UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

O PROTOCOLO SET PARA SEGURANÇA DE COMPRAS ELETRÔNICAS

Por

PAULO CESAR DE SOUZA CAVALCANTE

RECIFE/PE
2003

PAULO CESAR DE SOUZA CAVALCANTE

O PROTOCOLO SET PARA SEGURANÇA DE COMPRAS ELETRÔNICAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Pernambuco em cumprimento às exigências para obtenção do título de

Mestre em Engenharia Elétrica

Ricardo Menezes Campello de Souza
Orientador

Universidade Federal de Pernambuco

Pós-Graduação em Engenharia Elétrica

PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE DISSERTAÇÃO DE MESTRADO DE

PAULO CESAR DE SOUZA CAVALCANTE

TÍTULO

"O PROTOCOLO SET PARA SEGURANÇA DE COMPRAS ELETRÔNICAS"

A comissão examinadora composta pelos professores: RICARDO MENEZES CAMPELLO DE SOUZA, DES/UFPE, VALDEMAR CARDOSO DA ROCHA JÚNIOR, DES/UFPE e MANOEL JOSÉ MACHADO SOARES LEMOS, DM/UFPE, sob a presidência do primeiro, consideram o candidato PAULO CESAR DE SOUZA CAVALCANTE **APROVADO**.

Recife, 23 de junho de 2003.

RICARDO MENEZES CAMPELLO DE SOUZA

VALDEMAR CARDOSO DA ROCHA JÚNIOR

MANOEL JOSÉ MACHADO SOARES LEMOS

O PROTOCOLO SET PARA SEGURANÇA DE COMPRAS ELETRÔNICAS

DEDICATÓRIA

O autor deseja expressar seu reconhecimento.

Ao amor e dedicação da minha querida mãe que desde os primórdios de minha educação até grande parte deste trabalho esteve ao meu lado, e que Deus não permitiu estivesse presente para compartilhar a satisfação de mais uma etapa vencida.

Ao amor, compreensão, apoio e grandioso desprendimento de minha esposa e meus filhos pela aceitação resignada das longas ausências decorrentes da dedicação à obtenção de êxito em mais este desafio de minha vida.

A paciência e competência de meus mestres na impagável tarefa de transmissão de seus valiosos ensinamentos e experiências.

Ao companheirismo e incentivo de meus amigos nos momentos necessários durante a longa jornada que ora se conclui.

RESUMO

Em 1997 foi apresentado pelas empresas de crédito VISA e MasterCard, em conjunto com outras empresas de crédito e de computação, o protocolo SET (*Secure Electronic Transactions*), um conjunto de especificações de um padrão para permitir transações do comércio eletrônico com cartões de pagamento através da Internet, permitindo confidencialidade, autenticação das partes envolvidas e garantia de integridade de dados. Esse trabalho apresenta um estudo crítico do SET, do ponto de vista de sua segurança e funcionalidade.

O SET é um protocolo aberto que utiliza mensagens não proprietárias, agrupadas funcionalmente em protocolos para gerenciamento de certificados e para pagamentos, onde a segurança dos dados durante o tráfego na rede de computadores se dá pela utilização de criptografia de chave secreta e criptografia de chave pública. Os algoritmos criptográficos aplicados às mensagens e às estruturas de dados que as compõem são *hash* SHA-1, *hash* com chave HMAC, cifragem de dados com chave secreta DES CBC e CDMF CBC, envelopamento digital com enchimento OAEP, assinatura e cifragem de chave pública RSA. Grandes e reconhecidos fornecedores disponibilizam no mercado uma série de produtos de *hardware* e *software* para atendimento das necessidades de processamento da aplicação e da criptografia do SET para os Portadores de cartão, Comerciantes e Instituições Financeiras envolvidas.

O protocolo SET possui algumas importantes vantagens sobre outros de sua categoria, como por exemplo, o SSL, PGP e S/MIME e com a intenção de torná-lo um padrão de protocolo de segurança, o Consórcio SET, que reúne diversas empresas interessadas, lideradas pelas Visa e MasterCard, têm planejado sua evolução, através da adoção de algoritmos criptográficos como ECC, e muito provavelmente AES, e dedicado especial atenção à sua utilização com *SmartCards*.

O SET, principalmente devido ao uso inteligente de modernas técnicas criptográficas, tende a se tornar um padrão para transações financeiras *on-line*. Entretanto ainda há algumas dificuldades, intrínsecas ou extrínsecas ao protocolo, a serem eliminadas e só o futuro indicará sua capacidade de lidar com grandes populações de portadores de cartões de crédito ao redor do mundo.

ABSTRACT

In 1997, the credit card companies VISA and MasterCard, together with other financial and information technology companies, introduced the SET protocol (Secure Electronic Transactions), a group of specifications of a standard to allow e-commerce transactions with payment cards through the Internet, providing confidentiality, authentication of the involved parts and data integrity. This dissertation presents an evaluation study of the SET protocol, focusing on its viability to become a security standard for e-commerce.

SET is an open protocol that uses non-proprietary messages, functionally grouped in protocols for the administration of certificates and payments. In this scenario, data security is provided by the use of both secret and public-key cryptography. The cryptographic algorithms that are applied to the messages and the structures of data that compose them, are the hash secure algorithm SHA-1, hash with key HMAC, secret-key encryption with the DES operating in the CBC and CDMF CBC mode, public-key encription and digital signature with the RSA algorithm, and Optimal Asymetric Encryption Padding. Large and well known suppliers have provided the market with a series of hardware and software products to fulfill the processing needs of the application and of the cryptography of SET, for the Cardholders, Merchants and Financial Institutions involved.

The SET protocol possesses some important advantages over other similar protocols, such as, for instance, SSL, PGP and S/MIME. Aiming at making it a standard security protocol, the SET Consortium, led by VISA and MasterCard and involving several other companies, has been planning its evolution, through the adoption of new cryptographic techniques and algorithms such as Elliptical Curve Cryptography, and the recently established Advanced Encryption Standard (AES), and also dedicating special attention to its use with SmartCards.

Although SET, mainly due to its cryptographic robustness, tends to become a standard for on-line financial transactions, there are still some difficulties to overcome and its capacity for adequately handle large populations of cardholders throughout the world remains to be seen.

SUMÁRIO

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Desde que as sociedades humanas estruturaram-se, tem havido a necessidade de se ocultar informações entendidas, cada uma a seu tempo, como segredos. Sejam segredos familiares, segredos sentimentais, segredos pessoais, segredos religiosos, ou segredos militares ou governamentais. Tão forte quanto a necessidade de guardar estes segredos é o desejo de outros de desvendar esses mesmos segredos. Seja por dinheiro, poder, vingança, curiosidade, arrogância, ou qualquer outro sentimento, essa tem sido uma batalha que ao longo dos anos vem sendo travada entre aqueles que querem guardar segredos e os que querem desvendar esses segredos.

Nesse contexto, a disponibilidade, a qualidade e o controle sobre a informação ganham outro grau de magnitude na importância estratégica que esta sempre teve para os governos e para as empresas. Assim, quanto maior o fluxo de informações em redes de telecomunicações, ou maior a quantidade de informação armazenada em meios computacionais, maior é a necessidade de empresas, governos (e até de pessoas físicas) de se protegerem contra uma velha ameaça que agora ganha outras feições com desenvolvimento da informática: o furto e a adulteração de informação.

Na atualidade, com o avanço cada vez maior dos poderes das Redes de Computadores, as distâncias entre os vários agentes distribuídos ao longo do planeta tendem a ficar menores. Na mesma escala de crescimento dos poderes das Redes de Computadores crescem também as ameaças às informações que ali trafegam ou estão contidas. Estas ameaças são constituídas pela possibilidade de exploração de fragilidades dos sistemas em produção nas Redes de Computadores, de forma intencional ou não.

Os sistemas estão sujeitos a ataques de duas naturezas básicas:

- Ataque ativo - informações são modificadas. São eles: interrupção, modificação e embuste;
- Ataque passivo - informações não sofrem modificação, sendo somente copiadas. Caracteriza-se pela interceptação.

As ameaças típicas, contra as quais as organizações despendem maior esforço para coibir ou neutralizar, e que de forma direta ou indireta afetam as pessoas físicas são:

- Violação de autorização: Uso de autorização para outra finalidade;
- Recusa de Serviço: Não atendimento, sem motivo explícito, das requisições dos legítimos usuários;
- Espionagem: Obter a informação sem autorização do proprietário;
- Vazamento: Revelação indevida de informação;
- Violação de integridade: Edição não autorizada de informação;
- Mascaramento: Passar-se por outro, embuste;
- *Replay:* Retransmissão ilegítima;
- Repudiação: Negação imprópria de uma ação ou transação efetivamente realizada;
- Exaustão: Sobrecarga de utilização de recurso;
- Emulação: Imitação para conseguir informações sensíveis;
- Roubo: Posse ilegítima de informações;
- Porta dos Fundos: Programação inserida e escondida no sistema, que possibilita a entrada de forma não convencional;
- Cavalo de Tróia: Programa de captura indevida de informações [9].

O Instituto da Segurança do Computador, entidade americana que conduz o exame de crimes de segurança de computador em conjunto com o esquadrão de intrusão de computador do FBI, em São Francisco, anunciou os resultados de sua sexta pesquisa anual do crime e de segurança de computador. Com base nas respostas de 538 pesquisados em empresas dos Estados Unidos, agências de governo, instituições financeiras, instituições e universidades médicas, dão conta que a ameaça de crimes de computador e de outras rupturas da segurança da informação continua sendo uma intranquilidade. Os destaques da pesquisa do crime na segurança de computadores em 2001 incluem:

- 85 por cento dos pesquisados (primeiramente empresas e agências de governo) detectaram falhas na segurança do computador dentro dos últimos doze meses;
- 64 por cento reconheceram as perdas financeiras devido às rupturas da segurança dos computadores;
- 35 por cento (186 pesquisados) podiam quantificar suas perdas financeiras.

Estes últimos 186 pesquisados relataram U$ 377,828,700.00 em perdas financeiras, em contraste, com as perdas de 249 pesquisados em 2000 que totalizaram U$ 265,589,940.00 enquanto que o total anual médio sobre os três anos anteriores a 2000 foi de U$ 120,240,180.00. Como em anos anteriores, dentre as perdas financeiras, as mais sérias ocorreram como roubo da informação proprietária, onde 34 pesquisados relataram prejuízos na ordem de U$ 151,230,100.00 e 21 relataram U$ 92,935,500.00 em fraudes financeiras.

Pelo quarto ano consecutivo, mais de 70% citaram sua conexão por meio da Internet como o ponto freqüente do ataque e não os seus sistemas internos. Daquelas que citam suas conexões Internet como o ponto freqüente do ataque observou-se um crescimento de 59% em 2000 a 70% em 2001[28].

Comércio Eletrônico

A Grande Rede Mundial de Computadores, a Internet, está alterando a maneira pela qual nos comunicamos e pagamos por serviços, acessamos as informações, pagamos e adquirimos mercadorias. Vários serviços financeiros como pagamentos de conta, corretagem, seguros, e *home banking* estão ou estarão disponíveis em larga escala na Internet.

Não há dúvidas que o comércio eletrônico, a exemplo da popularidade da Internet, está causando um grande impacto nos serviços fornecidos pelas instituições financeiras. Nenhuma instituição financeira deixará de ser afetada direta ou indiretamente pela explosão do comércio eletrônico.

O número de compras com cartão de crédito realizado através deste meio deve crescer com os pedidos *on-line* dos sistemas baseados na Internet.

Vários bancos estão planejando aderir a esta nova forma de comércio eletrônico oferecendo autorizações para pagamentos com cartões de crédito diretamente pela Internet.

Considerando-se o aspecto de segurança da informação, para que tais transações sejam efetuadas sem prejuízo para o lado do consumidor e até mesmo do servidor dos fornecedores dos serviços, a criptografia na Internet se tornou mais que uma realidade, é sim uma necessidade, sendo que a cada dia tenta-se implementar algoritmos cada vez mais poderosos e difíceis de serem decifrados por pessoas não autorizadas.

Os protocolos de suporte à segurança dos sistemas de pagamento e de suas instituições financeiras têm uma função significativa, estabelecendo especificações abertas para transações de pagamentos com cartão que:

- Proporcionam transmissões confidenciais;
- Autenticam as partes envolvidas;
- Garantem a integridade das instruções de pagamento para bens e serviços.

Confidencialidade da Informação

Para facilitar e encorajar o comércio eletrônico usando produtos como cartões de pagamento, será necessário garantir aos portadores de cartão que as suas informações de pagamento estão seguras e somente podem ser acessadas pelo destinatário. Portanto, a conta dos portadores de cartão e as informações de pagamento devem ser asseguradas em suas viagens pela rede, com as especificações prevenindo a intercepção dos números das contas e suas datas de expiração por indivíduos não autorizados.

No ambiente de compras *on-line* atual, instruções contendo informações de pagamento são freqüentemente transmitidas pelos portadores de cartões aos comerciantes sobre redes abertas com poucas preucauções de segurança. Contudo, esta informação da conta proporciona os elementos chave necessários para criar cartões falsificados e/ou transações fraudulentas. Enquanto é possível obter informações de contas em outros ambientes, há um aumento da facilidade de se fazer isso com transações em redes públicas. Esta preocupação reflete o potencial para um alto volume de fraudes, fraudes automatizadas (como a utilização de filtros sobre todas as mensagens que passam sobre a rede para extrair todos os números de contas com pagamento em cartão fluindo na rede), e também o potencial para "fraudes exibicionistas", que parecem ser características de alguns *hackers*.

Integridade dos Dados

A informação do pagamento enviada pelos portadores de cartão aos comerciantes inclui a informação do pedido, os dados pessoais, e as instruções de pagamento. Se qualquer componente for alterado na transição, a transação não será processada corretamente. Para eliminar esta fonte potencial de fraude e/ou erro, as especificações do sistema de proteção de informação devem proporcionar os meios para garantir que o conteúdo de cada pedido e a mensagem de pagamento recebida correspondam ao conteúdo da mensagem enviada.

Autenticação da Conta do Portador do Cartão

Os comerciantes precisam de uma maneira para verificar que um portador de um cartão é o legítimo usuário da conta do cartão. Um mecanismo que usa tecnologia para ligar um portador de cartão a um número de uma conta de pagamento de um cartão específico reduzirá a incidência de fraude e por isso o custo global do processamento do pagamento.
As especificações devem definir o mecanismo para verificar que o portador do cartão é um usuário legítimo de um número válido da conta de pagamento do cartão.

Autenticação do Comerciante

As especificações devem proporcionar uma maneira para que os portadores de cartão confirmem que o comerciante possui um relacionamento com uma instituição financeira que o permite aceitar pagamentos em cartão. Os portadores de cartão também precisam estar aptos a identificar os comerciantes com os quais ele pode conduzir seguramente o comércio eletrônico.

Interoperabilidade

As especificações devem ser aplicáveis em uma variedade de plataformas de *hardware* e *software*, e não devem incluir uma preferência de uma sobre a outra. Qualquer portador de um

cartão de crédito com *software* compatível deve estar habilitado a se comunicar com o *software* do comerciante que também faz parte do padrão definido [9].

Protocolo SET (*Secure Electronic Transations)*

As especificações do protocolo para guiar informações eletrônicas de forma segura entre os diversos participantes de comércio eletrônico na *World Wide Web* que serão abordadas ao longo deste trabalho são a do SET. O protocolo é apresentado através de suas principais características, de modo a possibilitar o entendimento do seu funcionamento básico, mas principalmente para demonstrar como ele realiza sua principal tarefa , ou seja, como protege as informações sensíveis e importantes durante o tráfego das transações comerciais entre seus vários participantes, em diversos pontos do planeta, onde houver a possibilidade de acesso à Internet. Esta árdua e cada vez mais difícil tarefa de proteção de informações eletrônicas em tráfego na Internet é realizada através de tratamentos criptográficos em suas mensagens. Desta forma, a principal abordagem aqui, é como ele realiza este processamento criptográfico. Como não poderia deixar de ser, muitas outras informações sobre suas características e funcionalidades, principalmente no que se refere a como se compõem suas mensagens, e como elas fluem através de uma rede de comunicação de dados interligando os usuários finais, são apresentadas para formar um cenário de exposição de como as ferramentas criptográficas vêm atender às necessidades de segurança da informação anteriormente expostas.

Nesta introdução se expôs brevemente a necessidade e os aspectos da segurança da informação que trafega em redes que interligam tipos de plataformas de incontáveis configurações distintas. Um protocolo que se destine a promover a segurança desta informação tem que ter como premissa básica ser padrão não proprietário. O SET é a proposta deste padrão.

Na sequência, o capítulo 2 é dedicado à discussão das noções básicas da criptografia de chave secreta e de chave pública. São abordados os conceitos da criptografia utilizada pelos cripto-sistemas utilizados para assinatura e cifragem de mensagens.

O capítulo 3 apresenta o SET, como ele se insere na compra eletrônica e como seus participantes se conectam num ambiente SET. As principais características da segurança da informação providas pelo protocolo são expostas, bem como se discorre sobre os aspectos relevantes do gerenciamento de certificados digitais, um pilar fundamental da segurança do protocolo.

O capítulo 4 aborda a formatação e composição das mensagens do SET. Padrões de sintaxe de informação e de codificação utilizados são referenciados. A funcionalidade das mensagens do SET e seus fluxos nos protocolos de gerenciamento de certificados e de sistemas de pagamentos são descritos.

Com o foco na criptografia, o capitulo 5 aborda os padrões criptográficos em que se baseia o SET. É examinada a sintaxe para aplicação de operadores de assinatura, cifragem e encapsulamento das mensagens do SET.

Em continuidade ao capítulo 5, o capítulo 6 apresenta o processamento integral das mensagens do SET, desde a composição de envoltória de mensagem até o seu encapsulamento nos padrões aplicáveis, passando pela utilização dos operadores criptográficos conforme as especificações do protocolo. Neste capítulo encontram-se todas as principais mensagens do protocolo, com a criptografia especificada para as mesmas.

O capítulo 7 dá relevância ao esforço requerido para manutenção da segurança pelo tratamento criptográfico. Faz uma incursão nos aspectos técnicos de *software* e *hardware* criptográficos. Apresenta também empresas desenvolvedoras e produtos comercialmente disponíveis para o SET.

