante esteja ciente de que a sua denúncia, se infundada, pode retornar para ele em forma de ações jurídicas.

Abaixo apresentam-se trechos do Código Penal Brasileiro que mostra as penas cabíveis para cada delito cometido através de denúncia:

Capítulo III

DOS CRIMES CONTRA A ADMINISTRAÇÃO DA JUSTIÇA

Denunciação caluniosa

Art. 339. - Dar causa a instauração de investigação policial ou de processo judicial contra alguém, imputando-lhe crime de que o sabe inocente:

Pena - reclusão, de 2 (dois) a 8 (oito) anos, e multa.

1º - A pena é aumentada de sexta parte, se o agente se serve de anonimato ou de nome suposto.

2º - A pena é diminuída de metade, se a imputação é de prática de contravenção.

Comunicação falsa de crime ou de contravenção.

Art. 340. - Provocar a ação de autoridade, comunicando-lhe a ocorrência de crime ou de contravenção que sabe não se ter verificado:

Pena - detenção, de 1 (um) a 6 (seis) meses, ou multa.

Auto-acusação falsa

Art. 341. - Acusar-se, perante a autoridade, de crime inexistente ou praticado por outrem:

Pena - detenção, de 3 (três) meses a 2 (dois) anos, ou multa. Falso testemunho ou falsa perícia

Como se pode ver, as penas para o autor de um a denúncia infundada são duras e devem serem cumpridas conforme a lei.

10.4 Fluxo de uma Denúncia

Quando um usuário apresenta uma queixa na Polícia Civil, é aberto um Inquérito policial. Este Inquérito entra na fase de investigação que deve ser concluída em trinta dias segundo o Art 10. do “Do Inquérito Policial”, do Título II do Processo em Geral no Livro I do Código de Processo Penal [dJ 41a].

No momento em se que conclui o Inquérito policial, a autoridade policial remete-o ao Poder Judiciário, com vistas ao Ministério Público.

De posse do Inquérito policial, o representante do Ministério Público, que pode ser o Promotor de Justiça ou Procurador da República, poderá seguir algumas providências:

1. Se o inquérito puder ser resolvido, o representante do Ministério Público pedirá ao juiz a baixa do inquérito;

2. Se o delito contido no Inquérito Policial já estiver prescrito, o Ministério Público pedirá ao Juiz que seja decretada a extinção da punibilidade;

3. Se o Inquérito Policial não puder ser resolvido, o Ministério Público pedirá ao Juiz o arquivamento, segundo Art. 17 e Art. 18 do Código de Processo Penal [dJ 41a]. Ao apreciar o pedido de arquivamento, o Juiz poderá:

   (a) concordar com o pedido;

   (b) não concordar com o pedido, neste caso envia para arquivamento, caso o Promotor não concorde, ele mesmo pode oferecer a denúncia, como mostrado abaixo;

4. Se o Inquérito policial estiver formalmente perfeito, o representante do Ministério Público oferecerá Denúncia.

Art. 28 - Se o órgão do Ministério Público, ao invés de apresentar a denúncia, requerer o arquivamento de inquérito policial ou de quaisquer peças de informação, o juiz, no caso de considerar improcedentes as razões invocadas, fará remessa do inquérito ou peças de informação ao procurador-geral, e este oferecerá a denúncia, designará outro órgão do Ministério Público para oferecê-la, ou insistirá no pedido de arquivamento, ao qual só então estará o juiz obrigado a atender.

Um exemplo de denúncia pode ser visto a seguir segundo [dJ 41b]:

EXCELENTÍSSIMO SENHOR DOUTOR JUIZ DA.... VARA CRIMINAL DA COMARCA DE .....

O órgão do Ministério Público em exercício perante esse R. Juízo vem, pelo presente instrumento, oferecer denúncia contra....

(qualificação) ...., pelos fatos que passa a expor:

a) No dia.... o acusado trafegava com seu veículo pela rua...., no sentido...., desenvolvendo velocidade de 100 Km horários;

b) Ao passar pela .... (local do evento)...., o acusado avançou o sinal de tráfego, indo colher, em cheio,.... (vítima)...., acarretando-lhe ferimentos;

c) O acusado não parou para socorrer a vítima, evadindo-se do local.

Tendo em vista o exposto, o Ministério Público pede que, recebida a denúncia, seja o réu citado e, a final, condenado como incurso no Art. 129, §§ 6 e 7, do Código Penal. Local

Data e assinatura do promotor.

Rol de testemunhas:

10.5 Conclusão

O ponto central deste capítulo foi apresentar de uma forma simples, como acontece um processo de denúncia, visto que será utilizado para a confecção do protótipo do sistema que utiliza um dos protocolos propostos.

Viu-se também a questão de nomenclatura que envolve uma denúncia e os prazos pré-estabelecidos no Código de Processo Penal.

Capítulo 11

Detalhes sobre a Implementação

11.1 Introdução

Com intuito de demonstrar o funcionamento do protocolo SDAS (Sistema de Denúncia Anônima Segura), foi implementado um protótipo para viabilidade prática utilizando o protocolo apresenta na seção 9.7. Escolheu-se este protocolo por apresentar a característica de anonimato e também por ser implementado a título de exemplificação.

Este protótipo utiliza criptossistema de chave pública de acordo com T. ElGamal [ELG 85]. A aplicação também utiliza a técnica de rede de misturadores [CHA 81] com gerador de anonimato.

11.2 Descrição

O Sistema de Denúncia Anônima Segura, é um sistema que trata as denúncias que são feitas através da internet, utilizando certificados digitais e criptografia. O Sistema foi dividido em duas partes:

1. Acesso para usuários que irão cadastrar as denúncias;
2. Acesso para os administradores do sistema (entidade que administra e mantém o sistema).

11.2.1 Descrição do Acesso para Usuários

Para os usuários acessarem o sistema, é necessário possuir um certificado digital para que se possa cadastrar a denúncia, utilizando assinatura digital. Ao acessar a página do sistema, o usuário visualizará a tela inicial, mostrada na figura 11.1, com as informações de como utilizar o sistema e como fazer para adquirir um certificado digital.

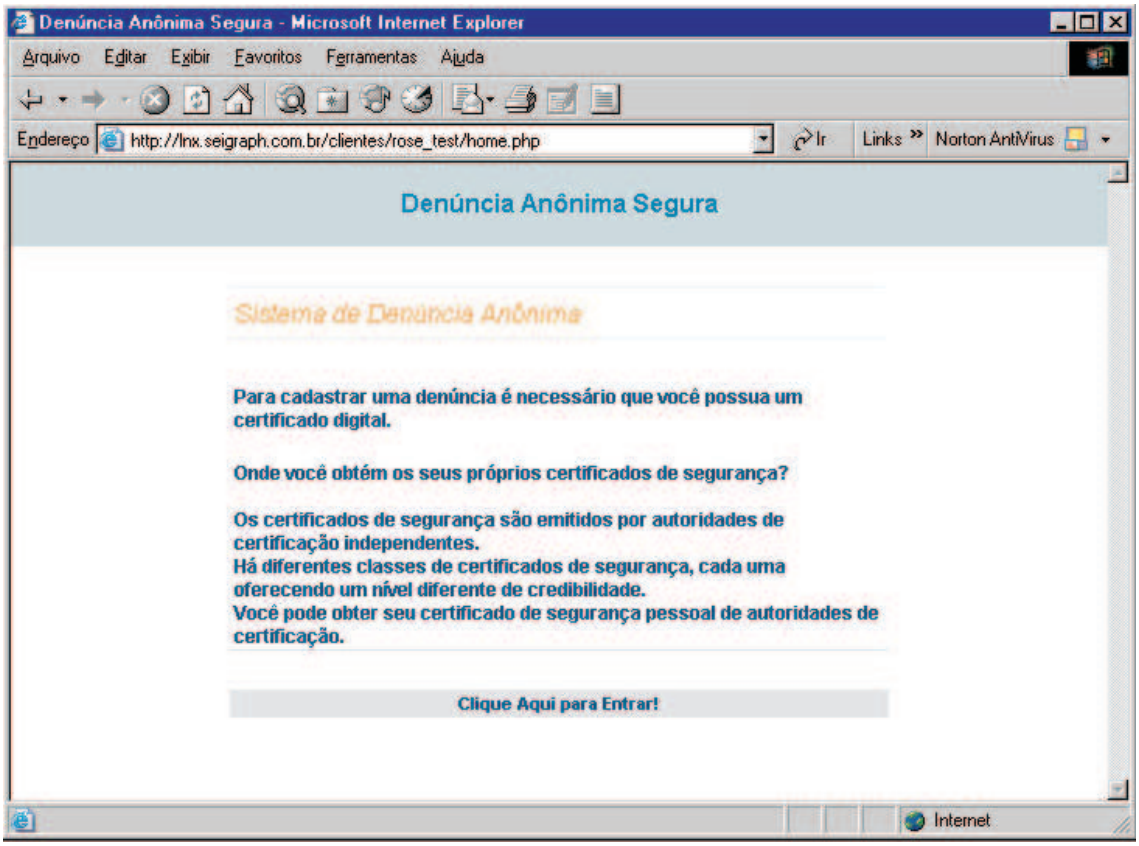

Figura 11.1: Tela Inicial: Tela inicial e informações importantes.