Finalmente, no capítulo 8 concluímos o trabalho apresentando diversas considerações sobre o SET e contexto do comércio eletrônico no mundo e no Brasil. As vantagens e dificuldades do protocolo são expostas juntamente com as perspectivas do protocolo para o futuro.

Como informações adicionais, o apêndice A exemplifica a aplicação do algoritmo de cifragem *DES*. Os Apêndices B, C, D são respectivamente a relação de acrônimos encontrados, o glossário e a listagem de padrões externos utilizados pelo SET.

Capítulo 2

Noções de Criptografia

O SET realiza o processamento de pagamentos e fluxos de certificados digitais de comércio eletrônico, utilizando-se de mensagens não proprietárias especificadas pelo protocolo. Estas mensagens, que trafegam nas redes públicas, são protegidas pelo protocolo através da utilização de algoritmos de criptografia. Os algoritmos criptográficos aplicados às mensagens ou estruturas de dados componentes das mensagens são cifragem de chave secreta, ou cifragem simétrica, usando os algoritmos *DES* ou CDMF com o modo de operação CBC, resumo e assinatura digital usando a função *hash* SHA-1 e a cifragem de chave pública, ou assimétrica, usando algoritmo RSA com enchimento prévio OAEP.

Assim, com o foco voltado ao protocolo SET, este capítulo não objetiva discutir em profundidade os conceitos da ciência criptográfica, mas sim apresentar noções das técnicas criptográficas de chave secreta e de chaves públicas usadas pelo protocolo.

2.1. Criptografia de Chave Secreta

2.1.1. Data Encryption Standard - DES

Este algoritmo foi o mais amplamente usado internacionalmente, e apresentou um avanço científico significativo no sentido de ter sido o primeiro algoritmo de criptografia cujo conhecimento se tornou público; até então todos os algoritmos eram secretos. Ou seja, a segurança do *DES* não se baseia no desconhecimento do algoritmo, mas apenas no desconhecimento da chave secreta. Foi projetado pela IBM e publicado pelo *National Bureau of Standards* – NBS em 1977 para ser adotado como padrão nos Estados Unidos para informações comerciais.

O seu projeto é inspirado em outro algoritmo da IBM chamado LUCIFER[29], que possui entrada e saída em 64 bits, mas chave mais longa, de 128 bits. Acredita-se que o encurtamento de chave tenha sido proposital, a pedido da *National Security Agency* – NSA – nos Estados Unidos. O líder da IBM para o projeto LUCIFER (final da decada de 1960) foi Horst Feistel, e os líderes do projeto *DES* foram Walter Tuchman e Carl Meyer (meados da década de 1970) [6].

Além dos pesquisadores da IBM, o projeto *DES* teve a participação de consultores da NSA para aprimorar o LUCIFER. Antes da adoção como padrão, o *DES* foi alvo de fortes críticas que persistem até hoje. A primeira crítica é o decréscimo do comprimento da chave de 64 para 56 bits; isso torna muito mais economicamente viável o cálculo da chave secreta mesmo por força bruta, isto é, enumerando-se as 2^{56} possíveis chaves. A segunda crítica é o segredo mantido quanto aos critérios de projeto da estrutura interna, os *S-boxes*. Os usuários do *DES* não possuem qualquer garantia de ausência de pontos vulneráveis nos *S-boxes* que permitam ao NSA decifrar sem conhecer a chave em uso. De acordo com os pesquisadores da IBM, as alterações feitas nos *S-boxes* do LUCIFER que resultaram nos *S-boxes* do *DES* foram sugeridos pela NSA para remover vulnerabilidades detectadas durante a avaliação do projeto.

Apesar dessas críticas, o DES é muito utilizado, especialmente na área de finanças . Em 1994 o NBS que agora se chama NIST – *National Institute of Standards and Technology* – prolongou por mais cinco anos a sua adoção como padrão para aplicações no governo dos EUA, após ter publicado a revisão do padrão *DES* através da *FIPS Publication 46-2, Data Encryption Standard* [12].

Em 1997 o NIST lançou uma competição aberta para o sucessor do *DES* , chamado AES - *Advanced Encryption Standard*. Dezoito propostas de vários países foram inscritas e três foram descartadas (isto é, mostraram fortes vulnerabilidades) por pesquisadores, em agosto de 1998 durante uma reunião de avaliação organizada pelo NIST, na Califórnia. Em março de 1999 foi realizada uma outra reunião em Roma, mas nenhuma proposta foi descartada [6].

Em meados de 1999 o NIST anunciou a lista dos cinco finalistas: Mars, RC6, Twofish, Serpent, e Rijndael. Hoje, o *DES* não é mais seguro. Em 2 de outubro de 2000, o Secretário de Comércio dos EUA anunciou o novo padrão, o *Advanced Encryption Standard* – AES [13] proposto à nação. Ele declarou vencedora a fórmula Rijndael (comumente pronunciada

"raine-dol") inventado por dois pesquisadores belgas: Vincent Rijmen e Joan Daemen [14]. Entretanto, existem algumas variantes mais seguras do *DES* e a maioria dos sucessores sugeridos para o *DES* são similares a ele. Portanto o *DES* ainda é um cripto-sistema importante.

Um substituto ao *DES* amplamente utilizado é o *Triple DES*. O *Triple DES* realiza três vezes o algoritmo *DES*. O bloco de dados é cifrado por meio do *DES* utilizando-se uma chave, então cifra-se esse resultado com uma outra chave do *DES*. Em seguida, o processo é repetido numa terceira vez com uma nova chave do *DES*.

Esse sistema apresenta dois problemas. Primeiro, os criptoanalistas descobriram uma maneira de simplificar o ataque por força bruta. Poderia-se pensar que isso requereria um ataque por força bruta de "168 bits", porém há maneiras inteligentes para reduzir isso ao equivalente de um ataque por força bruta de 108 bits [5]. Uma chave que seja equivalente a 108 bits ainda é segura , mas essa "fraqueza" incomoda. O segundo problema é a velocidade. O *Triple DES* é três vezes mais lento que o *DES*. Alguns aplicativos precisam de resposta de alta velocidade com vários megabytes de informações. O *Triple DES* reduz de tal forma o desempenho que alguns aplicativos podem não funcionar.

Commercial Data Masking Facility - CDMF

O *CDMF* [30] é uma técnica de espalhamento que utiliza o *DES* como um algoritmo criptográfico subjacente, porém torna a operação geral criptográfica mais fraca, pois define um método de transformação que produz uma chave de 40 bits, ao invés do comprimento de chave de 56 bits requerido para uma criptografia mais forte. A redução do comprimento de chave se destina ao atendimento de restrições de exportação de algoritmos de criptografia pelos EUA. Então, o comprimento de chave efetivo de 40 bits do *CDMF* é expandido para uma chave *DES* de 56 bits para uso no algoritmo *DES*. Desde que o algoritmo do CDMF não é tão resistente quanto o *DES* em relação à pesquisa exaustiva de chaves, o *CDMF* provê mais uma forma de mascaramento de dados do que cifragem. A chave *CDMF* transmitida no protocolo *SET* é a chave antes de sua transformação para uso pelo *DES*. Em outras palavras, a chave *CDMF* é tratada normalmente como uma chave do *DES*.

Cifragem DES

O DES é um tipo de cifra de Feistel [15], sendo uma cifra de blocos com alfabeto $\{0,1\}$. Fixa-se um número $r \geq 1$ de iterações , um espaço de chaves k e um método para gerar uma sequência de chaves de iteração, a partir de qualquer chave $k \in k$.

A função de codificação criptográfica E_k da cifra de Feistel para a chave $k \in k$ funciona como segue. Seja p um texto claro de comprimento $2t$, que é dividido em duas metades de comprimento t, ou seja , escreve-se $p = (L_0, R_0)$, onde L_0 é a metade esquerda e R_0 é a metade direita. Então, a sequência

$$(L_i, R_i) = (R_{i-1}, L_{i-1} \ldots \oplus f_{ki}(R_{i-1})), \quad 1 \leq i \leq r \tag{2.1}$$

é construída, onde $E_k(L_0, R_0) = (L_r, R_r)$.

Claramente, a segurança da cifra de Feistel depende da segurança da cifra em bloco interna. Desde sua invenção, a segurança do *DES* tem sido estudada com muita intensidade. Foram inventadas técnicas especiais como criptoanálise diferencial e linear para atacar o *DES*, mas o ataque mais bem sucedido tem sido uma pesquisa exaustiva do espaço de chave. Com *hardware* especial ou grandes redes de micros, agora é possível decifrar textos cifrados com o *DES* em poucos segundos.

As dezesseis iterações com as chaves K_i são mostradas na Figura 2.1, que representa a operação de cifragem do *DES*.

Os espaços de texto claro e texto cifrado do *DES* são $P = C = \{0,1\}^{64}$. As chaves no *DES* são todas as cadeias de bits de comprimento 64 com a seguinte propriedade. Se uma chave *DES* de 64 bits é dividida em 8 bytes, então a soma dos 8 bits de cada byte é ímpar. Isso significa que 7 dos 8 bits determinam o valor do oitavo bit. Podem ser detectados os erros de transmissão de um bit. Portanto o espaço da chave é:

$$k = \{(b_1, \ldots, b_{64}) \in \{0,1\} : \sum_{i=1}^{8} b_{8k+i} \equiv 1 \bmod 2, \qquad 0 \leq k \leq 7\}$$

O número de chaves *DES* é $2^{56} \approx 7.2 \times 10^{16}$.

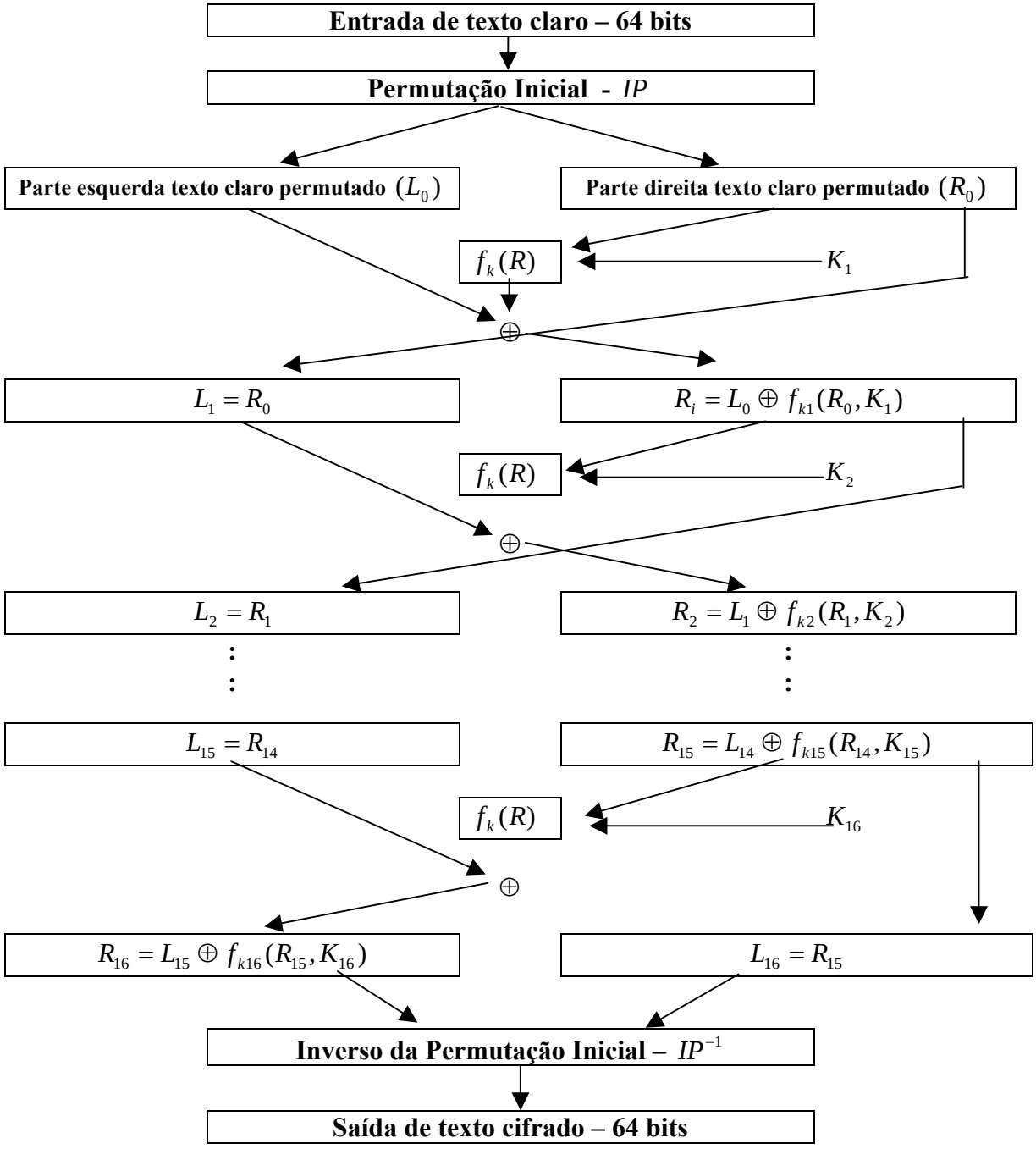

Figura 2.1: A cifragem *DES*

Permutação Inicial

Um exemplo de chave *DES* hexadecimal válida é: 133457799*BBCDFF*1.

Dado um texto inicial p, o *DES* trabalha em três etapas. A expansão binária da chave *DES* é exibida no quadro abaixo.

Expansão binária de Chave do *DES* : 133457799*BBCDFF*1							
0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

Antes da codificação criptográfica de Feistel, o *DES* aplica uma permutação inicial (*IP*) abaixo sobre a organização do texto inicial p. Isso é uma permutação de bits de grau 64 que é independente da chave escolhida.

$$\text{Se } p \in \{0,1\}^{64},\ p = p_1 p_2 p_3, \ldots, p_{64} \quad \textit{então} \quad IP(p) = p_{58} p_{50} p_{42} \ldots p_r$$

Os quadros abaixo mostram a organização do texto inicial p e o resultado da permutação *IP* sobre o mesmo.

Organização texto inicial p							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Resultado de *IP* sobre p							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

É aplicada ao texto claro permutado a cifra de Feistel em 16 iterações. Após a aplicação das 16 iterações sobre p, finalmente o texto cifrado $c = IP^{-1}(R_{16}, L_{16})$ é construído usando a permutação inversa IP^{-1}, conforme demonstrado abaixo. Após a aplicação de IP^{-1}, o texto cifrado c de saída se apresenta com essa organização exibida a seguir.

Texto final permutado – (R_{16}, L_{16})							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

$c = IP^{-1}(R_{16}, L_{16})$							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Cifra de bloco interna

A seguir é descrita a função criptográfica f implementada em cada iteração no DES. Seu alfabeto é $\{0,1\}$, o comprimento de bloco é 32 e seu espaço de chave é $\{0,1\}^{48}$. A função criptográfica é $f_k : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$ para uma chave $k \in \{0,1\}^{48}$. A Figura 2.2. representa esta função.

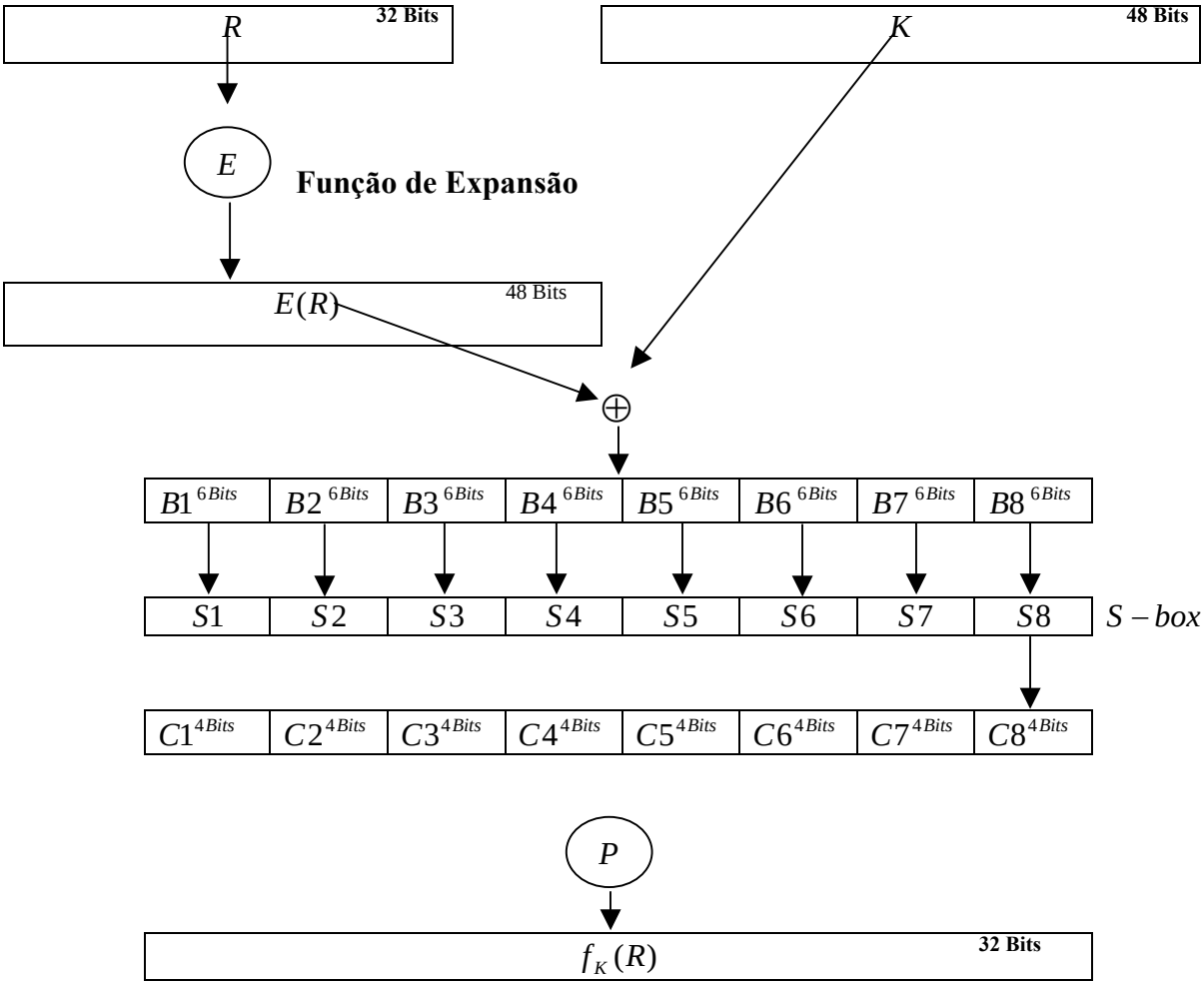

Figura 2.2: A função f do *DES*

O argumento $R \in \{0,1\}^{32}$ é expandido pela função expansão $E:\{0,1\}^{32} \to \{0,1\}^{48}$. A organização dos bits do argumento R e o resultado da aplicação da função E são mostrados nos quadros seguintes. Se $R = R_1 R_2 \ldots R_{32}$, então $E(R) = R_{32} R_1 R_2 \ldots R_{32} R_1$.

R			
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32

$E(R)$					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

A seguir é computado $E(R) \oplus K$, e o resultado é dividido em oito blocos B_i, $1 \le i \le 8$ de comprimento 6, a saber:

$$E(R) \oplus K = B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8 \qquad (2.2.)$$

computados com $B_i \in \{0,1\}^6$, $1 \le i \le 8$

S-boxes

Nesta etapa, são usadas as funções $S_i : \{0,1\}^6 \to \{0,1\}^4$, $1 \le i \le 8$, denominadas *S-boxes* ,cuja função é descrita abaixo.

Os *S-boxes* (*Substitution-boxes*), S_i , $1 \le i \le 8$ **,** representam a parte não linear do DES. Cada *S-box* é representado por uma tabela com 4 linhas e 16 colunas. Para cada cadeia $B = b_1 b_2 b_3 b_4 b_5 b_6$, o valor $S_i(B)$ é computado como se segue. O número inteiro com expansão binária $b_1 \; b_6$ é usado como índice da linha. O número inteiro com expansão binária $b_2 b_3 \; b_4 b_5$ é usado como índice da coluna. O elemento do *S-box* na linha e na coluna é escrito em expansão binária. Zeros são anexados a essa expansão, de modo que seu comprimento seja quatro. O resultado é $S_i(B)$ .A Tabela 2.1. expõe o resultado das operações *S-boxes* do DES.

Por exemplo, a computação de $S_1(001011)$. O primeiro bit é 0 e o último é 1. Portanto, o índice da linha é o número inteiro cuja expansão binária é 01, isto é, 1. Os quatro bits intermediários são 0101. Esta é a expansão binária de 5. Desta forma, o índice da coluna é 5. O elemento na linha 1 coluna 5 do primeiro S-box é 2. A expansão binária de 2 é 10. Assim, $S_1(001011) = 0010$.

Usando-se essas funções, a cadeia $C = C_1 C_2 C_3 \; C_4 C_5 C_6 C_7 C_8$ é determinada, onde

$Ci = S_i(B_i)$, $1 \le i \le 8$**.** Ela tem comprimento 32. É então aplicada a Permutação Final P à cadeia C resultando em é $f_k(R)$.

Bits da cadeia C			
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32

$f_K(R)$ **Resultado de** $P(C)$			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Chaves K do DES

Para finalizar o algoritmo criptográfico DES, são computadas as iterações com a chave k do DES. Fazendo-se $K \in \{0,1\}^{64}$ ser uma chave DES. Gera-se as chaves de iteração $K_i, 1 \le i \le 16$, de comprimento 48. Define-se os valores $v_i, 1 \le i \le 16$, como se segue:

$$v_i = 1 \ \ para \ \ i \in \{1, 2, 9, 16\}; \ v_i = 2, \ \text{caso contrário}$$

As chaves de iteração são computadas pelo seguinte algoritmo, usando-se as funções

$PC1: \{0,1\}^{64} \to \{0,1\}^{28} \ x\{0,1\}^{28}$, $PC2: \{0,1\}^{28} \ x\{0,1\}^{28} \to \{0,1\}^{48}$, que são assim descritas:

1- Estabelece-se $(C_0, D_0) = PC1(K)$.

2- Para $1 \le i \le 16$,

3- Faça-se C_i a cadeia que é obtida de C_{i-1} por deslocamento circular à esquerda de v_i posições;

(a) Faça-se D_i a cadeia que é obtida de D_{i-1} por deslocamento circular à esquerda de v_i posições;

(b) Determina-se $K_i = PC2(C_i, D_i)$

A função $PC1$ mapeia uma cadeia de k bits de comprimento 64 a duas cadeias de bits C e D de comprimento 28. Isso é mostrado nos quadros abaixo. A metade superior do quadro $PC1$ descreve C. Correto: Se $k = k_1 k_2 k_3 \ldots k_{64}$, então $C = k_{57} k_{49} \ldots k_{36}$. A metade inferior representa D, assim $D = k_{63} k_{55} \ldots k_4$.

Bits da Chave K							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

$PC1$ **Mapeamento da Chave** K **em** C_i **e** D_i							
57	49	41	33	25	17	9	C_i Cadeia de 28 Bits de Comprimento
1	58	50	42	34	26	18	
10	2	59	51	43	35	27	
19	11	3	60	52	44	36	
63	55	47	39	31	23	15	D_i Cadeia de 28 Bits de Comprimento
7	62	54	46	38	30	22	
14	6	61	53	45	37	29	
21	13	5	28	20	12	4	

A função $PC2$, mostrada abaixo, mapeia um par (C, D) de cadeias de bits de comprimento 28 (isto é, uma cadeia de 56 bits de comprimento) a uma cadeia de 48 bits de comprimento. O valor $PC2\ (b_1 \ldots b_{56})$ é $b_{14} b_{17} \ldots b_{32}$

Posicionamento bits do Par (C, D)						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56

$PC2$ **Mapeamento de Par** (C_i, D_i)						
14	17	11	24	1	5	Mapeamento de um par (C, D) de 58 bits em cadeia de comprimento de 48 bits
3	28	15	6	21	10	
23	19	12	4	26	8	
16	7	27	20	13	2	
41	52	31	37	47	55	
30	40	51	45	33	48	
44	49	39	56	34	53	
46	42	50	36	29	32	

Linha $(b_1 b_6)$	**Coluna -** $(b_2 b_3 b_4 b_5)$															
	(0)	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**	**(9)**	**(10)**	**(11)**	**(12)**	**(13)**	**(14)**	**(15)**
$S-box\ S_1$																

(0)	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
(1)	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
(2)	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
(3)	15	12	8	2	4	9	1	7	5	11	13	14	10	0	6	13
$S-box\ S_2$																

(0)	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
(1)	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
(2)	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
(3)	13	8	10	1	3	15	4	2	11	6	7	12	0	5	4	9
$S-box\ S_3$																

(0)	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
(1)	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
(2)	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
(3)	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
$S-box\ S_4$																

(0)	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
(1)	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
(2)	10	5	9	0	12	11	7	13	15	1	3	14	5	2	8	4
(3)	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
$S-box\ S_5$																

(0)	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
(1)	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
(2)	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
(3)	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
$S-box\ S_6$																

(0)	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
(1)	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
(2)	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
(3)	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
$S-box\ S_7$																

(0)	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
(1)	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
(2)	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
(3)	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
$S-box\ S_8$																

(0)	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
(1)	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
(2)	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
(3)	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Tabela 2.1: *S-boxes* do *DES*

Decifragem DES

Da expressão (2.1), obtém-se:

$$(R_{i-1}, L_{i-1}) = (L_i, R_i \oplus (f_{ki}(L_i)) \quad 1 \le i \le r \tag{2.3}$$

Usando essa equação em r iterações com a sequência inversa da chave $(K_r, K_{r-1}, \ldots K_1)$, o par de texto claro (R_0, L_0) é reconstruído a partir do texto cifrado (R_r, L_r). Portanto, para a cifra de Feistel, a cifragem e a decifragem são iguais, exceto que a sequência das chaves é invertida.

Um exemplo da aplicação do algoritmo de cifragem do DES aplicado sobre um texto claro está disponível no Apêndice A.

Operação do DES no Modo de Encadeamento de Blocos de Cifra DES-CBC

Vários métodos para implementação do DES são possíveis. Estes métodos, externos ao algoritmo do DES, são chamados "modos de operação". Quatro modos, denominados Modo *Electronic Codebook* (ECB), Modo *Cipher Block Chaining* (CBC), Modo *Cipher Feedback* (CFB), e Modo *Output Feedback* (OFB), são especificados para este padrão [16].

Conforme anteriormente visto, o *SET* especifica o modo de operação *CBC*. O modo *CBC*, encadeamento de blocos de cifra, é um sistema de cifragem de bloco no qual o primeiro bloco de dados de texto claro é submetido a uma operação ou-exclusivo com um bloco de dados pseudo-aleatório antes de ser processado pelo DES. O bloco de texto cifrado resultante é então submetido a uma operação ou-exclusivo com o próximo bloco de dados de texto claro para formar o próximo bloco de entrada para o DES, assim encadeando a cifragem dos blocos de dados.

O encadeamento de blocos de texto cifrados provê uma característica de extensão de erro que é valiosa protegendo contra alteração de dados fraudulenta. O modo *CBC* produz o mesmo texto cifrado sempre que o mesmo texto claro é cifrado usando a mesma chave e Vetor de Inicialização (*Inicializator Vector – IV*).

Cifragem e Decifragem no Modo CBC

Uma mensagem a ser cifrada é dividida em blocos de 64 bits. Na cifragem *CBC*, o primeiro bloco de entrada para o DES é formado pela operação ou-exclusivo do primeiro bloco de 64 bits de uma mensagem a ser cifrada, ou seja bloco de dados $D=(D1, D2, \ldots, D64)$ com um Vetor de Inicialização (*Inicializator Vector – IV*) de 64 bits, $IV=(IV1, IV2, \ldots, IV64)$, resultando em um bloco de 64 bits de entrada $I=(I1, I2, \ldots, 64)=(IV1 \oplus D1,\ IV2 \oplus D2, \ldots, IV64 \oplus D64)$ $I=(II, I2, \ldots, I64)=(IV1 \oplus D1, IV2 \oplus D2, \ldots, IV64 \oplus D64)$. O bloco de entrada é processado através do dispositivo do DES no estado de cifragem, e o bloco resultante na saída do dispositivo DES, $O=(O1, O2, \ldots, O64)$ é o texto cifrado $C=(C1, C2, \ldots, C64)=(O1, O2, \ldots, O64)$.
Este primeiro bloco de texto cifrado é então submetido a uma operação ou-exclusivo com o segundo bloco de texto claro para produzir o segundo bloco de entrada para o DES, isto é $(I1, I2, \ldots I64)=(C1 \oplus D1,\ C2 \oplus D2, \ldots,\ C64 \oplus D64)$. Note que agora I e D se referem ao segundo bloco. O segundo bloco de entrada é processado através do dispositivo DES no estado de cifragem para produzir o segundo bloco de texto cifrado. Este processo de cifragem continua a encadear sucessivos blocos de texto cifrado e claro juntos até que o último texto claro na mensagem seja cifrado.

Na decifragem CBC, o primeiro bloco de texto cifrado de uma mensagem cifrada é usado como o bloco de entrada e é processado por um dispositivo DES em estado de decifragem , isto é, $(I1, I2, \ldots, I64) = (C1, C2, \ldots, C64)$. O bloco de saída resultante, que é igual ao bloco de entrada original do DES durante a cifragem é submetido a uma operação ou-exclusivo com o IV (que deve ser o mesmo que aquele usado durante a cifragem) para produzir o primeiro bloco de texto claro, isto é, $(D1, D2, \ldots, D64) = (O1 \oplus IV1,\ O2 \oplus IV2,\ \ldots, O64 \oplus IV64)$.

O segundo bloco de texto cifrado é então usado como o bloco de entrada e é processado através do DES no estado de decifragem e o bloco de saída resultante é submetido a uma operação ou-exclusivo com o primeiro bloco de texto cifrado para produzir aquele segundo bloco de texto claro, isto é, $(D1, D2, \ldots, D64) = (O1 \oplus C1,\ O2 \oplus C2, \ldots, O64 \oplus C64)$. Note que novamente o D e O se referem àquele segundo bloco. O processo de decifragem CBC continua desta maneira até que o último bloco de texto cifrado tenha sido decifrado.

2.2. Criptografia de Chave Pública

Um problema importante dos cripto-sistemas de chave secreta vistos anteriormente é a distribuição e a gerência de chaves. Quando dois usuários, por exemplo, Ana e Beto, usam tal sistema, eles precisam intercambiar uma chave secreta antes que possam se comunicar secretamente. Para o intercâmbio de chaves eles precisam, por exemplo, de um canal seguro ou de um portador. O problema de intercâmbio torna-se ainda mais difícil se muitas pessoas desejam trocar mensagens cifradas, por exemplo, na Internet. Se uma rede de comunicação tem n usuários e dois quaisquer deles trocam uma chave, então são necessárias $n(n-1)/2$ trocas de chaves secretas e todas essas chaves precisam ser armazenadas em segurança. Para $n = 1.000$, existem 499.500 chaves. Outra possibilidade para organizar a troca de chaves é usar um centro de chaves, no qual todos os usuários trocam uma chave secreta com esse centro de chaves. Se Ana quer mandar uma mensagem para Beto, então ela cifra a mensagem usando sua chave secreta e a envia ao centro de chaves. O centro, conhecendo todas as chaves secretas, decifra a mensagem usando a chave de Ana, a cifra novamente usando a chave secreta de Beto e a remete a Beto. Dessa forma, o número de troca de chaves para n usuários fica reduzido a n. Entretanto, o centro de chaves passa a ter conhecimento de todas as mensagens secretas e deve armazenar todas as n chaves em segurança.

Nos sistemas de chave pública, a gerência de chaves é muito mais simples. Em um sistema de chave pública, usado para prover sigilo, apenas a chave de decifragem precisa ser mantida em segredo. Uma chave de decifragem é, portanto, chamada de chave privada. A chave de cifragem correspondente pode ser publicada. Ela é chamada de chave pública. Não é factível computar as chaves privadas a partir das suas correspondentes chaves públicas. Essa é a principal propriedade dos cripto-sistemas de chave pública. Se Beto deseja enviar uma mensagem para Ana, ele obtém a chave pública de Ana de um diretório de chaves. Então ele usa esta chave pública de Ana para cifrar a mensagem e remete essa mensagem cifrada para Ana. Ana, então, utilizando sua chave privada consegue decifrar a mensagem recebida.

Em cripto-sistemas de chave pública, não é necessário o intercâmbio de chaves entre os usuários. As chaves de cifragem são relacionadas em diretórios ou são obtidas através de certificados fornecidos por entidades que possuem essa finalidade. Embora todos possam ler os diretórios ou obter essas chaves públicas, elas precisam ser protegidas de escrita não autorizada. Se um usuário não autorizado for capaz de substituir a chave de cifragem, chave pública de Ana, por exemplo, pela sua própria, então ele poderá com sua chave de decifragem, chave privada, decifrar as mensagens que são destinadas a Ana.

Os cripto-sistemas de chave pública não apenas simplificam a gerência de chaves, mas podem também ser usados para gerar assinaturas digitais.

Resumindo, nos cripto-sistemas de chave pública, cada usuário possui um par de chaves K_s e K_p sendo K_s a chave privada, e K_p a chave pública. Estas chaves K_s e K_p são relacionadas matematicamente de tal forma que :

1. Se m denota um texto claro, c denota o texto cifrado, e $E_{kp}(.)$ denota o algoritmo de cifragem do texto claro m, ou seja $c = E_{kp}(m)$, e $D_{ks}(.)$ denota o algoritmo de decifragem do texto cifrado c, ou seja $m = D_{ks}(c)$, então K_s é a chave inversa de K_p, ou seja, $D_{ks}(E_{kp}(m)) = m$;

2. O cálculo do par de chaves $(K_s \, e \, K_p)$ é computacionalmente fácil;

3. É computacionalmente inviável calcular K_s a partir do conhecimento de K_p;

4. Os cálculos de $E_{ks}(.)$ e $D_{kp}(.)$ são computacionalmente fáceis para quem conhece as chaves;

Infelizmente, os sistemas conhecidos de chave pública, como por exemplo, o *RSA*, não são tão eficientes quanto muitos cripto-sistemas de chave secreta. Eles são lentos, ao passo que a criptografia de chave simétrica pode cifrar dados em grande quantidade bem mais rapidamente, cerca de centenas ou milhares de vezes mais rápido. Por isso, na prática são usadas combinações de sistemas de chave pública e sistemas de chave secreta, como no seguinte exemplo: Ana deseja enviar uma mensagem m cifrada para Beto. Ela gera uma chave secreta para utilização num cripto-sistema eficiente de chave secreta, aqui denominada então, como chave de sessão. Então ela cifra o texto claro m usando essa chave de sessão com o sistema de chave secreta, obtendo o texto cifrado c. Essa cifragem é rápida porque foi usado um cripto-sistema de chave secreta eficiente. Ana também cifra a chave de sessão com a chave pública do destinatário, ou seja, de Beto, a qual ela obtém de um diretório público ou certificado previamente recebido. Como a chave de sessão é normalmente menor que o texto claro, sua cifragem, embora pelo cripto-sistema de chave pública, normalmente mais lento que o cripto-sistema de chave privada, também é rápida. Então, Ana envia o texto cifrado **c** e a chave de sessão cifrada para Beto. Beto decifra a chave de sessão, e utilizando o cripto-sistema de chave secreta utilizado por Ana no modo de decifragem recupera o texto claro m. Aqui, o sistema de chave pública é apenas usado para a troca da chave de sessão. Isso combina a solução de gerência de chaves do cripto-sistema de chave pública com a eficiência do cripto-sistema de chave secreta.

O processo usado para cifrar grandes volumes de dados utilizando a criptografia de chave secreta e para cifrar a chave secreta com um algoritmo de chave pública é chamado de envelope digital. A idéia é que a criptografia de chave secreta cifra os dados em volume e a criptografia de chave pública, por sua vez, empacota a chave secreta em um envelope.