Após verificar as informações da tela inicial, o usuário passará para a tela que faz a autenticação do certificado digital, conforme a figura 11.2.

Caso a autenticação seja feita com sucesso, o usuário terá acesso ao cadastro e consulta das denúncias.

A tela de cadastro da denúncia, mostrada na figura 11.3, exige o preenchimento de todos os dados para que a entidade administradora possa averiguar os fatos

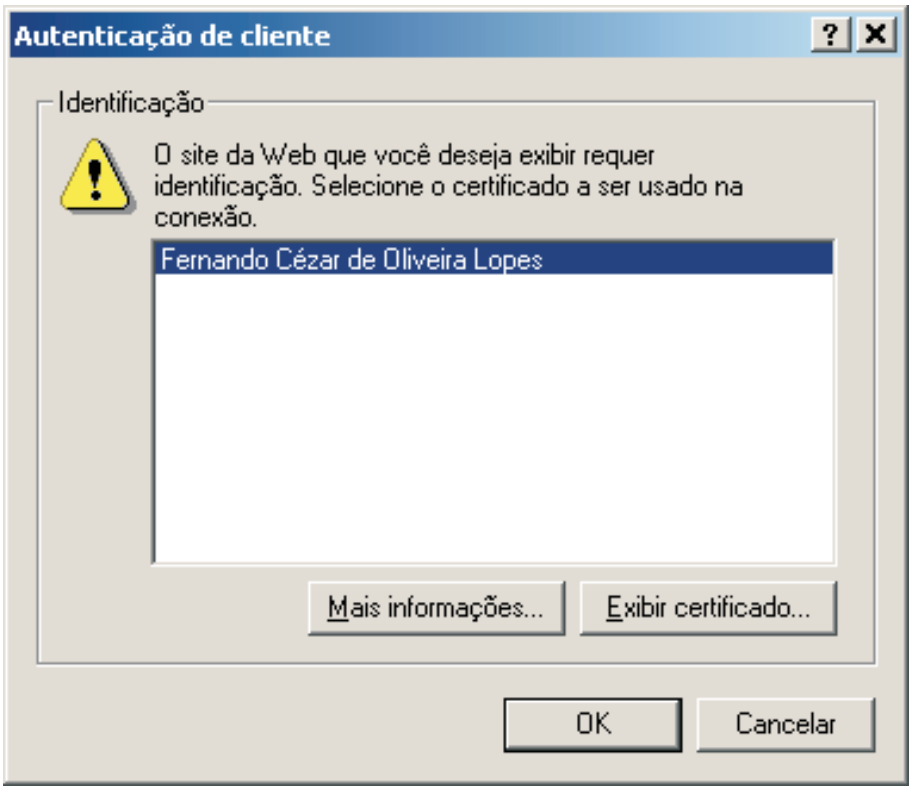

Figura 11.2: Seleção de Certificado: Tela para a seleção de certificado do usuário.

e dar continuidade ao processo.

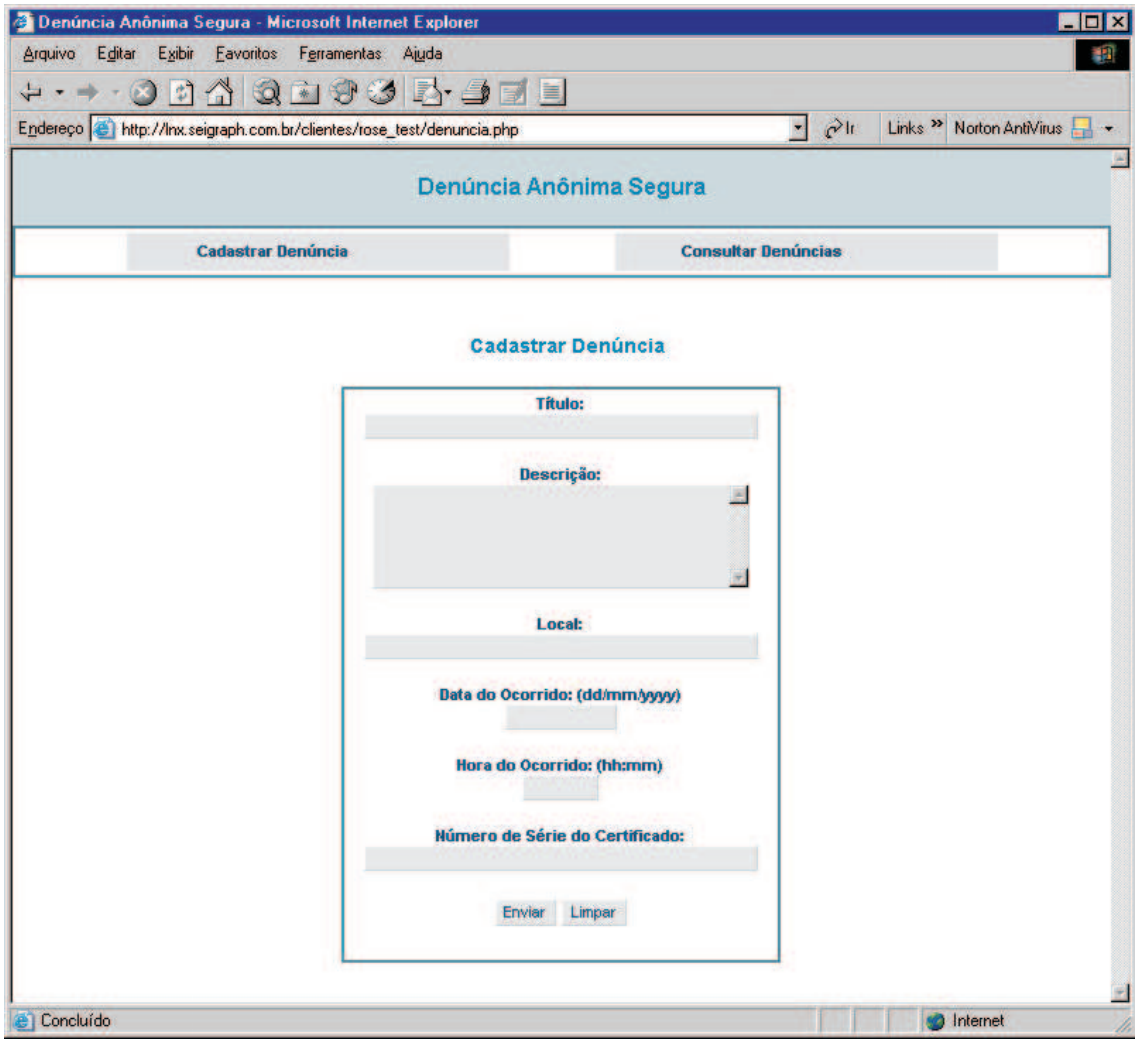

Figura 11.3: Cadastro da Denúncia: Tela para cadastrar os dados pertinentes à denúncia.

Após o cadastro da denúncia, os dados são enviados para os seguintes processos:

- Todo texto é transformado em número para fazer a cifragem dos dados;
- Os textos são cifrados;
- A assinatura digital é gerada para o caso de quebra de anonimato;
- É aplicada uma função resumo na denúncia para enviar para a datação utilizando um PDDE;

- Este resumo é cifrado novamente, utilizando criptografia temporal para limitar o tempo para a quebra de anonimato;
- São geradas as chaves para os administradores do sistema e enviado um correio eletrônico para cada administrador constando o texto cifrado e a chave para a quebra do anonimato (lembrando que isso somente será possível se for feito na data estipulada e quando todos os administradores estiverem em cooperação).

Além da tela de cadastro, o usuário tem acesso também a uma tela com a listagem das denúncias que foram publicadas, conforme figura 11.4. Uma denúncia só pode ser publicada por um administrador do sistema.

O fluxograma de como funciona o sistema para acesso dos usuários é mostrado na figura 11.5.

11.2.2 Descrição do Acesso para Administradores

Os administradores do sistema também devem possuir certificado digital para acessar o sistema, porém, além da autenticação do certificado, é necessário passar por uma tela que solicita o login de identificação e a senha do administrador para registrar um log das ações efetuadas no sistema, conforme a figura 11.6.

Após a autenticação com login e senha, o administrador possui acesso as seguintes telas:

- Publicação das denúncias;
- Quebra do Anonimato.

Para publicar a denúncia, é necessário preencher o parecer do Juiz referente a denúncia e publicar, conforme figura 11.7. Neste momento todos os usuários que possuem certificado digital poderão ter acesso às denúncias publicadas.

Para proceder a quebra do anonimato, é necessário que todos os administradores, que receberam o correio eletrônico com as chaves, entrem no sistema, preencham o texto cifrado e a chave para verificar a identificação do usuário que cadastrou a denúncia, conforme a figura 11.8.

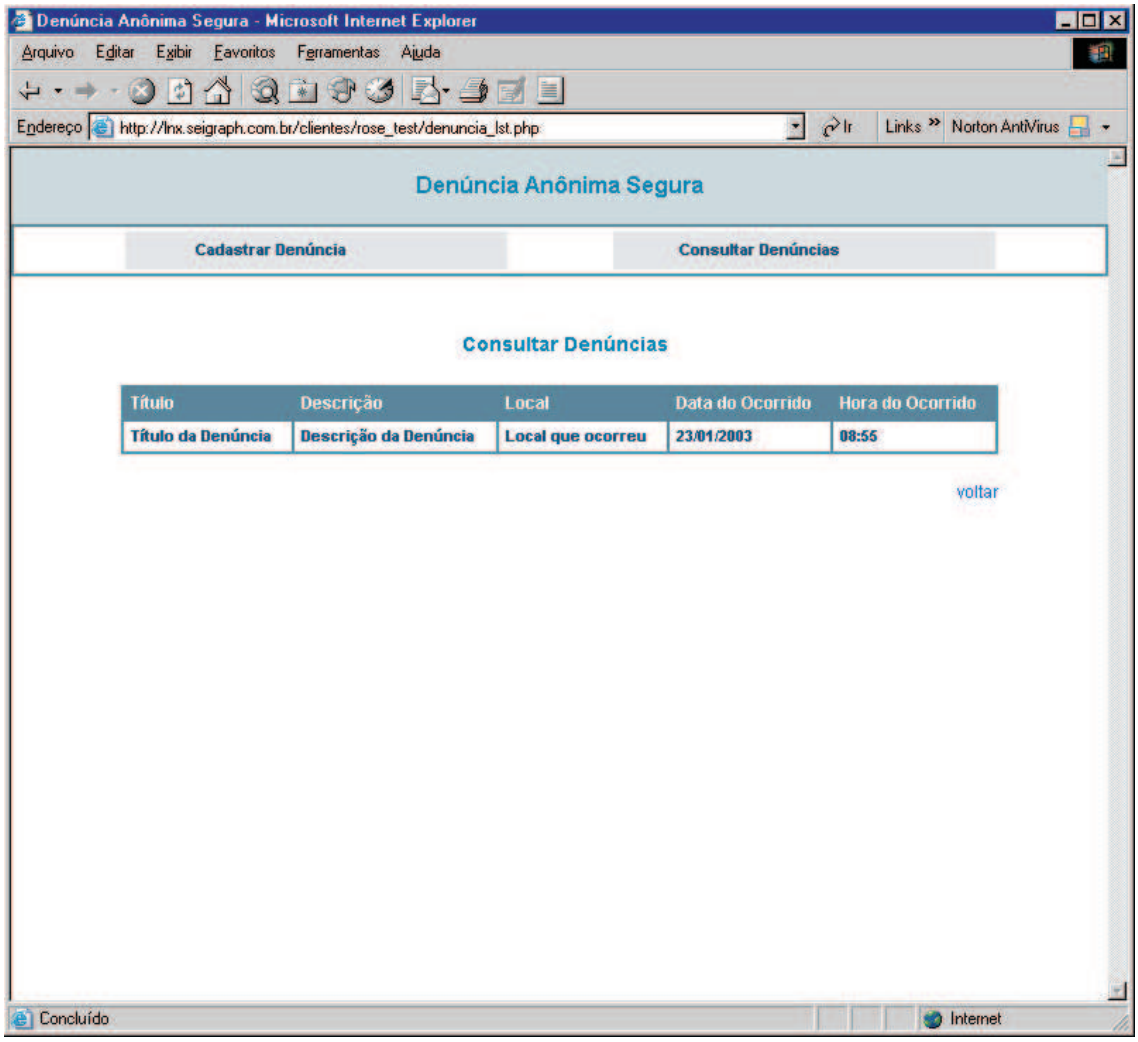

Figura 11.4: Listagem das Denúncias Publicadas: Tela para acompanhar a denúncia.

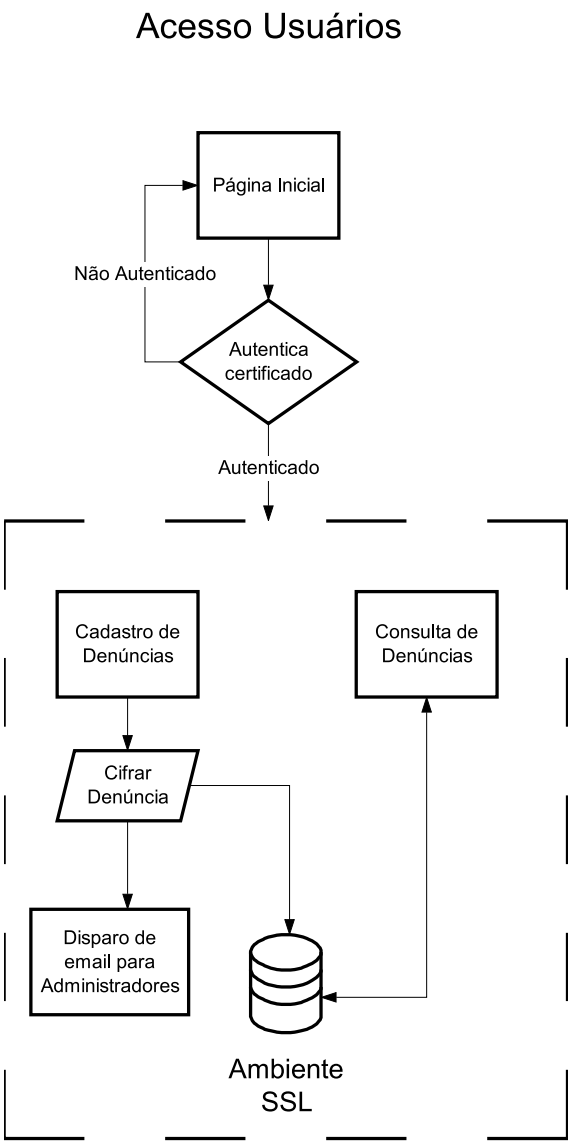

Figura 11.5: Fluxograma de Funcionamento para Usuário: Fluxo para os usuários.