Em 1976 foi publicado um artigo seminal de W. Diffie M. E. Hellman, o qual inspirou a criação dos chamados cripto-sistemas de Chave Pública (ou Assimétricos) [17]. O protocolo de troca de chaves proposto por Diffie e Hellman para troca de chaves secretas em canais inseguros, que em si não é um cripto-sistema de chave pública, mas a base para o sistema de ElGamal [7], é um marco na criptografia de chave pública.

2.2.1. Cripto-sistema RSA

Este algoritmo foi publicado em 1978 e seu nome é derivado das iniciais dos seus autores: Ronald Rivest , Adi Shamir e Leonard Adleman [18] [19]. Sua segurança está intimamente relacionada à dificuldade de encontrar a fatoração de um número inteiro que é o produto de dois números primos grandes.

Geração das Chaves do RSA

Para entendimento do funcionamento do cripto-sistema RSA, é descrita a geração das chaves privada e pública de um usuário, Ana. Inicialmente Ana gera aleatória e independentemente dois números primos grandes (ímpares) p RSA q e calcula o produto $n = pq$. Ana também escolhe um número e tal que:

$$1 \langle e \langle \phi(n) = (p-1)(q-1) \quad e \quad mdc \quad (e,(p-1)(q-1)) = 1,$$

onde $\phi(n)$ é chamada função de Euler[31], denota a quantidade de números inteiros a em $\{1,2,\ldots,n\}$ com mdc $(a,n)=1$.

Note que e é sempre ímpar, dado que $p-1$ é par. Ana calcula um número inteiro d tal que:

$$1 \langle d \langle (p-1)(q-1) \quad e \quad de \quad \equiv \quad 1 \bmod (p-1)(q-1)$$

Dado que o mdc $(e,(p-1)(q-1))=1$, esse número d existe. Ele pode ser calculado pelo algoritmo de Euclides [31].

A chave pública de Ana é o par (n,e). Sua chave privada é d**.** O número n é chamado de módulo RSA, e é chamado de expoente de cifragem e d é chamado expoente de decifragem. Note que a chave privada d pode ser calculada do expoente de cifragem e, se os fatores primos p e q de n forem conhecidos. Portanto, se um usuário não autorizado for capaz de descobrir a fatoração de n em números primos, então ele facilmente pode descobrir a chave privada d de Ana

Para exemplificar, considere que Ana escolhe os fatores primos $p=11$ e $q=23$. Então $n=253$ e $(p-1)(q-1)=10\times 22=4\times 5\times 11$. O menor de todos os e possíveis é 3, dado que $mdc\,(3, 253)=1$. O algoritmo de Euclides resulta $d=147$.

Cifragem RSA

Para descrever a cifragem RSA, considere-se o espaço de texto claro consistindo de todos os números inteiros m, tal que $0\le m\le n$. O texto cifrado c é obtido pelo seguinte cálculo sobre o texto claro $m, c=m^e \bmod n$. Se Ana conhece a chave pública do destinatário (n,e) ela pode cifrar sua mensagem m.

Para exemplificar-se a cifragem, supõe-se $n=253$ e $e=3$. Então, o espaço de texto claro é $\{0,1,\ldots,252\}$. Cifrando-se o texto claro, número inteiro $m=26$, obtém-se $c=26^3 \bmod 253=119$.

O RSA pode ser utilizado como uma cifra de bloco. As cifras de bloco cifram blocos de comprimento fixo. Um sistema é chamado de cifra de bloco se seu espaço de texto claro e texto cifrado são o conjunto $\sum^k$ de palavras de um comprimento fixo k em um alfabeto $\sum$. O comprimento k do bloco é um número inteiro positivo. Um exemplo simples de cifra de bloco é o código secreto de Cesar. Ele tem o comprimento de bloco 1.

Como cifra de blocos opera sobre blocos de dados de comprimento fixo, quando se está cifrando ou decifrando um fragmento de dados, o algoritmo criptográfico divide o texto claro em blocos e opera sobre cada bloco de maneira independente. A cifragem e decifragem devem ocorrer em tamanhos fixos de blocos. No caso do RSA, se o módulo tiver 1.024 bits, o bloco terá 128 bytes de comprimento. Assim cada bloco de dados deveria ter o tamanho de 128 bytes, e, por exemplo, se o último bloco tivesse um comprimento menor que 128 bytes, bytes adicionais seriam necessários como enchimento para completar o tamanho do bloco.

Para cifragem RSA da chave de sessão, a chave secreta do DES utilizada para cifrar simetricamente os dados, que tem um tamanho normal de 64 bits ou 8 bytes, seria necessário

o enchimento do bloco que conterá a chave com 120 bytes. O esquema mais comum de enchimento de bloco do *RSA* é definida pelo *Public Key Cryptograhy Standard #1* *PKCS* $\neq 1$ [18].

No caso específico do *SET*, este protocolo especifica que o bloco de dados que contém a chave secreta do *DES* deve ter um enchimento denominado *Optimal Asymmetric Encryption Padding* (*OAEP*) [18] anteriormente ao processo de cifragem *RSA*. Nos capítulos 4 e 5 estão maiores detalhes da composição e forma do enchimento de bloco do *RSA* no formato *OAEP*.

Decifragem RSA

A decifragem no sistema *RSA* está baseada no seguinte teorema [7]:

$$Se \quad c = m^e \mod n, \quad então \quad (m^e)^d \mod n = m = c^d \mod n$$

Prova.

Dado que $ed = 1 \bmod (p-1)(q-1)$, existe um inteiro l com $ed = 1 + l(p-1)(q-1)$.

Portanto, $(m^e)^d = m^{ed} = m^{1+l(p-1)} = m(m^{(p-1)(q-1)})^l$

Segue que $(m^e)^d \equiv (m^{(p-1)})^{(q-1)l} \equiv m \mod p$.

Se p não for um divisor de m, então essa congruência obedece ao pequeno teorema de Fermat [7]. Caso contrário, a afirmativa é trivial porque ambos os lados da congruência são iguais $0 \bmod p$. De modo análogo, vemos que $(m^e)^d \equiv m \mod q$

Dado que p e q são números primos distintos, obtemos $(m^e)^d \equiv m \bmod n$.

Isso demonstra que o sistema *RSA* é, de fato, um cripto-sistema, pois para cada função de cifragem, existe uma função de decifragem [7].

Nos exemplo de cifragem anterior escolheu-se $n = 253$, $e = 3$ e $d = 147$. Calcula-se o texto cifrado $c = 119$ a partir do texto claro $m = 26$. A decifragem $m = c^d \bmod n = 119^{147} \bmod 253 = 26$, o qual é o texto claro original.

Finalmente, conforme afirmado anteriormente, a segurança do *RSA* reside na dificuldade de se fatorar um número positivo múltiplo em dois números primos grandes. Para ilustrar essa dificuldade para fatorar um inteiro n, quando o número n possui 129 algarismos decimais, seria necessário gastar 5 mil MIPS-ano, um MIPS-ano significa usar um computador por um ano executando um milhão de instruções por segundo, utilizando um dos algoritmos mais rápido que se conhece, chamado *QS* (*Quadratic Sieve*), de autoria de C. Pomerance [20]. Um outro algoritmo para fatoração chamado *NFS* (*Number Field Sieve*), de autoria de K. Lenstra, H. W. Lenstra, M. S. Manasse, e J. M. Pollard, é mais rápido que o *QS* para números com mais de 350 bits , por ser mais rápido assintoticamente que QS [21]. Em fevereiro de 1999 foi estabelecido um recorde de fatoração de uma chave *RSA* de 465 bits (140 decimais) pela execução do algoritmo NFS distribuído em centenas de estações de trabalho, durante vários meses.

Mais recentemente Adi Shamir apresentou um computador específico para fatoração [22], baseado em dispositivos opto-eletrônicos. Com uma implementação do algoritmo *QS* (ou do *NFS*) o TWINKLE consegue analisar 100 milhões de inteiros e determinar em menos de 10 milisegundos quais destes são fatoráveis completamente sobre uma base dos primeiros 200 mil primos. Shamir afirma que assim torna viável a fatoração de inteiros de comprimento entre 565 e 665 bits; isto torna vulnerável os sistemas de comércio eletrônico que na sua maioria utiliza *RSA* com chaves de 512 bits. O SET utiliza chaves *RSA* de 1024 e 2048 bits.

Em 1996, 512 bits para o módulo n do *RSA* eram considerados razoavelmente seguros. Mas devido aos eventos mencionados e devido aos algoritmos *QS* e *NFS* , hoje se recomenda no mínimo 756 bits. Para longo prazo, recomenda-se desde já adotar 1024 bits.

É importante mencionar que até hoje os pesquisadores não conseguiram descobrir qualquer outro ponto fraco neste sistema de criptografia.

2.2.2. Assinatura de Mensagens

No cripto-sistema *RSA* verificou-se que estava estabelecida a Autenticação do Destinatário, pois só o autêntico destinatário da mensagem cifrada c possui o valor d, ou seja sua chave privada, utilizado para converter c em m e assim o remetente tem certeza que só o verdadeiro destinatário pode ter recuperado m. Entretanto, o destinatário não tem certeza de que o autêntico remetente enviou c para ele, pois a sua chave pública (e, n) pode ser utilizada por qualquer pessoa.

As autenticações do destinatário, autenticação do remetente e integridade são propriedades muito importantes para o tráfego de informações em redes de computadores, como por exemplo, a Internet. Para se estabelecer a autenticação de remetente são utilizadas as assinaturas digitais que possuem propriedades similares às das assinaturas feitas à mão. Se um usuário, por exemplo, Ana, assinar um documento com assinatura à mão, todos que virem o documento e que conhecerem a assinatura de Ana poderão verificar que Ana de fato assinou o documento. Em muitas situações, os documentos eletrônicos também devem ser assinados, para proporcionar a autenticidade de remetente.

A princípio, as assinaturas digitais funcionam como a seguir. Suponha que Ana queira assinar o documento m. Ela utiliza sua chave privada d e calcula a assinatura $s(d,m)$. Utilizando a chave pública correspondente e, Beto, o destinatário da mensagem assinada, pode verificar que $s(d,m)$ é, de fato, a assinatura de m. Tal esquema de assinatura é seguro, se ninguém puder produzir a assinatura $s(d,m)$ sem o conhecimento do d privado.

O sistema RSA pode também ser utilizado para gerar assinaturas digitais. Para tanto, Ana assina o documento $s = s(d,m) = m^d \bmod n$. Aqui, d é o expoente privado de Ana e n, público, é o módulo do *RSA*. Beto verifica a assinatura calculando $s^e \bmod n = m^{ed} \equiv m \bmod n$.

Por que isso é uma assinatura? Elevando o número de aparência aleatória s à potência e, Beto pode recuperar o documento m. Portanto, s pode ser considerado a $e-ésima$ raiz do documento m e, atualmente, cacular as raízes $e-ésimas$ de um inteiro $m \bmod n$ sem o conhecimento de d é inexequível. Mas Ana é a única pessoa que conhece d, então Ana deve ter calculado s e, por meio disso, assinado m [19].

A geração de chaves para assinaturas *RSA* é a mesma que a geração de chaves para cifragem *RSA*. Ana escolhe independentemente dois primos grandes aleatórios p e q e um expoente e.

com $1 \langle e \ (p-1)(q-1)$ *e* $mdc(e,(p-1)(q-1))=1$. Ana então calcula $n = pq$ *e* $d \ \varepsilon \ Z$ com $1 \langle d \langle (p-1)(q-1)$ *e* $de \equiv 1 \bmod (p-1)(q-1)$. Sua chave pública é (n,e) e sua chave privada é d.

Ana assina $m \in \{0,1,\ldots,n-1\}$

. O inteiro m pode ser uma mensagem curta ou um valor resumido de uma mensagem longa. Para assinar m, Ana calcula $s = m^d \bmod n$. A assinatura é s. Beto quer verificar a assinatura s. Ele obtém a chave pública de Ana (n,e) de algum diretório público ou certificado e recupera a mensagem assinada calculada $s = m^e \bmod n$. Agora ele conhece a mensagem assinada m. Como ele calculou m a partir de s, ele sabe que s é a assinatura de m. Ele não precisa conhecer m previamente. Mas ele tem certeza de que Ana gerou s. Dado seu conhecimento atual, s não pode ser calculado sem d, e d é o segredo de Ana. Qualquer um que conheça a chave pública de Ana pode verificar sua assinatura.

Exemplificando a assinatura digital, Ana escolhe $p = 11, q = 23$ e $e = 3$. Ela obtém $n = 253,\ d = 147$. A chave pública de Ana é (253,3). Sua chave privada é 147. Ana quer obter R$ 111,00 (cento e onze reais) de um caixa eletrônico. Ela assina 111,00 e calcula $s = 111^{147} \bmod 253 = 89$. O caixa automático calcula $m = s^3 \bmod 253 = 111$. A máquina sabe que Ana que retirar R$ 111,00 e ela pode provar isso a terceiros.

2.2.3. Assinatura com função *hash*

O resumo de mensagem pode ser, por exemplo, utilizado em assinaturas digitais. Os resumos de mensagens são obtidos através da utilização de uma função unidirecional denominada *Hash*. Seja $\sum^*$ um alfabeto, por função *hash* entendemos uma correspondência $h : \sum^* \to \sum^n,\ n \in N$. Assim as funções *hash* correspondem arbitrariamente sequências longas a sequências de comprimento fixo menores. Elas nunca são injetoras.

A correspondência que envia $b_1 b_2 \ldots b_3$ em $\{0,1\}^*$ a $b_1 \oplus b_2 \oplus b_3 \oplus \ldots \oplus b_k$ é uma função de *hash*. Ela corresponde, por exemplo, 01101 a 1. Por exemplo, ela envia uma sequência b para 1 se o número de algarismos 1 em b for ímpar e para 0 , caso contrário.

As funções *hash* podem ser geradas utilizando-se funções de compressão. Uma função de compressão é uma correspondência $h:\sum^m \rightarrow \sum^n, n, m \in N,\ m \rangle n$. Ela faz a correspondência entre sequências de comprimento fixo a sequências de comprimento menor. A correspondência que envia a palavra $b_1 b_2 \ldots b_{m\ m}$ em $\{0,1\}^*$ a $b_1 \oplus b_2 \oplus b_3 \oplus \ldots \oplus b_m$ é uma função de compressão se $m \rangle 1$.

As funções *hash* e de compressão utilizadas na criptografia devem possuir propriedades que garantam sua segurança. A seguir essas propriedades são descritas de modo informal. Seja $h:\sum^* \rightarrow \sum^n$ uma função *hash* ou $h:\sum^m \rightarrow \sum^n$ uma função de compressão. Denotamos o conjunto $\sum^m$ ou $\sum^n$ de argumentos de h por D. Se h for uma função *hash*, então $D = \sum^*$. Se h for uma função de compressão, então $D = \sum^m$ **.** No uso em criptografia $h(x)$ deve ser fácil de calcular para todo $x \in D$.

A função h é chamada unidirecional se for inexequível inverter h; isto é , calcular uma imagem inversa x tal que $h(x) = s$ para determinada imagem s. De maneira intuitiva, inexequível significa que qualquer algoritmo que na entrada de $s \in \sum^n$ tente calcular x com $h(x)$ quase sempre fracassará, porque utiliza espaço ou tempo demais. Não se sabe se existem funções de uma direção. Existem funções, porém, que são fáceis de se avaliar, mas para as quais não se conhecem algoritmos de inversão eficientes e que, portanto, podem ser utilizadas como funções de uma direção.

Uma colisão de h é um par $(x, x') \in D^2$ para a qual $x \neq x'$ e $h(x) = h(x')$. Existem colisões de todas as funções *hash* e funções de compressão porque elas não são injetoras. Uma colisão da função *hash* do exemplo visto, anteriormente em $\{0,1\}^* \rightarrow b_1 \oplus b_2 \oplus b_3 \oplus \ldots \oplus b_k$ é um par de sequências distintas, ambas as quais possuem um número ímpar de algarismos um, como (111,001). A função h é chamada de pouco resistente à colisão se for inexequível calcular uma colisão (x, x') para determinado $x \in D$. A função h é chamada de (muito) resistente à colisão se for inexequível calcular qualquer colisão (x, x') de h. Para algumas aplicações

criptográficas as funções pouco resistentes à colisão podem ser utilizadas. Um exemplo dessa utilização é para cálculo de valor de integridade de arquivos eletrônicos em computadores, onde o valor de *hash* é armazenado em cartão magnético. Uma vez que a função *hash* resistentente à colisão não permite o cálculo de um arquivo imagem que produza o mesmo valor *hash*, para conferir-se se não houve alteração no arquivo, o valor *hash* é recalculado e comparado com o valor *hash* armazenado no cartão. Entretanto, para utilização em assinaturas digitais é necessário o uso de funções *hash* muito resistentes à colisão.

É possível construir uma função de compressão a partir de uma função de cifragem que pareça ser resistente à colisão, contanto que o esquema de cifragem seja seguro. Descreve-se a seguir como se faz isso através de utilização de um sistema de chave secreta. Utilizando-se um cripto-sistema com espaço de texto claro, x, espaço de texto cifrado e espaço de chave $\{0,1\}_n$. As funções de cifragem são $e_k \{0,1\}^n \rightarrow \{0,1\}^n$, $k \in \{0,1\}^n$. Os valores de *hash* têm comprimento n. Escolhe-se $n \geq 128$. A função *hash* $h : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ pode ser definida como se segue:

$$h(k,x) = e_k(x) \oplus x$$

$$h(k,x) = e_k(x) \oplus x \oplus k$$

$$h(k,x) = e_k(x \oplus k) \oplus x$$

$$h(k,x) = e_k(x \oplus k) \oplus x \oplus k$$

Contanto que o cripto-sistema seja seguro, essas funções *hash* são resistentes à colisão. Infelizmente, não se conhece nenhuma prova para essa declaração.

Como já citado anteriormente, as funções *hash* podem ser utilizadas para verificar se uma mensagem ou arquivo foi modificado. O valor *hash* é calculado e enviado ou armazenado separadamente. A integridade da mensagem ou arquivo é verificado calculando-se o valor *hash* da mensagem ou arquivo real e comparando-o com o valor *hash* recebido ou armazenado. Se os dois valores *hash* forem iguais, então se considera que a mensagem ou arquivo não foi modificado.

O protocolo *SET* especifica o *Secure Hash Algorithm* (*SHA*−1), definido no padrão *Federal Information Processing Standards Publication – FIPS 180-1*[23], como algoritmo de resumo de mensagens requerido para uso no *Digital Signature Standard* (*DSS*), conforme definido no padrão *Federal Information Processing Standards Publication – FIPS 186*[24] . Para uma mensagem de comprimento $\langle 2^{64}$ bits, o *SHA*−1 produz uma representação condensada da mensagem de 160 bits chamada um resumo de mensagem. O *FIPS 180-1* apresenta o algoritmo *SHA*−1, a forma de enchimento dos blocos de mensagem a serem resumidas e exemplos de aplicação do algoritmo.

Se não somente a integridade de um documento, mas também a autenticidade precisa ser demonstrada, então podem ser utilizadas funções *hash* parametrizadas.

A função *hash* parametrizada é uma família de funções *hash* $\{h_k : k \in K\}$. Aqui, K é um conjunto. Ele é chamado de espaço de chave de h. Uma função *hash* parametrizada também é chamada de Código de Autenticação de Mensagem ou *MAC* (*Message Authentication Code*). Para exemplicar, considera-se a função *hash* $g : \{0,1\}^* \rightarrow \{0,1\}^4$. Ela pode ser transformada no *MAC* $h_k : \{0,1\}^* \rightarrow \{0,1\}^4,\ x \rightarrow g(x) \oplus k,$ com espaço de chave $\{0,1\}^4$. Segue-se um exemplo da utilização do *MAC* para garantia de autenticidade. Supondo-se que a professora Ana envia uma lista com os nomes de todos os alunos que passaram no curso de criptografia por *e-mail* à secretária de sua faculdade. É importante que a secretaria da faculdade fique convencida de que esse *e-mail* é autêntico. Para a prova de autenticidade, é utilizado um *MAC* $\{h_k : k \rightarrow K\}$. Ana e a secretaria da faculdade trocam uma chave secreta $k \in K$. Com sua lista x, Ana também envia o valor *hash* $y = h_k(x)$ à secretaria da faculdade. Beto, o secretário, também pode calcular o valor *hash* $y' = h_k(x')$ da mensagem recebida x'. Ele aceita x' se $y = y'$. O protocolo deste evento prova a autenticidade se, sem o conhecimento de k for inexequível calcular um par $(x', h_k(x'))$ a partir do par $(x, (h_k(x))$ com $x \neq x'$.

O protocolo *SET* especifica o *Keyed-Hash Mensagem Authentication Code* (*HMAC*), definido no padrão *Federal Information Processing Standards Publication – FIPS 198*[25], como algoritmo de autenticação de mensagens baseada em função criptográfica *hash*.

Neste ponto, pode-se abordar como se processa a assinatura digital com função *hash.* Na seção onde foi abordada assinatura digital no sistema *RSA*, viu-se como os documentos m

que são inteiros em $\{0,1,\ldots,n-1\}$ são assinados. Verificando a assinatura, o destinatário da mensagem assinada também obtém o documento que foi assinado. Se Ana, por exemplo, quiser assinar um documento arbitrariamente longo x, ela utiliza uma função *hash* resistente à colisão publicamente conhecida, $h:\{0,1\}^* \to \{0,\ldots,n-1\}$. Como h é resistente à colisão, h também é uma função unidirecional. Na prática, h é construída utilizando-se uma função *hash* padrão resistente à colisão. A assinatura do documento x é $s = h(x)^d \bmod n$. A partir dessa assinatura, somente o valor de *hash* $h(x)$, mas não o documento x, pode ser reconstruído. Portanto Beto, o destinatário da mensagem, pode verificar a assinatura x somente se ele também conhecer o documento x. Depois de Ana calcular a assinatura s de x, ela envia s com o documento x para Beto. Beto calcula $m = s^e \bmod n$ e compara esse número com o valor *hash* de x. Como a função *hash* é pública, Beto pode calcular esse valor *hash*. Se m e $h(x)$ forem iguais, Beto aceita a assinatura. Caso contrário, ele a rejeita.

Esse procedimento torna a falsificação existencial computacionalmente inviável. Supondo-se que um usuário não autorizado tenha escolhido a assinatura s. Como ele deve enviar um documento x com s para Beto, ele deve criar um x de modo que $h(x) = s^e \bmod n$. Isso é exatamente o que Beto calcula quando tenta verificar a assinatura, então x é uma imagem inversa de $m = s^e \bmod n$ sob h. Como a função *hash* h é unidirecional, Beto não pode calcular tal x. Finalmente, o usuário não autorizado não pode substituir o documento x assinado por Ana por outro documento x', pois o par (x, x') é uma colisão de h e h é resistente à colisão [7].

2.3. Sumário de Criptografia

Para finalizar este capítulo é apresentado um sumário de utilização da criptografia abordada. Os diagramas apresentados na Figuras 2.3.e 2.4. fornecem uma visão geral do processo de criptografia quando Ana deseja assinar um texto claro, e enviá-lo a Beto como uma mensagem criptografada. Os passos numerados nos diagramas são detalhados nas tabelas abaixo.

O sumário do processo de cifragem na Figura 2.3 consiste dos passos constantes do quadro Descrição do sumário de cifragem, a seguir:

De forma semelhante, o sumário do processo de decifragem na Figura 2.4 consiste dos passos constantes do quadro Descrição do sumário de decifragem, exposto mas adiante.

Passo	Descrição do sumário de cifragem
1	Ana processa seu texto claro através de um algoritmo unidirecional de *hash* SHA-1 para produzir um valor único conhecido como resumo de mensagem. Este valor é uma espécie de impressão digital da mensagem e irá ser usado posteriormente para testar a integridade da mensagem.
2	Ela então cifra com o cripto-sistema assimétrico RSA o resumo de mensagem com sua chave privada de assinatura para produzir a assinatura digital da mensagem.
3	Em seguida, ela gera aleatoriamente uma chave secreta e a utiliza para cifrar com o cripto-sistema simétrico *DES* o texto claro, sua assinatura digital e uma cópia de seu certificado digital que contem sua chave pública de assinatura. Para decifrar o texto claro, Beto irá precisar uma cópia segura da chave secreta do *DES* usada por Ana.
4	O certificado de Beto, que Ana deve ter obtido anteriormente para o início da comunicação segura com ele, contem uma cópia da chave pública de cifragem *RSA*. Para garantir uma transmissão segura da chave secreta *DES* usada por Ana, ela cifra-a com a chave pública de cifragem *RSA* de Beto. A chave secreta cifrada, referenciada como envelope digital irá ser enviada a Beto junto com o próprio texto cifrado.
5	Ana envia a mensagem a Beto consitindo do seguinte: o texto claro cifrado simetricamente, assinatura digital e certificado, bem como sua chave secreta assimetricamente cifrada (o envelope digital).

Passo	Descrição do sumário de decifragem
6	Beto recebe a messagem de Ana e decifra o envelope digital com sua chave privada de decifragem *RSA* para recuperar a chave secreta *DES* utilizada por Ana.
7	Ele usa a chave secreta para decifrar o texto claro, a assinatura de Ana e seu certificado.
8	Ele decifra a assinatura com a chave pública de assinatura *RSA* de Ana, que ele adquire do seu certificado. Ele recupera o resumo de mensagem original do texto claro.

9	Ele processa a decifragem do texto claro através do mesmo algoritmo unidirecional de *hash* $SHA-1$ usado por Ana e produz um novo resumo de mensagem do texto claro decifrado.
10	Finalmente, ele compara seu resumo de mensagem com aquele obtido da assinatura digital de Ana. Se eles são exatamente o mesmo, ele tem a confirmação que o conteúdo da mensagem não sofreu alteração durante a transmissão e que ele foi assinado pela chave privada de assinatura *RSA* de Ana. Se eles diferem, então a mensagem ou foi originada em outro lugar ou então foi alterada depois que foi assinada. Neste caso, Beto toma uma ação apropriada tal como notificando Ana ou descartando a mensagem.

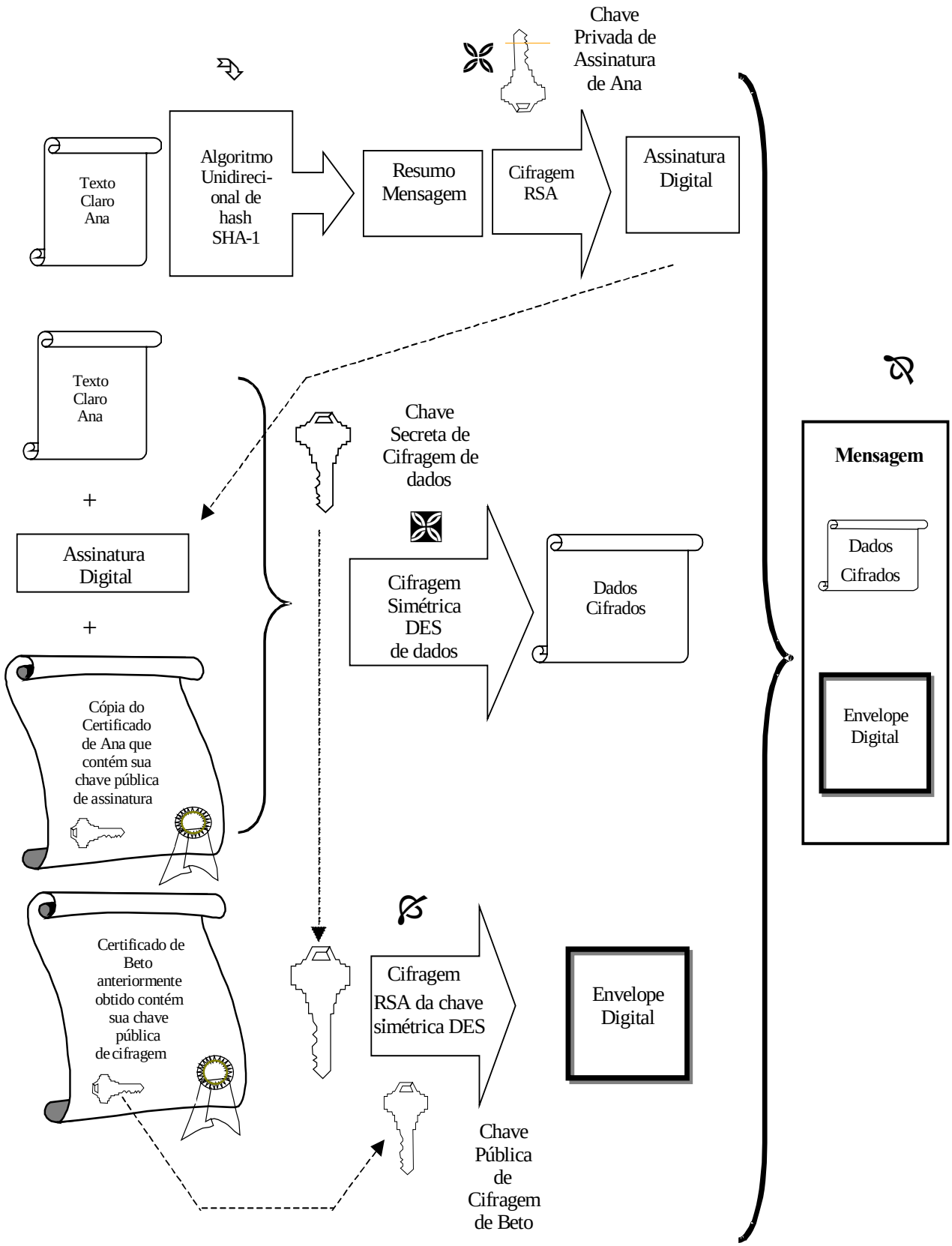

Figura 2.3. Diagrama do Sumário do Processo de Cifragem

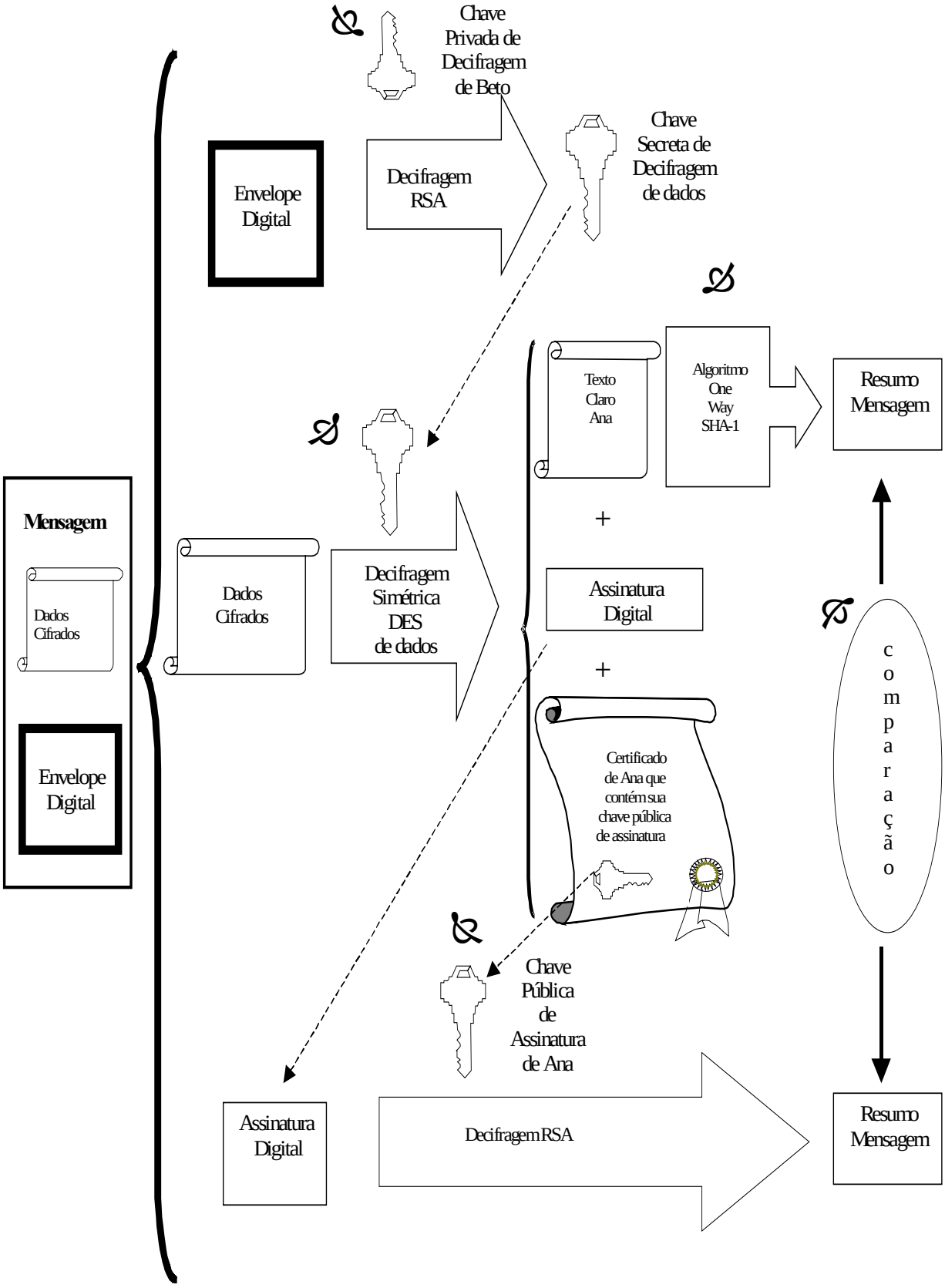

Figura 2.4. Diagrama do Sumário do Processo de Decifragem

Capítulo 3

Apresentação do Set

Este capítulo apresenta o *SET* , como ele se insere na compra eletrônica, seus participantes e como eles se conectam num ambiente *SET* . As principais características da segurança da informação providas pelo protocolo são expostas. Os aspectos relevantes de gerenciamento de certificados digitais, um pilar fundamental da segurança do protocolo, são discorridos. A adaptabilidade a ambientes onde certificados não são utilizados é rapidamente abordada, como também medidas de interesse de segurança de chaves e dados usados pelo protocolo.

3.1. Proteção de Informações no Comércio Eletrônico

A Internet, ou *World Wide Web*, como nenhum outro meio, facilitou muito para os consumidores comprarem, transferirem dinheiro, e pagarem contas com um simples apertar de botões. Entretanto, o preço que pagamos por essa facilidade é o aumento de oportunidades de fraudes.

No início dos anos 90 os bancos americanos pressionados tanto pelos consumidores quanto pelos Comerciantes interessados no comércio eletrônico via Internet, começaram a pressionar a Visa e a Mastercad para desenvolver um padrão seguro para utilização de cartões sobre qualquer canal inseguro, tal como a Internet. A Visa e a Microsoft responderam ao clamor dos bancos publicando um padrão em setembro de 1995. A especificação *Secure Transaction Technology* (*SST*) foi disponibilizada no *site Web* da Visa para *download* pelas partes interessadas. Nesse mesmo tempo, a MasterCard e seus parceiros, Natscape, IBM, Cybercash, e GTE, tinham desenvolvido o *Secure Electronic Payment Protocol* (*SEPP*) como uma especificação proposta e disponibilizada no *site* da MasterCard na *Web.* A MasterCard esperava que o *SEPP* pudesse estar em uso nas transações na Internet em abril de 1996. No final de 1995, os bancos que operavam com ambos os cartões Visa e MasterCard estavam

preocupados com a necessidade de utilização de dois padrões separados para a realização do mesmo trabalho. Os bancos persistiram, e finalmente forçaram a Visa e a MasterCard a trabalhar juntas num único padrão. Assim, sob acordo, a Visa e a MasterCard, em conjunto com a GTE, IBM Microsoft, Netscape Communications Corp., SAIC, Terisa Systems, VeriSign, e RSA Data Security, formaram o Consórcio SET. Sua meta era resolver as diferenças e conflitos entre o STT e SEPP e desenvolver um novo padrão unificado. O trabalho do Consórcio SET, o *Secure Electrocnic Transactions* (SET), Versão 0.0, foi publicado aos desenvolvedores de aplicações na forma de rascunho em 24 de junho de 1996. Para esta versão inicial do SET foi estabelecido o prazo de janeiro de 1997 para efetivação de melhorias necessárias para mudança para a Versão 1.0, e março de 1997 como prazo proposto para teste da nova versão. Em 21 de abril de 1997, a versão 0.2 do SET foi publicada contendo os requisitos de melhoria que satisfizeram as necessidades adicionais de companhias não Visa/MasterCard, tais como American Express, Japan Comerce Bank (JCB), e Novus/Discover. Em 31 de maio de 1997 a atual versão 1.0 do SET foi publicada.