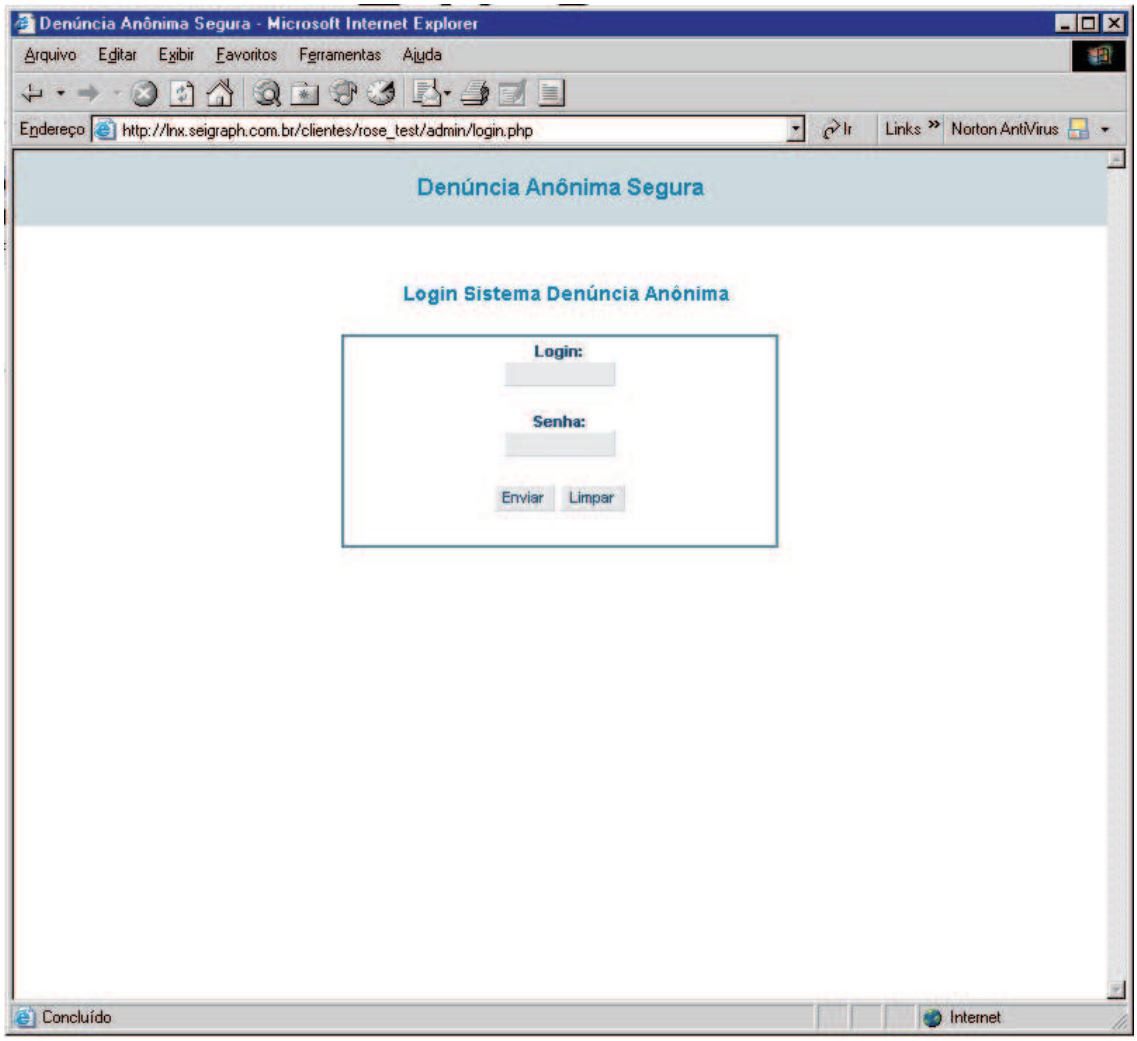

Figura 11.6: Tela de Acesso aos Administradores: Tela de login dos administradores.

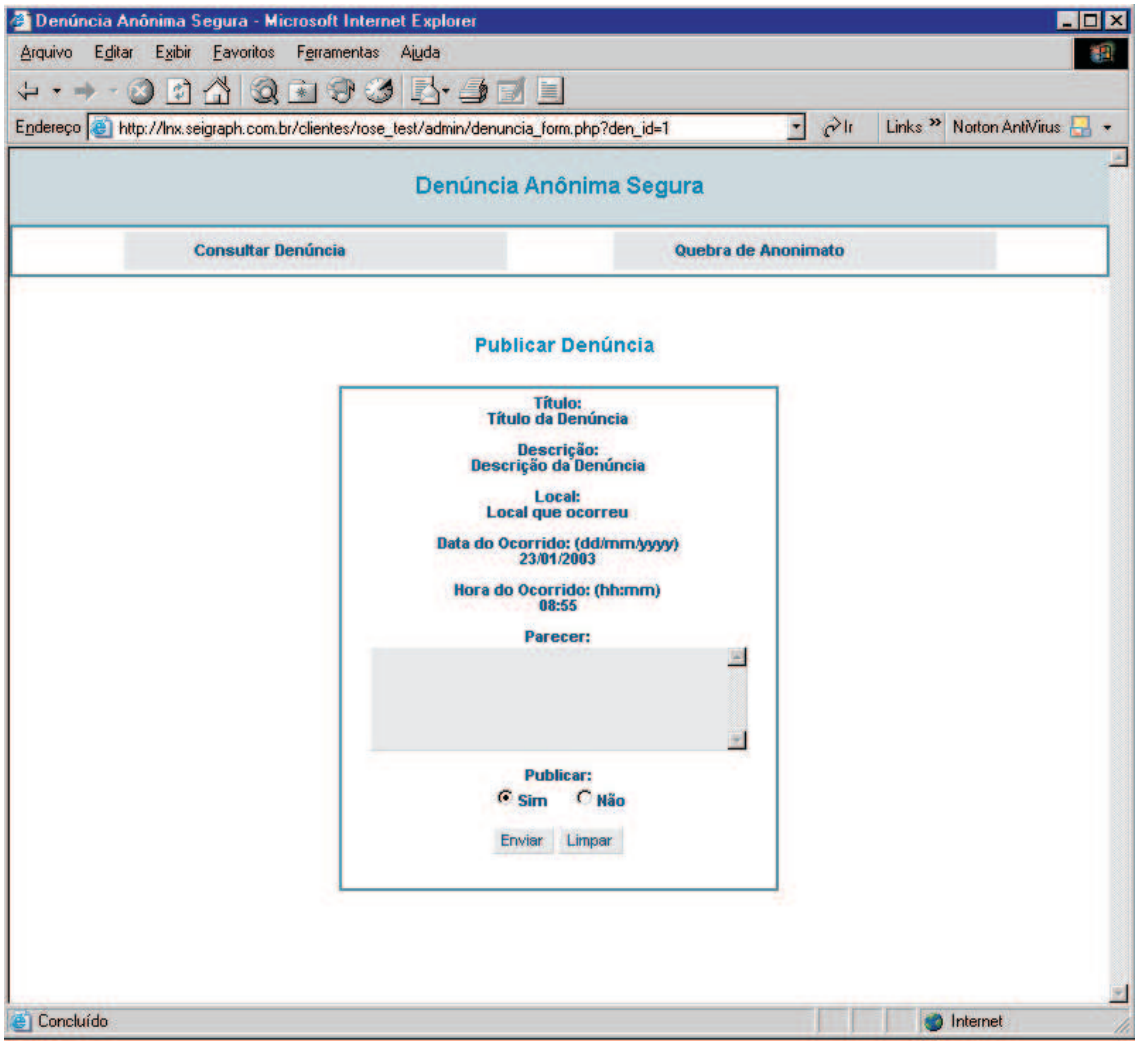

Figura 11.7: Tela de Publicação: Tela de publicação dos administradores.

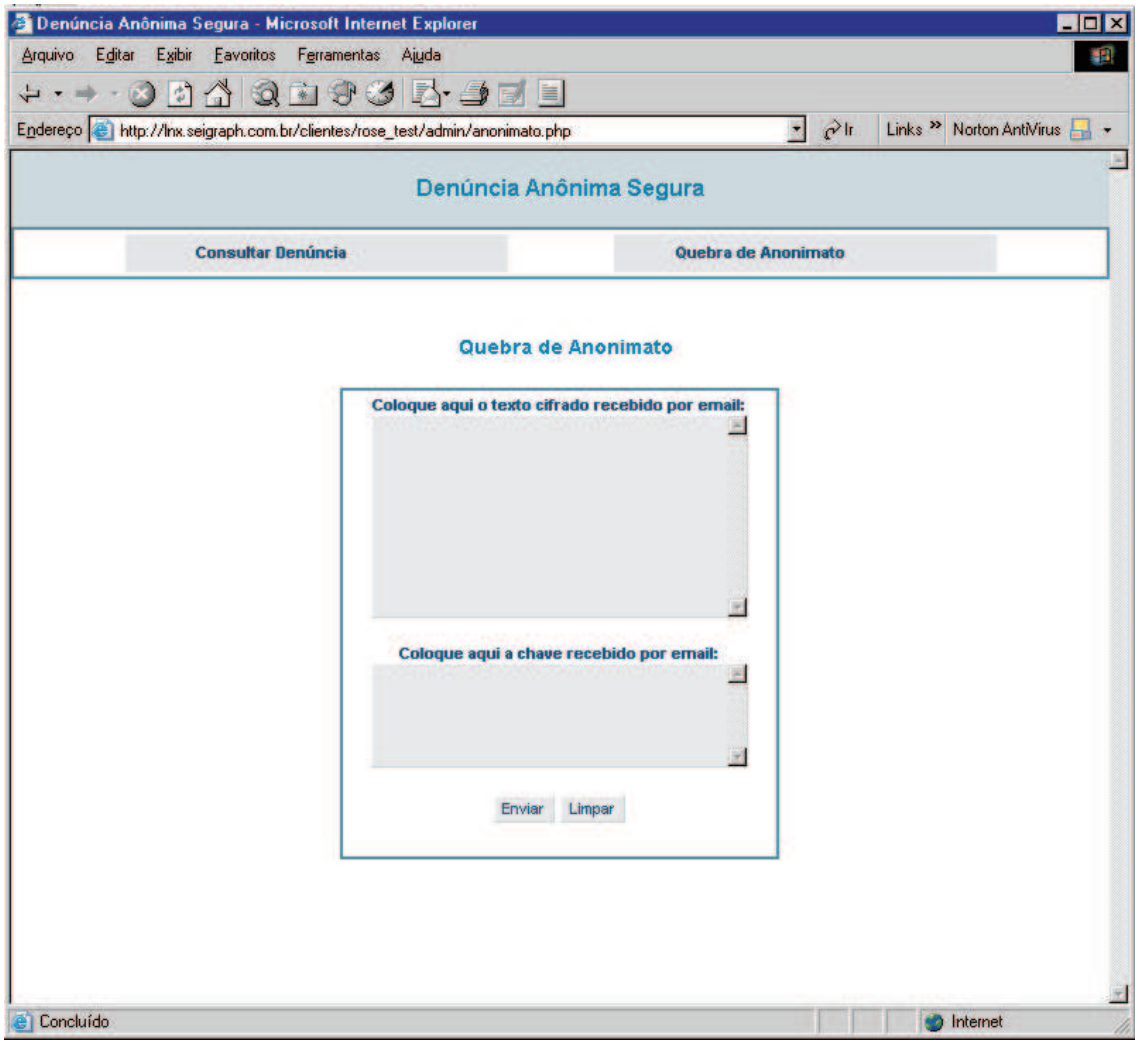

Figura 11.8: Quebra de Anonimato: Tela de quebra de anonimato dos administradores.