O Consórcio SET estabeleceu a *Secure Electronic Transaction Mark*, a SETMark com a finalidade de indicar a certificação bem sucedida de vendedores de *software* e *sites* de Comerciantes na *Web*, para prover aos consumidores o conhecimento de que eles estavam realizando transações usando o SET. A SETMark é licenciada para utilização de qualquer vendedor ou bandeira de cartão que é certificada como obediente ao protocolo SET.

As especificações das transações seguras eletrônicas do SET fornecem uma estrutura para proteger contra fraudes os cartões de pagamento utilizados em transações do comércio eletrônico pela Internet. Como o SET opera em diversos tipos de ambientes de pagamento baseados em cartão, o “termo cartão de pagamento” é usado ao longo deste estudo para coletivamente se referir a qualquer dos seguintes: cartão de crédito, cartão de débito e cartão de banco. O SET protege os cartões de pagamento, assegurando a confidencialidade e a integridade dos dados do portador do cartão, ao mesmo tempo, oferecendo um meio de autenticação para o cartão.

Os mecanismos de transporte de informação usados para implementar o SET estão fora do escopo deste estudo, mas são reconhecidas duas classes: interativo e não interativo. A *World Wide Web* é um mecanismo interativo, e correio eletrônico ou postal são mecanismos não interativos.

O sistema SET é composto de uma coleção de entidades envolvidas no comércio eletrônico. A coleção consiste de:

- **Portador de cartão (*Cardholder*)**, um proprietário autorizado de um cartão de pagamento expedido por um Emissor e registrado para executar comércio eletrônico;

- **Comerciante (*Merchant*)**, um comerciante provendo bens, serviços, e/ou informações que aceita pagamentos por eles eletronicamente e pode prover serviços de vendas com entrega de produtos por meios tradicionais ou entrega eletrônica;

- **Emissor (*Issuer*)**, uma instituição financeira, normalmente companhias de cartão de crédito ou de pagamento, que emite ou dá suporte à emissão de produtos de cartão de pagamento a indivíduos;

- **Adquirente (*Acquirer*)**, uma instituição financeira, normalmente banco, que apoia os comerciantes pelo fornecimento de serviços para processamento de transações de cartão de pagamento;

- **Portal de Pagamento (*Payement Gateway*)**, um sistema que provê os serviços de comércio eletrônico aos comerciantes em suporte ao Adquirente, e interfaces com o Adquirente para permitir as transações de autorização e captura;

- **Marca (*Brand*)**, franqueador de sistemas de pagamento / instrumentos;

- **Autoridade de Certificação (*Certificate Authority* - CA)**, um agente de uma ou mais marcas de cartão de pagamento que provê a criação e distribuição de certificados eletrônicos para portadores de cartão, comerciantes, e portais de pagamento, é uma espécie de cartório eletrônico virtual; e

- **Rede financeira de marcas de cartão de pagamento**, a rede privada existente operada por uma marca de cartão de pagamento que liga Adquirentes a Emissores.

3.2. O SET e a Compra Eletrônica

O escopo do SET é limitado ao processo de pagamento e aos serviços de segurança necessários para dar apoio aos aspectos de pagamento de uma compra eletrônica. Para prover estes serviços, o SET define não só o protocolo de pagamento eletrônico, mas também o processo de administração de certificado.

A Figura 3.1 abaixo exibe os participantes de sistemas de pagamento do SET e suas interações.

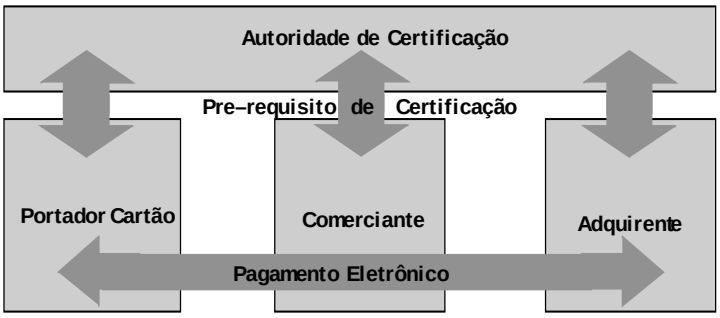

Figura 3.1: Participantes do Sistema de Pagamentos

A compra eletrônica se dará tipicamente pelas fases abaixo relacionadas. O SET suporta três destas fases abaixo relacionadas: a) autorização de pagamento e remessa; b) confirmação e investigação; e c) reembolso ao comerciante.

- Navegando e comprando;
- Seleção de Comerciante e de Item;
- Pedido e Negociação;
- Seleção de Pagamento;
- SET → Autorização de Pagamento e Remessa;
- SET → Confirmação e Investigação;
- Entrega de Bens;
- SET → Reembolso ao Comerciante;

No processo de compra eletrônica o Portador do cartão decide começar a compra por um item. Os itens comprados podem ser bens tangíveis, mídias eletrônicas (por exemplo, informação, software, etc.), ou serviços. O processamento da compra eletrônica se dá conforme a Tabela 3.1 abaixo, onde a relação do SET com as outras fases do modelo de compra eletrônica é descrito. O SET enfoca as fases 5, 6, 7, e 8.

Fase	Descrição
1	O Portador do cartão navega nos itens. Isto pode ser realizado numa variedade de modos, como: • Usa um navegador para examinar um catálogo on-line na página de um comerciante na *World Wide Web*; • Examina um catálogo provido pelo comerciante em um CD-ROM; ou • Examina um catálogo em papel.
2	O Portador do cartão seleciona os itens a serem comprados de um comerciante.
3	O Portador do cartão é apresentado a um formulário de pedido que contém a lista de itens, seus preços, e um preço total incluindo a remessa, manuseio, e impostos. Este formulário de pedido pode ser entregue eletronicamente pelo servidor do comerciante ou pode ser criado no computador do Portador do cartão através do *software* de compra eletrônica. Alguns comerciantes *on-line* também podem permitir a possibilidade de um Portador do cartão negociar o preço dos itens (tal como apresentando identificação de comprador freqüente ou informação sobre o preço de um competidor).
4	O Portador do cartão seleciona os meios de pagamento. O SET se destina ao caso onde um cartão de pagamento é selecionado.
5	O Portador do cartão envia para o comerciante um pedido completo junto com os meios de pagamento. No SET, o pedido e as instruções de pagamento são digitalmente assinados pelos Portadores de cartão que possuem certificados.
6	O comerciante solicita autorização de pagamento da instituição financeira do Portador do cartão, via Adquirente. Se a autorização tiver sucesso, o comerciante pode enviar confirmação do pedido por algum meio de comunicação que não utilize o SET.
7	O comerciante envia os bens ou executa os serviços do pedido.
8	O comerciante solicita o pagamento da instituição financeira do Portador do cartão via Adquirente.

Tabela 3.1 : Processamento da Compra Eletrônica

3.3. Participantes do SET e seu papel

A arquitetura do SET é projetada para proteger a transmissão de informação financeira envolvida em uma transação de pagamento entre um Portador de cartão, Comerciante, e Adquirente.

Portador de Cartão de Pagamento

O Portador de cartão de pagamento do SET é representado no SET por uma estação de trabalho. Isto proporciona ao Portador de cartão a flexibilidade para fazer compras e conduzir as negociações com os sistemas dos comerciantes que oferecem itens para venda. A estação de trabalho deve dar suporte a todas as fases do modelo de compra eletrônico descritas na Tabela 3.1. Em apoio ao SET, a estação de trabalho deve ter a funcionalidade para executar o processo de pagamento.

A interface, ou ligação, primária do Portador de cartão no SET é com os sistemas dos comerciantes. Esta interface dá suporte a porção do protocolo de pagamento do Portador de cartão, que permite ao Portador de cartão iniciar o pagamento, executar investigações e receber reconhecimento de pedido e status.

O Portador de cartão também tem uma interface indireta com o Adquirente através do sistema do Comerciante. Esta interface dará suporte aos campos de dados cifrados que são enviados via Comerciante ao Adquirente, porém só podem ser decifrados pelo Portal de Pagamento. Isto permite ao Adquirente intermediar as interações entre o Portador de cartão e Comerciante, e por isso prover serviços de segurança ao Portador de cartão. Estes serviços de segurança garantem que o Portador de cartão está negociando com um comerciante válido, aprovado pelo cartão de pagamento.

Sob as políticas estabelecidas pela marca do cartão de pagamento, o Portador de cartão também possui interface com a Autoridade de Certificação do Portador de cartão (CCA) para pedir e renovar certificados de chave pública que dão suporte às funções de segurança do comércio eletrônico. A execução das funções criptográficas executadas pelo sistema do Portador de cartão em módulos criptográficos de *hardware* é recomendada, mas não é

requisito obrigatório. A geração de chave secreta e o armazenamento usando módulos criptográficos de *hardware* resistentes tais como cartões inteligentes são bastante encorajados.

Em adição, o sistema do Portador de cartão dará suporte aos serviços de segurança (integridade, autenticação e gerenciamento de certificado, como prescrito pelo SET), e também dará suportar à compra, seleção de pagamento e funções de comunicações.

Comerciante do SET

O sistema de computador do comerciante SET provê uma interface conveniente ao Portador de cartão para o suporte de pagamentos eletrônicos. Em adição, o comerciante tem interface com o Adquirente e usa o protocolo de pagamento para receber os serviços de autorização e captura de transações eletrônicas de pagamento. O comerciante fará interface com a Autoridade de Certificação do Comerciante (MCA) para pedir e renovar certificados de chave pública que suportam as funções de segurança do comércio eletrônico.

O comerciante dará suporte aos protocolos do SET para a autorização de transações de comércio eletrônicas iniciadas pelo Portador de cartão. O sistema do comerciante também deve dar suporte às capturas. Em adição, o sistema do comerciante deverá apoiar os serviços de segurança (integridade, autenticação e gerenciamento de certificado). Os sistemas do comerciante darão suporte à compra, seleção de pagamento, e funções de comunicações. A execução de funções criptográficas por módulos criptográficos de *hardware* é fortemente recomendada, mas não são requisitos obrigatórios. A geração da chave privada e armazenamento usando módulos criptográficos de *hardware* resistentes como cartões inteligentes é fortemente encorajada. As exigências da marca de cartão de pagamento para uma implementação específica e ambiente nos quais o servidor do comerciante deve operar ditarão as exigências para o uso suporte de *hardware* criptográfico.

Portal de pagamento

O sistema do portal de pagamento é operado pelo Adquirente. Ele irá prover serviços de comércio eletrônico aos comerciantes em suporte ao Adquirente, e terá interface com a rede

financeira do cartão de pagamento para dar suporte à autorização e captura de transações. A interface da rede financeira do cartão de pagamento é muito semelhante à interface que dá suporte aos Adquirentes. O Portal de Pagamento também terá interface com a Autoridade de Certificação do Portal de Pagamento (PCA) para pedir e renovar certificados de chave pública para apoio às funções de segurança do comércio eletrônico. Ele irá apoiar a distribuição da Lista de Revogação de Certificados (CRLs) em nome da marca e instituição financeira. Serão executadas funções criptográficas em módulos de *hardware* criptográficos. Em adição, geração de chave privada e armazenamento usarão módulos de *hardware* criptográficos resistentes.

Adquirente

Um Adquirente é a instituição financeira (ou seu agente) que dá suporte à atividade do comerciante através de relacionamentos de conta com os mesmos. O Adquirente é responsável pela junção dos dados financeiros relacionados à transação para obter autorização de pagamento do Emissor do Portador de cartão.

Emissor

Um Emissor é a instituição financeira que estabelece uma conta para um Portador de cartão. O Emissor garante o pagamento de transações autorizadas usando cartão de pagamento. O processamento e a interface com o Emissor estão fora da área de atuação do SET.

A Terceira Parte para Processamento

Em alguns ambientes, os Emissores e Adquirentes podem escolher para assinar o processamento de transações de cartão de pagamento uma terceira parte independente.

Rede Financeira de Marca de Cartão de Pagamento

A rede financeira de marca de cartão de pagamento é a rede privada existente pela qual Adquirentes obtêm autorização para pagamento dos Emissores. VisaNet e Banknet são exemplos destes tipos de redes. Estas redes são protegidas por cada marca de cartão de pagamento e provêem interfaces para trocas de mensagens.

3.4. Serviços do SET

Aqui veremos um breve resumo dos serviços de segurança fundamentais e serviços de certificados providos na arquitetura do SET, através dos seguintes tópicos:

- Serviços;
- Certificados;
- Identificador de CRL de Marca .

3.4.1. Serviços de Segurança do SET

Confidencialidade

A confidencialidade de dados é a proteção da informação sensível e pessoal contra ataques não intencionais e intencionais para o acesso e/ou revelação não autorizados. Em ambientes não controlados, como redes inseguras, tais dados requerem cifragem e gerenciamento de chaves de cifragem associadas. O SET usa algoritmos de criptografia assimétricos e simétricos em conjunto com um envelope digital para prover a confidencialidade dos dados.

O SET é responsável pela confidencialidade dos dados de pagamento que ele precisa gerenciar. Onde a confidencialidade de dados não referentes a pagamento é necessária, a confidencialidade é provida no protocolo de mensagens pela inclusão de uma referência aos dados em lugar dos próprios dados. Por exemplo, o SET não troca a Descrição do Pedido, mas inclui um *hash* (resumo) da Descrição do Pedido no Pedido de Compra. Embora a

confidencialidade dos dados não referentes a pagamento estejam fora do escopo do SET, os desenvolvedores de sistema são encorajados a proteger estes dados.

Autenticação

O SET provê autenticação da origem de uma mensagem empregando algoritmos de verificação de assinatura digitais quando os certificados de assinatura estiverem disponíveis. A autenticação provê a garantia de que os dados recebidos foram enviados de fato pela parte que reivindica tê-los enviado. Assim, o receptor pode autenticar o remetente pela verificação dos dados recebidos. Isto é obtido usando assinaturas digitais e certificados de chaves públicas emitidos por uma Autoridade de Certificação.

As assinaturas digitais requerem uma terceira parte de confiança para atestar a autenticidade da chave pública usada para verificar a assinatura. O processo dita que uma terceira parte de confiança, uma Autoridade de Certificação (CA), provê um certificado eletrônico que atesta o fato que uma chave pública é "possuída" por uma certa entidade. Este certificado eletrônico (ele próprio assinado digitalmente pela CA) é armazenado pela entidade em seu computador. O sistema do receptor usa o certificado para verificar a chave pública do emissor. Neste ponto o receptor está seguro que:

- Os dados originais não foram alterados (integridade de dados);
- A mensagem só poderia ter sido assinada pelo proprietário daquela chave privada (autenticação da entidade); e
- Uma terceira parte de confiança tenha atestado o fato que o signatário é na realidade o proprietário daquele par de chaves.

Então, a singularidade da assinatura digital e o valor *hash* subjacente acoplado com a força do certificado de chave pública provêem um nível aceitável de garantia para autenticar o emissor e verificar que o emissor foi o originador dos dados assinados.

Os comerciantes e os Adquirentes irão verificar que um Portador de cartão está usando um número de conta válido. Também, indivíduos sem autorização que tenham roubado os números de conta de cartões de pagamento com datas de vencimento válidas podem tentar iniciar transações de comércio eletrônicas. Um mecanismo que liga um usuário a um número

de conta específico irá reduzir a incidência de fraude, e então o custo global de processo de pagamento. O certificado do Portador de cartão emitido pela CCA é a evidência de que a chave pública do Portador de cartão foi atrelada ao número de conta.

Um comerciante é submetido a verificação do acordo que detém com o Adquirente através da emissão de um certificado. Os Adquirentes irão autenticar o pedido de certificado do comerciante e, se apropriado, emitir um certificado através da sua MCA. Este certificado provê a garantia de que o comerciante detém um acordo válido com um Adquirente. Em essência, este é um "decalque eletrônico" que é equivalente ao decalque da marca do cartão na janela da loja do comerciante.

Os Portadores de cartão e Portais de pagamento irão autenticar os comerciantes verificando as assinaturas no certificado do comerciante e validando a cadeia do certificado.

Os certificados de Portal de pagamento são emitidos pela PCA de uma marca de cartão de pagamento. As marcas de cartão de pagamento autenticarão o pedido do certificado do Adquirente antes da emissão dos certificados. Estes certificados provêem a garantia de que um portal de pagamento foi autorizado pela Marca, Adquirente, ou Autoridade de Certificação de Marca Geo-política.

Os Comerciantes irão autenticar os Portais de pagamento pela verificação das assinaturas no certificado do portal de pagamento e pela validação da cadeia do certificado.

Desde que o de Portador de cartão usa a chave pública do Portal de Pagamento para cifragem da chave simétrica usada para cifrar a instrução de pagamento, o sistema do Portador de cartão necessita da habilidade de autenticar o Portal de Pagamento. O comerciante provê ao Portador de cartão o certificado de cifragem do Portal de Pagamento. O sistema do Portador de cartão irá validar este certificado e assim será assegurado que o Portal de Pagamento é legítimo e que a instrução de pagamento permanece confidencial.

Integridade

A Integridade de dados é a garantia de que os dados recebidos são na realidade os dados enviados. Isto é obtido por um valor de integridade que é gerado usando os dados

transmitidos. Os dados e o valor de integridade são transmitidos pelo emissor ao receptor. O receptor verifica que os dados não foram alterados durante transmissão validando o valor de integridade dos dados.

A integridade de dados é implementada pelo uso de uma função *hash* unidirecional. Uma função *hash* é aplicada aos dados apropriados para produzir um valor de integridade estatisticamente único, valor *hash*. As funções *hash* por elas mesmo não garantem a integridade absoluta dos dados. Para prover esta garantia, as funções *hash* precisam ser combinadas com uma quantidade secreta ou chave.

Conforme abordado no capítulo 2, as funções *hash* são diferentes das cifras simétricas e têm as seguintes propriedades:

- A função *hash* é um algoritmo público.
- A função *hash* é unidirecional, isto é, dado um valor de *hash*, não é possível recriar os dados originais.
- O valor *hash* é computado de tal maneira que não é possível identificar outros dados que darão o mesmo valor *hash*.