O fluxograma que mostra o funcionamento para os administradores é apresentado na figura 11.9.

11.3 Conclusão

Com esta implementação, verificou-se que a utilização das técnicas para programação segura requer um conhecimento de linguagem de programação bem aprimorado. Este fato determinou a utilização do terceiro protocolo apresentado, que utiliza uma rede de misturadores para a geração de anonimato.

Com a confecção dos módulos que compõem o SDAS na sua versão mais simplificada, adquiriu-se também experiência de implementação de um sistema seguro.

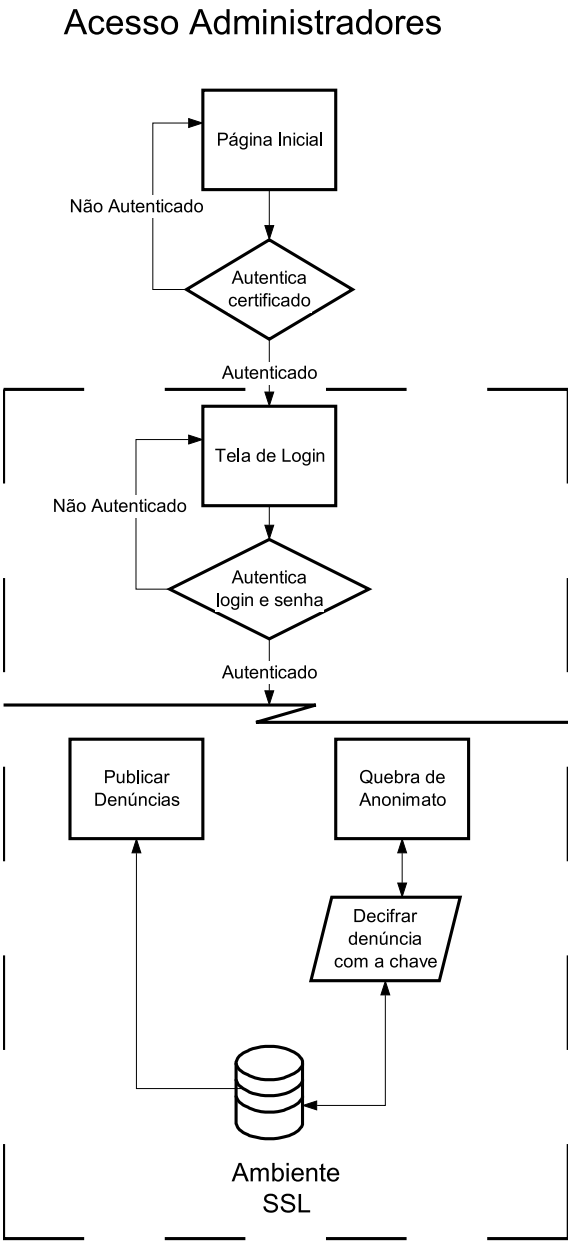

Figura 11.9: Fluxograma de Funcionamento para Administradores: Fluxo para os administradores.

Capítulo 12

Considerações Finais

O principal objetivo deste trabalho foi propor uma solução para a emissão de um documento eletrônico, de tal forma que somente em um período de tempo pré-determinado no futuro a identidade do emissor pudesse ser revelada.

Uma possível aplicação para esta solução é a implementação de um sistema de denúncia anônima usando a Internet. De fato, o problema acima surgiu durante a discussão de como implementar este sistema.

Apesar do tema estar em geral ligado a órgãos de justiça dos governos, ainda assim, existem várias aplicações passíveis de atender à iniciativa privada e organizações não governamentais.

Foi feita uma intensa pesquisa bibliográfica com vistas a encontrar algum trabalho similar ou relacionado ao problema acima. Não se encontrou qualquer trabalho na literatura.

De forma a poder avaliar as possíveis soluções propostas, foi construída uma lista de requisitos de segurança que os protocolos deveriam atender. Esta lista traça aspectos primordiais para se atingir o objetivo geral do trabalho.

Além do objetivo geral acima, podem-se listar várias etapas que foram cumpridas no decorrer deste trabalho como objetivos específicos.

Na abordagem inicial de como são utilizados os métodos de autenticação existentes hoje, viu-se que ainda confiam-se muito em sistemas que utilizam somente de

senha para assegurar a correta identificação do indivíduo. Os estudos feitos sobre as medidas biométricas, como fonte de autenticação, mostrou que está recebendo contribuições técnicas a cada dia, tornando-se melhor e mais precisa.

As questões jurídicas que envolvem as comunicações anônimas foram discorridas visando apresentar as citações encontradas na Constituição Federal [BRA 88], o Código Civil e o Penal, bem como pareceres de autoridades do judiciário. Como a questão é contraditória, caberia um estudo aprofundado de repercussão nacional.

Diferenciar anonimato e privacidade, e como estes se correlacionam, foi atendido, levantando-se as definições e questões relativas à aplicação de cada termo em uma determinada situação.

Através do relato sucinto de aplicações, pode-se identificar as que necessitam de comunicações anônimas como ponto base do seu funcionamento.

O estudo de protocolos criptográficos, que garantem comunicações anônimas, foi realizado através da descrição de forma concisa do seu funcionamento e seus relacionamentos relevantes.

Verificar como deve ocorrer uma comunicação entre um grupo fechado numa rede de comunicação de dados e, mesmo assim, garantir o anonimato das mensagens, foi apresentado no capítulo 8 sobre comunicação em grupo. Ali apresentam-se as definições sobre esta comunicação, e uma descrição breve de quatro protocolos.

A proposta de um protocolo criptográfico que atenda ao objetivo geral, detalhado na seção 9.10 dos protocolos criptográficos propostos, deu-se seguindo certa ordem de construção. Começou-se com um protocolo básico, até chegar no protocolo final que atendia a todos os objetivos e requisitos de segurança.

As análises foram efetuadas segundo os requisitos de segurança propostos na seção 9.3.

Levantar a lista dos requisitos de segurança necessários para que haja uma comunicação anônima com revelação no futuro da identidade, fazia parte direta da busca do protocolo final. Ela está representada na seção 9.3, e foram analisados segundo a sua presença ou não nos protocolos criptográficos propostos.

Implementar uma aplicação que se baseia em um dos protocolos pro-

postos, está descrita no capítulo 11. Foi implementado o terceiro protocolo como base de comportamento prático. Este protocolo atende a vários requisitos de segurança listados, e portanto consagra-se um bom exemplo de aplicação.

Apresentam-se a seguir algumas contribuições identificadas no decorrer deste trabalho e que são cabíveis de relevância, podendo auxiliar outrem, no desenvolvimento de outros trabalhos:

- Definição de um problema inédito relacionado à comunicação anônima;
- Levantamento dos requisitos de segurança necessários que o objetivo geral deveria atender;
- Desenvolvimento de um sistema para Internet de Denúncia Anônima Segura - DAS;
- Proposição de seis protocolos criptográficos e análise de segurança em relação aos requisitos pré-definidos;
- A proposta de um *Bloco Inverso de Shamir*, responsável pela resolução do quebra-cabeça temporal, comparações entre partes de segredo, e a destruição das partes do quebra-cabeça iguais.
- Estudo e apresentação de tecnologias como compartilhamento de segredo, criptografia temporal, comunicação em grupo e anonimato em um mesmo documento e fazendo parte de um mesmo protocolo.
- Visando o melhor entendimento da tecnologia e também ilustrar o seu funcionamento, foi desenvolvido um exemplo numérico de criptografia temporal usando ElGamal [ELG 85]. Este exemplo se deu através da construção e resolução do quebra-cabeça.

12.1 Trabalhos Futuros

No decorrer desta dissertação, várias idéias para desenvolvimentos futuros foram encontradas:

1. Cabe um estudo mais profundo da legislação vigente do país e também para comunicações através da Internet;

2. O estudo das implicações sociais sobre comunicação anônima no Brasil;

3. Especificação e análise formal dos protocolos criptográficos propostos;

4. Implantação prática do Bloco Inverso de Shamir, conforme proposto no trabalho;

5. Levantar os requisitos necessários para a implementação do sistema SDAS junto a órgãos governamentais;

6. Efetuar a implementação do sistema DAS em repartições jurídicas e delegacia de polícia, com o intuito de avaliar dados práticos das funcionalidades;

7. Implantar o sistema SDAS em empresas públicas ou privadas, como um meio de recebimento de reclamações, internas ou externas;

8. Utilização do sistema SDAS junto com o protocolo proposto por Amauri Sant´Anna Ghisleri [GHI 02], Sistema Seguro de Atendimento ao Cliente Garantia da Qualidade de Serviço, dentro de uma empresa atenderia alguns casos de reclamação dos clientes;

9. João Luiz Francalacci Rochati [ROC 01], no seu trabalho sugere a utilização de um sistema de denúncia que atenda a alguns requisitos. Nota-se que o sistema SDAS atende a estes requisitos podendo fornecer este suporte para a denúncia de pirataria de software;

10. Melhorias no sistema SDAS, incluindo mais funcionalidade, ou até a integração com plataformas comerciais existentes;

Referências Bibliográficas

[AP 87] A. PFITZMANN, M. W. Networks without user observability. **Computer Security**, [S.l.], v.6/2, p.158–166, 1987.

[AP 01] A. PFITZMANN, M. K. Anonymity, unobservability, and pseudonymity - a proposal for terminology. http://123.koehntopp.de/marit/pub/anon/Anon-Terminology.txt, Junho, 2001.

[BRA 88] BRASIL, Constituição (1988). **Constituição da República Federativa do Brasil:** promulgada em 5 de outubro de 1988. Organização do texto: Juarez de Oliveira. 4. ed. São Paulo: Saraiva, 1990. 168 p. (Série Legislação Brasileira).

[CAM 97] CAMENISH, J. Efficient and generalized group signatures. **Advances in Cryptology - EUROCRYPT'97, LNCS**, [S.l.], v.1233, 1997.

[CG 96] CEKI GÜLCÜ, G. T. Mixing email with babel. In: INTERNET SOCIETY SYMPOSIUM ON NETWORK AND DISTIBUTED SYSTEM SECURITY, 1996. **Proceedings...** San Diego, CA:IEEE, 1996. p.2–16.

[CHA 81] CHAUM, D. Untraceable electronic mail, return address and digital pseudonyms. **Communications of the ACM**, [S.l.], v.24, p.84–88, 1981.

[CHA 85] CHAUM, D. Security without identification: transaction systems to make big brother obsolete. **Communications of the ACM**, [S.l.], v.28, p.1030–1044, 1985.

[CHA 91] CHAUM, D. Group signatures. **Lecture Notes in Computer Science - Springer-Verlag**, [S.l.], v.547, p.257–265, 1991.

[CHE 95] CHEN, L. New group signature schemes. **Lecture Notes in Computer Science - Springer-Verlag**, [S.l.], v.950, p.171–181, 1995.

[D'A 00] D'ANDRÉA, E. R. P. **Segurança em banco eletrônico**. Pricewaterhouse-Coopers, 1. ed., 2000.

[dC 00] DE CARVALHO, D. B. **Segurança de Dados Com Criptografia**. RJ:Editora Book Express Ltda, 2000.

[dHF 89] DE HOLANDA FERREIRA, A. B. **Minidicionário da Língua Portuguesa**. Editora Nova Fronteira, 2. ed., 1989.

[dJ 41a] DE JUSTIÇA, T. Código de processo penal. OUTUBRO, 1941. DECRETO-LEI N.º 3.689.

[dJ 41b] DE JUSTIÇA, T. Código de processo penal. No sitío da DJI - Direito & Justiça Informática Ltda., OUTUBRO, 1941. DECRETO-LEI N.º 3.689.

[dSB 72] DA SILVEIRA BUENO, F. **Novíssimo dicionário brasileiro da língua portuguesa**. LISA - Livros Irradiantes S.A., 1. ed., 1972.

[EDW 99] Edward, J., editor. **Many-to-one Cryptographic Algorithms and Group Signatures (Extended Abstract)**, v.6, New Zealand - Auckland, January, 1999. Australian Computer Science Communications,Proceedings of the Twenty Second Australasian Computer Science Conference (ACSC'99).

[ELG 85] ELGAMAL, T. A public key cryptosystem and signature schema based on discrete logarithms. **IEEE Transactions on Information Theory**, [S.l.], v.31, p.473–481, 1985.

[FS 99] FRANK STAJANO, R. A. The cocaine auction protocol: On the power of anonymous broadcast. In: PROCEEDINGS OF INFORMATION HIDING WORKSHOP 1999, 1999. **Proceedings...** Dresden, Germany:Springer-Verlag Berlin Heidelberg, 1999. p.01–14.

[GAR 99] GARFINKEL, S. **Comércio e Segurança na WEB**. Editora Market Books do Brasil Ltda, 1. ed., 1999.

[GHI 02] GHISLERI, A. S. **Sistema seguro de atendimento ao cliente garantia da qualidade de serviço.** Florianópolis - SC:UFSC - Universidade Federal de Santa Catarina, Março, 2002. Mestrado.

[IT 00] ITU-T. The directory - authentication framework. 2000. Recommendation x509.

[JF 99] J. FEGHII, P. W. **Digital Certificates - Applied Internet Security**. Addison Wesley Longman, 1. ed., 1999.

[LC 95] L. CHEN, T. P. P. On the efficiency of group signatures providing information-theoretic anonymity. **International Conference on the Theory and Application of Cryptographic Techniques Advances in Cryptology – EUROCRYPT '95, Springer-Verlag**, Saint-Malo, France, v.921, n.5, p.39–49, 1995.

[LIU 01] LIU, S. A pratical guide to biometric security technology. **IEEE IT Pro**, [S.l.], v.1, p.27–32, 2001.

[LTD 00] LTDA., A. M. S. **Dicionário Michaelis da língua brasileira**. Amigo Mouse Software Ltda., 1. ed., 2000.

[MAY 93] MAY, T. C. Timed-release crypto. Disponível em: http://cypherpunks.venona.com/date/1993/02/index.html, Fevereiro, 1993.

[MEN 96] MENEZES, A.; OORSCHOT, P. V. **Handbook of Applied Cryptography**. California:CRC Press, 1996.

[MGR 98] MICHAEL G. REED, PAUL F. SYVERSON, D. M. G. Anonymous connections and onion routing. **IEEE Journal on selected areas in Communications**, [S.l.], v.16, p.482–494, 1998.

[MJ 99] MARKUS JAKOBSSON, A. J. Millimix: Mixing in small batches. 1999. Dimacs technical report 99-33.

[MJ 02] MARKUS JAKOBSSON, A. JUELS, R. L. R. Making mix nets robust for electronic voting by randomized partial checking. **Communications of the ACM**, [S.l.], v.24, p.84–88, 2002.

[MKR 98] MICHAEL K. REITER, A. D. R. Crowds: Anonymity for web transactions. **ACM Transactions on Information and System Security**, [S.l.], v.1, p.66–92, 1998.

[MO 00] M. OHKUBO, M. A. A length-invariant hybrid mix. **ASIACRYPT '00**, [S.l.], v.191, p.178, 2000.

[NIS 97] NIST. **An Introduction to Computer Security, The NIST Hand Book**. CRC Press, 1997.

[NOT 02] NOTOYA, A. E. **IARSDE - infra estrutura de armazenamento e recuperação segura de documentos eletrônicos: Validade do documento eletrônico por tempo indeterminado.** Florianópolis - SC: UFSC - Universidade Federal de Santa Catarina, Março, 2002. Mestrado.

[PER 03] PEREIRA, F. C. **Protocolo criptográfico para selar e lacrar documentos eletrônicos aplicados a sistemas de compras seguro.** Florianópolis - SC: UFSC - Universidade Federal de Santa Catarina, Março, 2003. Dissertação de Mestrado.

[RIB 03] RIBEIRO, P. S. **Um protocolo criptográfico para comunicação anônima segura em grupo.** Florianópolis - SC: UFSC - Universidade Federal de Santa Catarina, Fevereiro, 2003. Dissertação.

[RIV 78] RIVEST, R.; SHAMIR, A. A method for obtaining digital signatures and public key cryptosystems. **Communications of the ACM**, [S.l.], v.28, February, 1978.

[RIV 95] RIVEST, R. L. The RC5 encryption algorithm. **Dr. Dobbs Journal**, [S.l.], v.226, 1995.

[RIV 99] RIVEST, R. L. Description of the LCS35 time capsule crypto-puzzle, April 4, 1999.

[RLR 96] RIVEST, Ronald L.; SHAMIR, Adi; WAGNER, David A. Time-lock puzzles and timed-release crypto, Setembro, 1996. Disponível em: http://theory.lcs.mit.edu/řivest/RivestShamirWagner-timelock.ps.

[ROC 01] ROCHATI, J. L. F. **Proteção de software por certificação digital.** Florianópolis - SC:UFSC - Universidade Federal de Santa Catarina, Dezembro, 2001. Mestrado.

[SCH 96] SCHNEIER, B. **Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition**. John Wiley and Sons, 2. ed., 1996.

[SCH 01] SCHNEIER, B. **Secrets and Lies, 1nd Edition**. John Wiley and Sons, 1. ed., 2001.