Assinatura Digital

Uma assinatura digital é definida como dados anexados, ou uma transformação criptográfica de uma unidade de dados que permite ao receptor da unidade de dados comprovar a fonte e integridade da unidade de dados, e também protege contra a falsificação, por exemplo, pelo receptor.

Na arquitetura do SET, uma assinatura digital é um valor *hash* cifrado usando a chave privada do emissor. O valor *hash* provê integridade dos dados dentro da mensagem; se os dados de pagamento são modificados, o valor *hash* será diferente, e aquela diferença pode ser detectada quando o receptor re-computar o *hash*. O *hash* é cifrado para assegurar que uma terceira parte não possa mudá-lo, desde que a cifragem de novo valor *hash* não seria possível sem a chave de cifragem privada.

O SET provê integridade empregando algoritmos criptográficos *hash* unidirecionais e assinaturas digitais para assegurar que uma mensagem não foi modificada em trânsito.

Assinatura Dual

O SET introduz uma nova aplicação de assinaturas digitais, um conceito denominado de assinaturas duais. No SET, assinaturas duais são usadas para ligar uma mensagem que contém informações do pedido enviadas pelo Portador de cartão ao Comerciante juntamente com as instruções de pagamento destinada ao Adquirente, contendo informações da conta, que devem ser conhecidas somente pelo Adquirente. A assinatura dual é gerada pela criação de dois resumos, um de cada parte da mensagem, concatenando-se estes dois resumos e então computando-se o resumo de mensagem do resultado da concatenação, ou seja a assinatura dual. Isto é cifrado com uma única chave privada de assinatura. O assinante deve usar um mecanismo de ligação para unir a mensagem destinada ao receptor ao resumo da outra mensagem que não lhe é destinada, enviando ao receptor, de forma a permitir que o receptor possa verificar que os dados se referem à mesma transação e também possa verificar a assinatura dual. Um dos receptores recebe a mensagem que lhe é destinada e computa seu resumo. Então ele pode verificar sua autenticidade pela concatenação do resumo por ele gerado e o resumo da mensagem destinada a outro destinatário por ele recebido, e em seguida computando o resumo do resultado da concatenação. Se este resumo gerado por último coincide com a assinatura dual decifrada, o receptor pode confiar na autenticidade da mensagem.

Ligação

O SET provê um mecanismo de ligação para examinar que uma mensagem contém uma referência a outra mensagem pela verificação de uma ligação embutida que usa algoritmo criptográfico *hash* unidirecional.

3.4.2. Certificados do SET

Gerenciamento de Certificados

O gerenciamento de Certificados consiste em um ou mais sistemas de CA de confiança que implementam a emissão e renovação de certificados de chave pública para Portadores de cartão, Comerciantes, e Adquirentes. Em adição, a arquitetura do SET define uma hierarquia de confiança de sistemas de CA que começam com uma CA Raiz (RCA - *Root Certificate Authority*), seguida por uma CA de Marca específica (BCA – *Brand Certificate Authority*) e uma CA de Marca Geo-política (GCA – *Geo-Political Certificate Authority*) opcional. Por exemplo, os sistemas de CCA (CCA – *Cardholder Certificate Authority*) têm interface com os Emissores para autenticar pedidos de certificados. São executadas funções criptográficas em módulos de *hardware* criptográficos. A geração de chave privada e armazenamento usa módulos de *hardware* criptográficos resistentes. O gerenciamento de certificados é executado em um ambiente físico seguro obediente aos padrões da marca do cartão de pagamento.

Uma assinatura digital criptograficamente liga os dados assinados com uma chave privada única, que é assumida estar sob o controle exclusivo do Portador de cartão, do Comerciante, Instituição financeira, ou CA, como apropriado. A chave privada é ligada matematicamente à chave pública do par de chaves. Assumindo que a chave privada não tenha sido comprometida, a assinatura digital também tem o efeito de ligar a chave pública aos dados. Porém, qualquer um pode gerar um par de chaves pública/privada, e assim é essencial que algum mecanismo seja estabelecido que ligue a chave pública à entidade de uma maneira confiável. Este é o propósito fundamental de um certificado - ligar uma chave pública a uma entidade unicamente identificada.

No caso do Portador de cartão, a assinatura do certificado implicitamente liga a chave pública ao Número de Conta Primário do Portador de cartão (PAN - *Primary Account Number*), porém o PAN é efetivamente escondido usando uma técnica de tal forma que só a CCA, o Portador de cartão, e o Emissor saibam o número de conta. O Portador de cartão passa o número da conta e uma variável secreta ao Adquirente, assim o Adquirente pode verificar o número de cartão contra o valor encoberto contido no certificado do Portador de cartão. Para proteger a confidencialidade do Portador de cartão, o nome do Portador de cartão não é

incluído no certificado. Com efeito, o número de conta encoberto é um pseudônimo do Portador de cartão.

Considerando que uma falsa Autoridade de Certificação pudesse ser montada para criar certificados que conteriam informações quase idênticas àquelas contidas em um certificado válido, a própria assinatura da CA será certificada como autêntica por uma CA de nível mais alto. A única exceção para esta exigência é a entidade Autoridade de Certificação Raiz. Ela é a única Autoridade de Certificação implicitamente confiada.

Certificado do Portador de Cartão

Uma função do Adquirente é assegurar que a chave privada usada para assinar um pagamento é, de fato, associada à conta do cartão de pagamento certa. Para evitar revelar o PAN (*Primary Account Number*) do Portador de cartão à terceiras partes, o número é escondido usando um mecanismo *hash* com chave como uma função obscurecente. O resultado desta função é o que é armazenado nos certificados do Portador de cartão.

A arquitetura do SET permite que os Portadores de cartão sem certificados de assinatura conduzam transações no SET. Esta é uma opção interina destinada somente ao uso em situações onde o banco emissor do Portador de cartão não provê serviços de certificado. Os Adquirentes devem escolher se aceitam ou não esta opção. Um sinalizador no certificado do portal de pagamento do Adquirente indica suporte para transações de Portador de cartão nas quais o Portador de cartão não tem nenhum certificado.

Os *software* do Portador de cartão e do portal de pagamento usarão uma extensão do certificado X.509[26], **CardCertRequired**, um sinalizador booleano ligado como verdadeiro, para assegurar que estejam incluídos os certificados nas transações conforme requeridos. Marcas que aceitam Portadores de cartão sem certificados devem remover tal suporte reemitindo os certificados de Portal de pagamento e omitindo esta extensão, ou ligando este sinalizador booleano como falso. Se um Portador de cartão tiver certificados disponíveis, o *software* deveria somente executar transações assinadas.

A aceitação dos Portadores de cartão com certificados é obrigatória: os Comerciantes e Portais de pagamento deverão dar suporte completo aos certificados do Portador de cartão e transações neles baseadas.

Certificados do Comerciante

Um comerciante terá pelo menos dois pares de chaves (cifragem e assinatura) para participar em transações do SET. Um comerciante pode ter dois conjuntos adicionais de pares de chaves de cifragem e assinatura por causa de implementação física, preocupações de segurança, política do Adquirente, ou uma variedade de outras razões. Por exemplo, um comerciante que opera servidores múltiplos deve eleger ter um conjunto separado de par de chaves de cifragem e assinatura para cada servidor. Em adição, novos pares de chaves serão gerados periodicamente.

O número de certificados necessários para um comerciante é função do número de pares de chaves de cifragem e assinatura do comerciante, o número de portais de pagamento que tem interface com o comerciante, e o número de marcas aceitas pelo comerciante. Há uma variedade de assuntos que influem na quantidade de portais de pagamento que tem interface com um comerciante. No caso mais simples, o comerciante terá interface com um único portal de pagamento para processar todas as marcas. Entretanto, um comerciante pode ter relacionamentos com múltiplos Adquirentes. Por exemplo, um único Adquirente pode não processar todas as marcas que o comerciante aceita, ou o comerciante pode negociar em múltiplos mercados nacionais (e moedas diferentes) e ter os correspondentes relacionamentos com Adquirentes. Em adição, os Adquirentes podem escolher operar portais múltiplos para balanceamento da carga de trabalho.

O SET permite ao Adquirente mandar informação de pagamento do Portador de cartão de volta ao comerciante, cifrada com a chave do comerciante. Esta capacidade é designada por um indicador no certificado do comerciante. Esta opção é destinada permitir que os comerciantes usem mecanismos de compensação utilizando-se de outros meios que não o SET.

Certificados do Portal de Pagamento

Dois pares de chave são necessários no Portal de pagamento:

- Um par de chaves de assinaturas que é usado para assinar e verificar mensagens providas pelo Portador de cartão e Comerciante; e
- Um par de chaves de cifragem que é usado para proteger instruções de pagamento geradas pelo Portador de cartão e pelo Comerciante.

Validação da Cadeia de Certificados

Os certificados serão validados através de uma hierarquia de confiança. Cada certificado é ligado ao certificado de assinatura da entidade emissora de certificado. Os certificados são validados seguindo a hierarquia de confiança até a CA Raiz. O caminho pelo qual os certificados são validados é chamado de “cadeia de certificados”, que tem a composição ilustrada na Figura 3.2.

A validação de cada certificado será obrigatória em todos os níveis da Cadeia de Certificados. Por exemplo, um Portador de cartão deverá validar o Comerciante, a CA do comerciante, a CA da Marca Geo-política, a CA de Marca, e os certificados da CA Raiz. O processo de validação deve parar em um nível que tenha sido previamente validado.

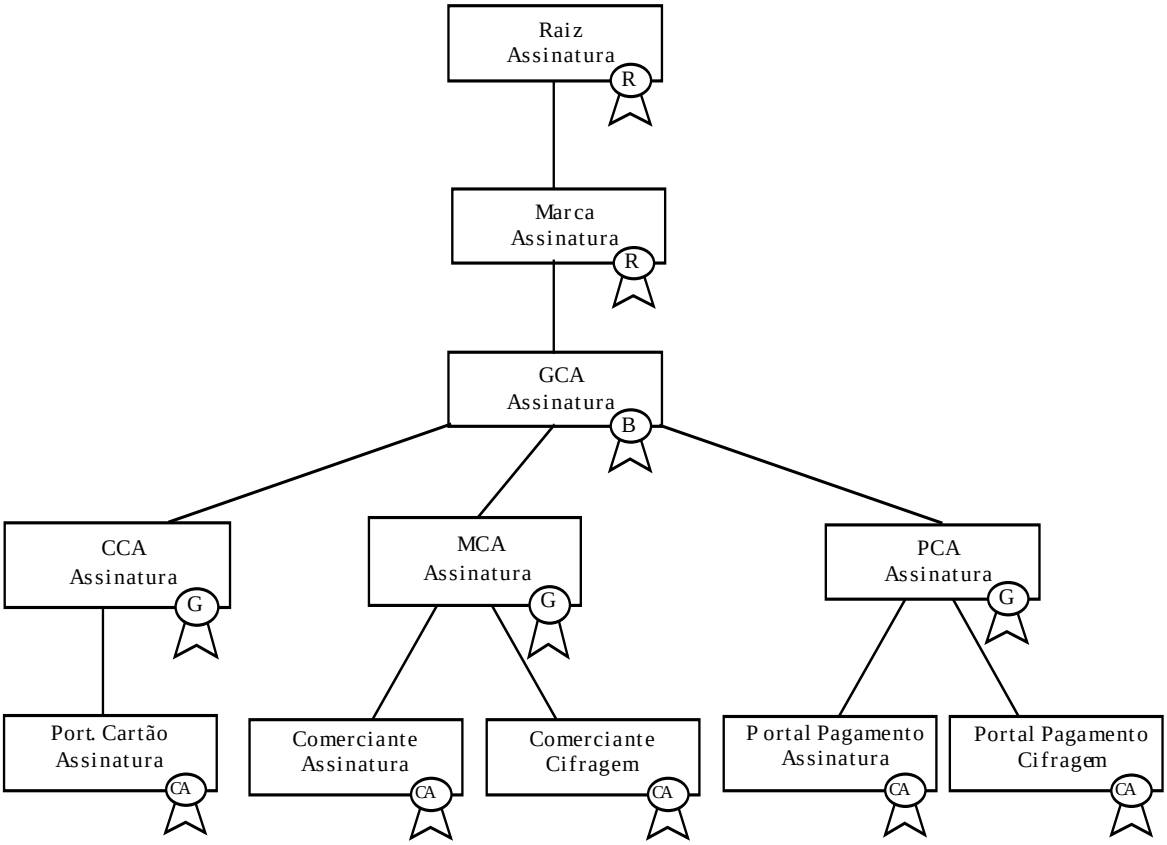

Figura 3.2: Cadeia de Certificados - Hierarquia de Confiança

Resumo de Tipos de Certificado

O quadro abaixo exibe a lista de todos os certificados necessários no SET:

Tipos de Certificados	Chave de Assinatura Digital	Chave de Cifragem	Certificado e CRL assinados
Portador de cartão	X		
Comerciante	X	X	
Portal de pagamento	X	X	
CA do Portador de cartão	X	X	X
CA do Comerciante	X	X	X
CA do Portal de pagamento	X	X	X
CA de Marca Geo-política	X		X
CA de Marca			X
CA Raiz			X

Identificador da CRL de Marca

Cada marca é responsável pelo gerenciamento das CRLs (Lista de Certificados Revogados) dentro de seu próprio domínio. A arquitetura do SET introduz o conceito de BrandCRLIdentifier (BCI). Um BCI é assinado digitalmente pela marca e usado para identificar as CRLs do SET que o Portador de cartão, Comerciante, Portal de Pagamento, e sistemas de CA precisam filtrar sempre que validando certificados como parte das verificações de assinatura.

Cada instância de um BCI identifica a marca e os tipos dos assuntos de CA da marca que têm CRLs que precisam ser processadas ao validar assinaturas nas mensagens do SET. Cada BCI tem um número de sequência e período de validade.

O quadro que se segue lista os tipos de entidades CA do SET que devem existir em uma BCI e a motivação para a inclusão de cada entidade.

Entidade	Razão do BCI
CA Raiz	Substituição não programada ou expiração de certificados da Raiz ou da CA de marca.
Qualquer CA de Marca	Substituição não programada ou expiração de um certificado emitido pela CA de marca.
CA de Marca Geo-política	Substituição não programada ou expiração de entidades CCA, MCA, ou PCA.
CA de Portal de Pagamento	Substituição não programada ou expiração de certificados de Portal de Pagamento.

3.4.3. Adaptabilidade do SET

A arquitetura do SET foi projetada para ser adaptável a diferentes modelos de negócios e ambientes operacionais, tal como suporte a Portadores de cartão sem certificados.

O SET usa a versão 3 do padrão X.509, para o esquema de certificados que dão suporte às chaves públicas de assinatura e cifragem. Estes certificados incluem uma chave pública junto com a autenticação daquela chave.

O certificado de assinatura do Portador de cartão provê autenticação e integridade da informação enviada ao Comerciante e ao Portal de Pagamento. O SET permite ambientes nos quais são requeridos certificados de assinatura de Portador de cartão e também ambientes onde certificados de Portador de cartão são opcionais. Uma marca de cartão de pagamento determina se sua aplicação do SET requer certificados de assinatura ou não.

Em ambientes onde são requeridos certificados, todas as mensagens que requerem autenticação e integridade do Portador de cartão serão assinadas com uma assinatura autenticada pelo certificado de Portador de cartão. Existem pedidos de iniciação de protocolo que não incluem tais assinaturas, considerando que nenhuma falha de protocolo significativa seria o resultado do mau uso destes pedidos. Todas as outras mensagens são assinadas, e os receptores destas mensagens terão recepção assegurada pelos certificados correspondentes do protocolo.

Quando um Portador de cartão não tem um certificado de assinatura, nenhuma assinatura digital é gerada. Em lugar da assinatura digital, o Portador de cartão gera o *hash* dos dados e insere o *hash* no envelope digital para assegurar a integridade de seus conteúdos.

3.4.4. Segurança Primária para o SET

A intenção de SET é focalizar os assuntos de segurança relacionados aos mecanismos de pagamento das três partes envolvidas, Portador de cartão, Comerciante e Portal de pagamentos, conduzidos sobre a Internet.

Os mecanismos de assinatura de chave pública são criticamente dependentes da segurança das chaves privadas correspondentes. O SET requer pares de chaves pública/privada para os Portais de pagamento e Comerciantes. Assim como também recomenda esta opção aos Portadores de cartão. Os desenvolvedores de aplicações para suporte ao SET deverão prestar particular atenção aos métodos usados para armazenagem das chaves privadas destes participantes. As chaves privadas deverão ser protegidas por cifragem, ou talvez usando outros mecanismos resistentes contra acesso indevido. Os Portais de pagamento irão usar módulos criptográficos de *hardware* resistentes para executar funções criptográficas e para geração e armazenamento de chaves privadas. Os servidores do comerciante e as aplicações

do Portador de cartão também deveriam empregar módulos criptográficos de *hardware* para executar funções criptográficas e gerar e armazenar chaves privadas.

O SET oferece uma opção que permite ao Portal de pagamento prover informações da conta do Portador de cartão ao Comerciante, cifrada sob a chave pública do Comerciante. Quando esta opção for usada, cuidado deverá ser tomado para assegurar a segurança da informação de pagamento e como esta reside nos sistemas do comerciante. O *software* do Comerciante irá armazenar a informação de pagamento de forma cifrada. Os Comerciantes também deveriam armazenar a informação de pagamento *off-line*,atrás de um *firewall* ou mecanismo semelhante.

Os certificados, CRLs e BCIs irão ser acessados frequentemente quando processando-se mensagens do SET. Assim, o processamento de sucessivas mensagens do SET deve ser otimizado pela manutenção de uma memória local segura contendo certificados, CRLs e BCIs, freqüentemente acessados. Os sistemas do Portador de cartão e do Comerciante que suportam o SET deverão manter uma política para proteger contra acesso sem autorização ou contra modificação de sua memória local contendo certificados, CRLs, BCIs e as impressões digitais a eles correspondentes.