[SHA 79] SHAMIR, A. How to share a secret. **Communication of the ACM**, [S.l.], v.22, n.11, p.612–613, November, 1979.

[SMI 02] SMITH, R. E. **Authentication: from password to public keys**. Addison Wesley, 1. ed., 2002.

[SP 97] S. PARK, S. K. D. W. ID-based group signature. **Eletronic Letter**, [S.l.], v.33, p.1616 - 1617, 1997.

[STA 98] STALLINGS, W. **Cryptography and Network Security**. Prentice Hall, 2. ed., 1998.

[STI 95] STINSON, D. R. **Cryptography : Theory and Practice**. CRC Press, 1995.

[TER 00] TERADA, R. **Segurança de Dados - Criptografia em Redes de Computador**. SP:Editora Edgard Blücher Ltda, 2000.

[WO 97] WAKAHA OGATA, K. K. Fault tolerant anonymous channel. In: LECTURE NOTE IN COMPUTER SCIENCE, 1997. **Proceedings...** ,:ICICS'97, 1997. p.440–444.

[YMT 99] YUH-MIN TSENG, J. K. J. A novel ID-based group signature. **Information Sciences - Elsevier**, [S.l.], v.120, p.131–141, May, 1999.

Apêndice A

Implementação de uma aplicação com algoritmo de ElGamal

Este anexo consiste na implementação de uma urna de Votação Eletrônica para demonstrar o funcionamento de uma Rede de Misturadores, em linguagem de programação ANSI C++.

Votação Eletrônica

O algoritmo de ElGamal é um algoritmo de criptografia assimétrica. Como tal, possui uma chave pública e uma chave privada. A chave pública é constituída de três campos: y, g e p. A chave privada é constituída por um número único. Assim, os seguintes tipos foram definidos para armazenar as chaves de ElGamal:

```
typedef UINT32 ChavePrivada;
typedef struct {
  UINT32 y;
  UINT32 g;
  UINT32 p;
} ChavePublica;
```

Para criar estas chaves, uma classe chamada ParChaves foi implementada. Uma instância desta classe com os argumentos apropriados cria a chave pública. A assinatura e a implementação do construtor da classe ParChaves é mostrada abaixo:

```
ParChaves(UINT32 p, UINT32 g, ChavePrivada x);
```

```
ParChaves::ParChaves(UINT32 p, UINT32 g, ChavePrivada x) {
  if ((teste_primo(p)) && (pertence_campo(p, g)) && (x > 0) && (x < (p-2)))
  {
    chavePrivada = x;
    chavePublica = (ChavePublica *)malloc(sizeof(ChavePublica));
    chavePublica->g = g;
    chavePublica->y = x_pow_y_mod_p(g, x, p);
    chavePublica->p = p;
  }
  else
  {
    chavePrivada = 0;
    chavePublica = NULL;
  }
}
```

Foi criada uma classe chamada ElGamalCifrador que utiliza as chaves criadas pela classe ParChaves para cifrar e decifrar um texto plano. A interface da classe ElGamalCifrador é mostrada abaixo:

```
class ElGamalCifrador {
  public:
    ElGamalCifrador();
    void cifre(ChavePublica *chavePublica, TextoPlano textoPlano, TextoCifrado *textoCifrado);
    void re_cifre(ChavePublica *chavePublica, TextoCifrado *textoCifrado);
    BOOL decifre(ChavePublica *chavePublica, ChavePrivada chavePrivada, TextoCifrado *textoCifrado,
    TextoPlano *textoPlano);
};
```

A implementação dos métodos cifre, recifre e decifre é mostrada a seguir:

```
void ElGamalCifrador::cifre(ChavePublica *chavePublica, TextoPlano
textoPlano, TextoCifrado *textoCifrado) {
  int r;
  if (chavePublica != NULL) {
      r = (rand() % (chavePublica->p - 3)) + 2;
    // calcula x^y mod p
    textoCifrado->a = x_pow_y_mod_p(chavePublica->g, r, chavePublica->p);
    // calcula m*x^y mod p
    textoCifrado->b = z_times_x_pow_y_mod_p(textoPlano, chavePublica->y, r, chavePublica->p);
  }
}
void ElGamalCifrador::re_cifre(ChavePublica *chavePublica,
TextoCifrado *textoCifrado) {
  int r;
  if (chavePublica != NULL) {
    r = (rand() % (chavePublica->p - 3)) + 2;
    textoCifrado->a = z_times_x_pow_y_mod_p(textoCifrado->a, chavePublica->g, r,
    chavePublica->p);
    textoCifrado->b = z_times_x_pow_y_mod_p(textoCifrado->b, chavePublica->y, r,
    chavePublica->p);
  }
}

BOOL ElGamalCifrador::decifre(ChavePublica *chavePublica,
ChavePrivada chavePrivada, TextoCifrado *textoCifrado, TextoPlano
*textoPlano) {
  BOOL ok = FALSE;
  UINT64 inverso;
  if (chavePrivada != 0) {
    if (inverso_multiplicativo(x_pow_y_mod_p(textoCifrado->a, chavePrivada, chavePublica->p),
    chavePublica->p, &inverso)) {
      *textoPlano = ((textoCifrado->b
       % chavePublica->p)*(inverso % chavePublica->p)) % chavePublica->p;
      ok = TRUE;
    }
  }
  return(ok);
}
```

Para simplificar a classe ElGamalCifrador, foram criadas algumas funções úteis para o cálculo de algumas expressões de aritmética modular. Estas funções são as seguintes:

```
UINT64 x_pow_y(UINT64 x, UINT64 y); BOOL teste_primo(UINT32 q);
UINT64 x_pow_y_mod_p(UINT64 x, UINT64 y, UINT64 p); UINT64
z_times_x_pow_y_mod_p(UINT64 z, UINT64 x, UINT64 y, UINT64 p);
BOOL inverso_multiplicativo(UINT64 b, UINT64 n, UINT64 *inverso);
BOOL relativamente_primos(UINT32 p, UINT32 g);
```

A.1 Implementação da rede de misturadores

Foi criada uma classe RedeMisturadores para implementar a rede baseada em [MJ 99]. Veja-se a interface da classe RedeMisturadores abaixo:

```
struct S_ElemListaUrna {
  TextoCifrado *texto;
  struct S_ElemListaUrna *prox;
};
typedef struct S_ElemListaUrna ElemListaUrna;

class RedeMisturadores {
  private:
    UINT8 votoMin;
    UINT8 votoMax;
    UINT8 numMisturadores;
    ElGamalCifrador *cifrador;
    ChavePublica *chavePublica;
    UINT32 tamUrnaMisturada;
    TextoCifrado **urnaMisturada;
    ElemListaUrna *urna;
    void depositeVoto();
    void misture();
    void compare(TextoCifrado *texto1, TextoCifrado *texto2);
  public:
    RedeMisturadores(ChavePublica *chavePublica, FILE *stream);
    RedeMisturadores(ChavePublica *chavePublica, UINT8 votoMin, UINT8 votoMax,
     UINT8 numMisturadores, UINT32 numElemUrnaMisturada);
    ~RedeMisturadores();
    void recebaVoto(TextoPlano voto);
    void escrevaCedulas(FILE *stream);
    void escrevaVotos(char *stream, ChavePrivada chave);
    void escrevaUrnas(char *stream);
    void escrevaResultado(char *stream, ChavePrivada chave);
};
```

A classe RedeMisturadores é constituída, dentre outras coisas, por uma urna onde os votos são misturados (atributo urnaMisturada) e por outra urna onde os votos são depositados (atributo urna). A urna de mistura possui um número fixo de votos (este número é informado na inicialização da rede) e, por essa razão, foi implementada em um vetor. Visto que o número de votantes não é pré-determinado, a urna de depósito é implementada em uma lista encadeada. A inicialização da rede é feita quando um objeto RedeMisturadores é instanciado. Mostra-se a implementação do construtor desta classe abaixo:

```
RedeMisturadores::RedeMisturadores(ChavePublica *chavePublica,
UINT8 votoMin, UINT8 votoMax, UINT8 numMisturadores, UINT32
numElemUrnaMisturada) {
  UINT8 numAlternativas = votoMax - votoMin + 1;
  INT32 i, j, k, n;
  this->votoMin = votoMin;
  this->votoMax = votoMax;
  this->chavePublica = chavePublica;
  this->numMisturadores = numMisturadores;
  this->tamUrnaMisturada = numElemUrnaMisturada - (numElemUrnaMisturada % numAlternativas);
  this->urna = NULL;
  this->urnaMisturada = (TextoCifrado **)malloc(sizeof(TextoCifrado *)*(tamUrnaMisturada+1));
  this->cifrador = new ElGamalCifrador();
  k = 0;
  n = tamUrnaMisturada/numAlternativas;
  for (i = 0; i < n; i++) {
    for (j = votoMin; j <= votoMax; j++) {
      this->urnaMisturada[k] = (TextoCifrado *)malloc(sizeof(TextoCifrado));
      cifrador->cifre(chavePublica, j, this->urnaMisturada[k]);
      k++;
    }
  }
  misture();
}
```

A inicialização da rede envolve preencher a urna de mistura com uma quantidade igual de votos para cada opção de voto (isto é, para cada candidato). Por esta razão, a inicialização da rede precisa saber qual é o intervalo dos números que constituirão cada voto. Quando a rede receber um voto, este voto deverá ser cifrado com o algoritmo de ElGamal e incluído na urna de mistura. A urna de mistura, então, é embaralhada e re-encriptada (método misture). Na seqüência, um voto aleatório será retirado dela. Este voto extraído da urna de mistura será acrescentado na urna de depósito. Este processo é implementado no método recebaVoto que é mostrado abaixo:

```
void RedeMisturadores::recebaVoto(TextoPlano voto) {
  tamUrnaMisturada++;
  urnaMisturada[tamUrnaMisturada-1] = (TextoCifrado *)malloc(sizeof(TextoCifrado));
  cifrador->cifre(chavePublica, voto, urnaMisturada[tamUrnaMisturada-1]);
  misture();
  depositeVoto();
  tamUrnaMisturada--;
}
O m{\'e}todo que efetivamente "embaralha" a urna de mistura {\'e} mostrado
a seguir:
void RedeMisturadores::misture() {
  UINT8 k, i, j;
  for (k = 0; k < numMisturadores; k++) {
    for (i = 0; i < tamUrnaMisturada-1; i++) {
      for (j = i+1; j < tamUrnaMisturada; j++) {
        compare(urnaMisturada[i], urnaMisturada[j]);
      }
    }
  }
}
```

Este método foi implementado baseado em [MJ 99], segundo ele, a mistura de um conjunto de encriptados com ElGamal envolve ordená-los em ordem crescente passando-os por um comparador. Este comparador é um dispositivo que recebe como entrada $(x1, x2)$ e retorna $(x1, x2) se x1 <= x2$ ou retorna $(x2, x1)$, caso contrário. Antes

de o comparador retornar o par ordenado, entretanto, o texto cifrado é re-encriptado com o algoritmo de ElGamal (isto é possível em função somente do texto cifrado e da chave pública). Este processo de ordenação deve ser feito por cada misturador da rede (em cadeia). Na implementação feita aqui, basta invocar o método misture tantas vezes quantas forem o número de misturadores. A classe RedeMisturadores provê o método escrevaVotos capaz de fornecer os votos (isto é, o texto plano) e o método escrevaResultado que provê o resultado da apuração dos votos, ambos em função da chave privada. Para isto, os votos acrescentados na inicialização da rede para fins de mistura deverão ser excluídos. A implementação destes métodos se encontra a seguir:

```
void RedeMisturadores::escrevaVotos(char *stream, ChavePrivada
chave) {
  INT32 n = (votoMax - votoMin + 1);
  INT32 k = tamUrnaMisturada/n;
  UINT32 *ocorrencias;
  TextoPlano texto;
  ElemListaUrna *elem;
  INT32 i;
  char buffer[16];
  ocorrencias = (UINT32 *)malloc(sizeof(UINT32)*n);
  for (i = 0; i < n; i++) {
    ocorrencias[i] = k;
  }
  // percorre a urna misturada
  for (i = 0; i < tamUrnaMisturada; i++) {
    cifrador->decifre(chavePublica, chave, urnaMisturada[i], &texto);
    // tenho que descontar os votos inclu{\'\i}dos na inicializa\c{c}{\~a}o da rede...
    if (ocorrencias[texto-votoMin] == 0) {
      sprintf(buffer, "%u ", texto);
      strcat(stream, buffer);
    }else
      ocorrencias[texto-votoMin]--;
  }
  // percorre a urna de votos (lista encadeada)
  elem = urna;
  while (elem != NULL) {
    cifrador->decifre(chavePublica, chave, elem->texto, &texto);
    // tenho que descontar os votos inclu{\'\i}dos na inicializa\c{c}{\~a}o da rede...
    if (ocorrencias[texto-votoMin] == 0) {
      sprintf(buffer, "%u ", texto);
      strcat(stream, buffer);
    }else
      ocorrencias[texto-votoMin]--;
    elem = elem->prox;
  }
  free(ocorrencias);
}

void RedeMisturadores::escrevaResultado(char *stream, ChavePrivada
chave) {
  INT32 n = (votoMax - votoMin + 1);
  INT32 k = tamUrnaMisturada/n;
  UINT32 *ocorrencias;
  UINT32 *numVotos;
  TextoPlano texto;
  ElemListaUrna *elem;
  INT32 i;
  char buffer[128];
  ocorrencias = (UINT32 *)malloc(sizeof(UINT32)*n);
  numVotos = (UINT32 *)malloc(sizeof(UINT32)*n);
  for (i = 0; i < n; i++) {
    ocorrencias[i] = k;
    numVotos[i] = 0;
  }
  // percorre a urna misturada
  for (i = 0; i < tamUrnaMisturada; i++) {
    cifrador->decifre(chavePublica, chave, urnaMisturada[i], &texto);
    // tenho que descontar os votos inclu{\'\i}dos na inicializa\c{c}{\~a}o da rede...
    if (ocorrencias[texto-votoMin] == 0)
```

```
      numVotos[texto-votoMin]++;
    else
      ocorrencias[texto-votoMin]--;
  }
  // percorre a urna de votos (lista encadeada)
  elem = urna;
  while (elem != NULL) {
    cifrador->decifre(chavePublica, chave, elem->texto, &texto);
    // tenho que descontar os votos inclu{\'\i}dos na inicializa\c{c}{\~a}o da rede...
    if (ocorrencias[texto-votoMin] == 0)
      numVotos[texto-votoMin]++;
    else
      ocorrencias[texto-votoMin]--;
    elem = elem->prox;
  }
  for (i = votoMin; i <= votoMax; i++) {
    sprintf(buffer, "Votos em %u: %u%c%c", i, numVotos[i-votoMin],13,10);
    strcat(stream, buffer);
  }
  free(ocorrencias);
  free(numVotos);
}
```

MixNet.cpp

Apresenta-se aqui a implementação do código principal.

```
#include <stdlib.h> #include <stdio.h>
#include "common_types.h" #include "uRedeMisturadores.h" #include
"uParChaves.h"
ParChaves *chaves; RedeMisturadores *rede;
void votar() {
  TextoPlano voto;
  printf("Voto: ");
  scanf("%d", &voto);
  rede->recebaVoto(voto);
  // rede->escrevaUrnas(stdout, chaves->informe_chavePrivada());
}
void resultados() {
  fprintf(stdout, "---- Cedulas ----\n");
  rede->escrevaCedulas(stdout);
  fprintf(stdout, "\n\n---- Votos ----\n");
  rede->escrevaVotos(stdout, chaves->informe_chavePrivada());
  fprintf(stdout, "\n\n---- Resultado ----\n");
  rede->escrevaResultado(stdout, chaves->informe_chavePrivada());
}
void reinicializar() {
  UINT32 p;
  UINT32 g;
  ChavePrivada x;
  UINT32 votoMin;
  UINT32 votoMax;
  UINT32 numMisturadores;
  UINT32 numElemUrnaMisturada;
  printf("\nCriando par de chaves\n");
  printf("  p (primo):");
  scanf("%d", &p);
  printf("  g (g pertence a Zp*):");
  scanf("%d", &g);
  printf("  x (chave privada):");
  scanf("%d", &x);
  chaves = new ParChaves(p, g, x);
  printf("\nInicializando rede\n");
  printf("  Voto minimo: ");
  scanf("%d", &votoMin);
  printf("  Voto maximo: ");
  scanf("%d", &votoMax);
  printf("  No. Misturadores: ");
  scanf("%d", &numMisturadores);
  printf("  No. elem urna misturada");
  scanf("%d", &numElemUrnaMisturada);
  rede = new RedeMisturadores(chaves->informe_chavePublica(), (UINT8)votoMin,
         (UINT8)votoMax, (UINT8)numMisturadores, (UINT32)numElemUrnaMisturada);
  rede->escrevaUrnas(stdout, chaves->informe_chavePrivada());
}
```

```
main(int argc, char *argv[]) {
    int op;
  chaves = new ParChaves(2579, 2, 765);
  rede = new RedeMisturadores(chaves->informe_chavePublica(), 1, 4, 3, 8);
    do {
        printf("\n\n");
        printf("1. Reinicializar rede\n");
    printf("2. Votar\n");
    printf("3. Resultados\n");
        printf("0. Sair");
        printf("\nOpcao: ");
        scanf("%d", &op);
        switch (op) {
        case 1:
      reinicializar();
            break;
        case 2:
      votar();
            break;
        case 3:
      resultados();
            break;
        case 4:
            break;
        case 5:
            break;
        }
    } while (op != 0);
  delete rede;
  delete chaves;
    return(0);
}
```