Capítulo 4

Formato e Protocolos das Mensagens do Set

A implementação do SET reside nos fluxos de pares de mensagens de Pedido e Resposta entre as entidades para realizar trabalho útil. O objetivo deste capítulo é fornecer uma visão geral de como são constituídas as mensagens do SET, os protocolos de fluxos de comunicação, sua funcionalidade. A término do capítulo é apresentado o par de mensagens de pedido de autorização do Comerciante ao Portal de pagamento para venda de uma compra iniciada pelo Portador de cartão, como exemplo de composição de mensagens do SET.

4.1. Formato Geral da Mensagem no SET

As mensagens no SET são formatadas como mensagens técnicas não proprietárias, considerando que permitem a comunicação sobre uma variedade de mecanismos de tempo real ou não. Onde possível, são empregados padrões para permitir que o SET seja implementado facilmente e assegure a que interoperalidade entre aplicações seja possível. Tratamentos criptográficos são obrigatórios para garantir os níveis de proteção requerido pelas necessidades de segurança de transações de cartão de pagamento sobre redes abertas, como a Internet. Para promover a interoperabilidade e a capacidade de atualização, o SET usa os Padrões de Criptografia de Chave Pública (*Public Key Cryptography Standards* - PKCS) para representação dos parâmetros criptográficos e encapsulamento das mensagens.

As mensagens do SET são definidas usando os padrões ISO/IEC e a Notação de Sintaxe Abstrata do ITU-T (*Abstract Syntax Notation* - ASN.1) e serão codificadas usando as Regras de Codificação Distintas *(Distinguished Encoding Rules* - DER). Isto permite codificação não ambígua através de um padrão bem compreendido e extensamente aceito.

A especificação do SET não define como uma mensagem do SET é transportada entre entidades. Estas mensagens podem ser transportadas usando qualquer mecanismo que o emissor e o receptor concordem. É esperado que padrões de transporte sejam desenvolvidos para atender ao aspecto de interoperabilidade das aplicações do SET.

É esperado que as aplicações do SET operem em um de dois ambientes:

- Interativo - neste ambiente, as entidades se comunicam em “tempo real” com demoras de tempo pequenas entre a troca de mensagens (como a *World Wide Web*); e

- Não interativo - neste ambiente, as entidades se comunicam em “não em tempo real” com demoras de tempo grandes entre as trocas de mensagens (como *E-mail*).

Em um ambiente interativo, é esperado que um "Processo de Iniciação do SET" tome lugar e que ative o protocolo do SET. Este processo permitirá que o Portador de cartão de pagamento e o Comerciante troquem as informações requeridas pelo SET. Tais informações incluem, mas não se limita à marca do cartão que o Portador de cartão tenha selecionado, a descrição do pedido, e a quantidade da compra. É esperado que padrões sejam desenvolvidos para atender como esta informação é trocada e como o protocolo do SET é iniciado.

4.1.1. Mensagens Codificadas ASN.1/DER

De acordo com o *Federal Standard* 1037 C – um glossário de Termos de Telecomunicações publicado pela Administração Geral de Serviços dos EUA, a *Abstract Syntax Notation One* é definida como :

“Um padrão, método flexível que: (a) descreve as estruturas de dados para representação, codificação, transmissão, e decodificação de dados, (b) fornece um conjunto formal de regras para descrição das estruturas independentes de objetos das técnicas de codificação específica-de-máquina, (c) uma linguagem formal de gerência de rede *Transmission Control Protocol/ Internet Protocol* (TCP/IP) que usa notação legível pelo homem e uma compacta representação codificada da mesma informação usada em protocolos de comunicação, e (d) é uma precisa notação formal que remove ambiguidades.”

Embora se pense que a ASN.1 pareça ser obscura, seu uso pode ser encontrado em ocorrências comuns diárias. Cada vez que se faz uma chamada telefônica celular na América do Norte, Europa, ou Japão, sua chamada resulta em mensagens no Protocolo TCAP (*Transaction Capabilities Procedures*) que é definido usando ASN.1. Quando, nos Estados Unidos, se usa um número de chamada 1- 800, as mensagens ASN.1 são trocadas entre computadores de comutação e base de dados de redes para rotear essa chamada a um transportador comum e o número de telefone local que está mapeado para o número 800. Os protocolos de troca de mensagens terra-a-terra e terra-ar usados pela Agência Federal de Aviação dos EUA e Organização de Aviação Civil Internacional são descritos através da ASN.1 e codificados usando as Regras Básicas de Codificação da ASN.1. A Federal Express usa a ASN.1 para rastrear o movimento de pacotes de estação para estação, também. Onde se encontra alguma forma de protocolo de comunicação, com certeza se achará a ASN.1 também [4].

A ASN.1 é baseada no conceito de digitação de dados, o mesmo conceito usado em muitas linguagens de programação. Sua sintaxe abstrata libera o usuário de quaisquer obrigações ou restrições orientadas-à-máquina. A ASN.1 se refere a quatro classes de tipos-de-dados:

- Tipo-de-dados Universais - Aplicações de construções e dados independentes
- Larga Aplicação - Tipos de dados relevantes definidos por outros padrões.
- Contexto específico - Tipos de dados relevantes aplicados num contexto limitado.
- Privado - Tipos de dados definidos pelos usuários e cobertos por algum padrão.

A ASN.1 é usada para definir os tipos de dados e valores de dados. Uma definição de tipo assume a forma de:

$$\langle \textit{type name} \rangle ::= \langle \textit{type definition} \rangle$$

Um exemplo de codificação ASN.1 de um nome de pessoa deve ser implementada desta maneira:

> Informal name: Jose R. Silva
>
> Definição de tipo ASN.1 (IA5String é equivalente a uma sequência de caracteres ASCII):

```
Name:: = [APPLICATION 1] IMPLICIT SEQUENCE {
firstName IA5String ,
midInitial IA5String ,
familyName IA5String }
```

Uma definição formal do nome sob esta estrutura é:

```
{ firstName  “Jose”, midInitial “R”, familyName “Silva”}
```

Na ASN.1, um tipo é um domínio de valores, alguns com um número finito de possíveis valores e alguns com um infinito número de possíveis valores. A ASN.1 usa quatro qualidades de tipos, tipos simples (aqueles que não possuem componentes), tipos estruturados (aqueles que consistem de componentes), tipos etiquetados (que são componentes de outros tipos), e os tipos CHOICE e ANY para finalidades de seleção. Um exemplo de código ASN. 1 para uma mensagem simples do SET, como por exemplo AuthReq é:

```
665     AuthReq::= EncB  {M, P, AuthReqData, PI}
```

(como no exemplo anterior)

```
1061    AuthReqData::= SEQUENCE {
1062    authTags              AuthTags,
1063    checkDigests          [0] CheckDigests OPTIONAL,
1064    mThumbs               [1] Thumbs OPTIONAL,
1065    authReqPayload        AuthReqPayload
1066    }
```

```
628     PI::= CHOICE {
629     piUnsigned            [0] PIUnsigned,
630     piDualSigned          [1] PIDualSigned
631     authToken             [2] AuthToken
632     }
```

Este exemplo do padrão ASN.1 nos informa que a mensagem **AuthReq** é uma mensagem enviada pela entidade **M** (*Merchant*) à entidade **P** (*Payement Gateway*), onde **AuthReqData** é uma estrutura tipo sequência, constituída das estruturas de dados **authTags**, tendo em

seguida uma estrutura de dados **checkDigests**, que é opcional, sendo então seguida da estrutura de dados **mThumbs**, também opcional, ou seja, pode ou não estar presente.

Os números entre colchetes servem como etiquetas para escolhas ou para determinar a sequência dentro de um conjunto de valores. A última estrutura de dados componente da sequência **AuthResData** é a estrutura de dados **authReqPaylo**ad. Os números de linhas de código-fonte mostrados indicam o número da fileira do código fonte definido pelo protocolo, e são similares aos números de linhas usados por outras linguagens de programação.

A estrutura de dados *PI* , componente da sequência **AuthReqData**, já havia sido definida anteriormente (vide linha de código fonte). O padrão ASN.1 exposto nos diz que esta estrutura de dados é do tipo escolha, e que uma das três estruturas de dados definidas **piUnsigned**, **piDualSigned** e **authToken** estará presente e é a estrutura de dados *PI* . Os valores alinhados entre os colchetes são correspondentes às estruturas de dados com os identificadores após os colchetes.

Como em outras linguagens de programação, os compiladores ASN.1 compilam o código ASN.1, porém sua saída é um código fonte da linguagem específica que o compilador de linguagem de programação irá re-compilar num código executável. Tipicamente, as definições de tipo de dados são compiladas em estruturas *C* ou *C* ++ e classes de Java.

Geradores comerciais de código ASN.1 estão disponíveis e permitirão aos desenvolvedores de *software* gerar e receber estas mensagens com um esforço de programação modesto, além de prover a própria especificação da ASN.1 ao compilador.

Regras de Codificação Distintas (*Distinguished Encoding Rules* – DER)

As regras de codificação da ASN.1 são usadas na transformação de dados especificados pela linguagem num formato padrão que pode ser decodificado por qualquer outro sistema baseado na mesmas regras do decodificador. As regras de decodificação representam os objetos abstratos ASN.1 como sequencias de 0s e 1s. As Regras Básicas de Codificação (*Basic Encoding Rules* – BER) oferecem uma ou mais maneiras de representar qualquer valor ASN.1

como uma sequencia de octetos e é o *default* para a codificação sob o padrão OSI – *Open Systems Interconnect*.

As Regras de Codificação Distintas (DER) , por outro lado, provêem exatamente uma maneira para representar os valores ASN.1 como uma sequência de octetos. Octeto é a definição de cadeia de 8 bits justificados nas posições de menor valor, com enchimento, se necessário, das posições de maior valor por bits 0. O padrão DER é projetado para o uso em aplicações onde valores únicos e não ambíguos são requeridos, tais como os requeridos em aplicações com foco na segurança como o SET. A ASN.1 provê uma definição clara não ambígua, do conteúdo de mensagens; O DER provê uma codificação que é precisa e assegura um único formato dos dados codificados, o que é crítico para poder apoiar as operações que envolvem *hashes* e assinaturas.

Como exemplo apresentamos a codificação DER num segmento de mensagem do SET. As mensagem do SET são constituídas por campos, que por sua vez são constituídos de estrutura de dados. Os pontos (.) que precedem os nomes de campos ou estrutura de dados sinalizam um aninhamento de outros campos ou estruturas de dados em um nível interno ao atual. Na tabela do exemplo abaixo, a estrutura de dados apresentada é **AuthReqData**, dados do pedido de autorização, que faz parte da mensagem **AuthReq**, utilizada quando o comerciante faz o pedido de autorização de uma compra ao Adquirente.

Campos / Estrutura de Dados	**Conteúdo**	**Codificação DER**
AuthReqData		30 81 FF
.authReqItem		30 81 FC
..authTags		30 7B
...authRRTags		30 35
....rrpid		04 14 D1 65 CE 64 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70
....merTermIDs		30 0C
.....merchantID	MerchantID	13 0A 4D 65 72 63 68 61 6E 74 49 44
....currentDate	19970509175416Z	18 0F 31 39 39 37 30 35 30 39 31 37 35 34 31 36 5A
...transIDs		30 42

....localID-C		04 14 6C 69 64 63 2D 6C 69 64 63 2D 6C 69 64 63 2D 6C 69 64 63 2D
....Xid		04 14 78 69 64 2D 78 69 64 2D 78 69 64 2D 78 69 64 2D
....pReqDate	19970509175416Z	18 0F 31 39 39 37 30 35 30 39 31 37 35 34 31 36 5A
....language	en	1A 03 65 6E 20
. . .	. . .	. . .

Impressões Digitais

Para apoiar as exigências de segurança do SET, certificados de chaves públicas e CRLs deveriam ser transportadas no protocolo. Considerando que estas estruturas de dados são grandes, um mecanismo de impressão digital é provido para reduzir o tráfego exigido associado aos certificados, CRLs e BCIs. Uma impressão digital é um resumo (*hash*) da porção de dados de um certificado, da CRL, ou BCI. Mais especificamente, é o *hash* padrão SHA-1 dos dados que são assinados por uma das entidades certificadoras precedentes[2]. Se uma entidade do SET precisasse de um certificado ou CRL de outra entidade com que está se comunicando, ela deveria enviar à entidade remota o conjunto de impressões digitais que correspondem aos certificados, CRLs, e BCIs que possui. A aplicação do emissor enviará só as impressões digitais que ele espera serem relacionadas à transação. Por exemplo, o *software* do Comerciante não enviará as impressões digitais de outros Portadores de cartão ou de outras marcas. A entidade que responde deveria omitir de sua mensagem de resposta qualquer certificado e CRLs para os quais recebeu impressões digitais. Considerando que as impressões digitais são muito pequenas comparadas aos certificados e CRLs que eles representam, muito tráfego é evitado.

4.1.2. Envoltória de Mensagem (*MessageWrapper*)

A envoltória de mensagem é o nível de topo da estrutura de dados ASN.1 / DER no protocolo SET. Provê outras informações apresentadas ao receptor de uma mensagem bem no começo

do processamento da mensagem, sem envolver qualquer criptografia. Identifica o tipo de mensagem e tem seus identificadores únicos, dados suficientes para basear as decisões iniciais, como por exemplo a detecção duplicada, ou outras. A Tabela 4.1 apresenta as descrições dos campos contidos na Envoltória de Mensagem. A notação usada nesta a tabela se encontra detalhada no próximo capítulo.

Todo o processo relacionado ao SET começa com a envoltória de mensagem. Cada mensagem do SET contém uma envoltória de mensagem em texto claro que será decodificada antes do processamento da mensagem. Na envoltória foram colocados os campos **TransIDs** e **RRPID** para permitir logo a detecção de duplicidade; estes campos da envoltória estão repetidos dentro do corpo da mensagem envolvida, de forma que estes dados tenham sua integridade protegida.

O campo *Message* contém a identificação do tipo de mensagens que estão efetivamente sendo transmitidas. A notação deste campo, $\langle Pinit\,\mathrm{Re}q,\ldots,Error\rangle$, representa que somente uma opção de mensagem entre todas as descritas pode ser enviada numa envoltória. No tempo em que a envoltória esta sendo decodificada, o componente mensagem não pode ser processado, mas seu tipo pode ser determinado da informação codificada DER contida nesse campo. Depois que o processamento da envoltória é executado, a mensagem é decifrada e/ou sua assinatura é verificada como é apropriado, então o conteúdo é decodificado para fornecer a informação que é processada individualmente para cada tipo de mensagem.

Para que o SET tenha êxito, suas novas versões devem ser capazes de interoperar com versões anteriores. Em geral, as aplicações de *software* devem interoperar com a revisão atual de SET e a versão anterior. Quer dizer, uma aplicação que opere com a versão 2 do SET, quando publicada, poderá enviar e receber mensagens da versão 1. Para conferir a versão da mensagem, a aplicação conferirá primeiro os campos **MessageHeader.Version** e **MessageHeader.Revision**.

A extensão da versão 1 do SET foi limitada intencionalmente à funcionalidade mínima necessária para apoiar os Portadores de cartão e Comerciantes que negociam na Internet. Consequentemente, algumas funções empresariais não foram incluídas na definição de mensagens de pagamento do SET. Além disso, é improvável que o SET sempre pudesse ser robusto bastante para cobrir as práticas empresariais de todos os mercados nacionais e de

todos Adquirentes. Então, é necessário prover um mecanismo para estender as mensagens de pagamento do SET .

Um exemplo de uma função empresarial que não é apoiada pelas mensagens do SET são as opções de pagamento japonesas. Os Emissores no Japão têm opções de pagamento que são selecionadas pelo consumidor na hora da compra. Desde que não há nenhum lugar na mensagem do SET para levar esta informação, uma extensão do protocolo é necessária.

Nome do Campo	Descrição
Message-Wrapper	**{MessageHeader, Message, [MWExtensions|}**
MessageHeader	**{Version, Revision, Date, [MessageIDs], [RRPID], SWIdent}**
Version	Versão da mensagem do SET.
Revision	Revisão da mensagem do SET.
Date	Data/Hora da geração da mensagem.
MessageIDs	**{[LID-C], [LID-M], [XID]}** (vide abaixo).
RRPID	ID do par Pedido/resposta para este ciclo.
SWIdent	Identificação do *software* (vendedor e versão) iniciando o pedido. Isto é uma sequência de dados.
Message	**< PinitReq, PinitRes, PReq, PRes, InqReq, InqRes, AuthReq, AuthRes, AuthRevReq, AuthRevRes, CapReq, CapRes, CapRevReq, CapRevRes, CredReq, CredRes, CredRevReq, CredRevRes, PcertReq, PcertRes, BatchAdminReq, BatchAdminRes, CardCInitReq, CardCInitRes, Me-AqCInitReq, Me-AqCInitRes, RegFormReq, RegFormRes, CertReq, CertRes, CertInqReq, CertInqRes, Error>**
LID-C	ID do Local; rótulo de conveniência gerado pelo e para o sistema do Portador de Cartão.
LID-M	ID do Local; rótulo de conveniência gerado pelo e para o sistema do Comerciante.
XID	ID único globalmente gerado pelo Comerciante (ou Portador de Cartão, se não existe **PInitRes**).
MWExtensions	Extensões de Mensagem do SET. Uma extensão poderia ser apropriada na envoltória de mensagem sob uma das duas condições: • Os dados na extensão são informação de finalidade geral sobre as mensagens do SET; ou • Os conteúdos da mensagem são codificados e a extensão contém dados não financeiros que não requerem confidencialidade. Nota: A envoltória de mensagem não é codificada assim a extensão não conterá informação confidencial.

Tabela 4.1 – Descrição dos Campos da Envoltória de Mensagens do SET

Os mecanismos usados para estender as mensagens do SET seguem a forma de como certificados X.509 são estendidos [26]. Especificamente, um campo de extensões é provido para conter uma sucessão de dados de extensão.

4.2. Protocolos de mensagens do SET

Aqui são abordadas as mensagens que varrem todas as fases de uma transação de cartão de pagamento do SET para processamento de uma compra eletrônica, conforme descritas na Tabela 3.1. Processamento da Compra Eletrônica, vista no capítulo 3 deste trabalho. As mensagens pertinentes a esta discussão incluem:

- Mensagens de Investigação
- Mensagens de Processamento de Pagamento
- Mensagens de Captura Reversa
- Mensagens de Fornecimento de Crédito
- Mensagens de Requisição de Certificados do Portal de Pagamento
- Mensagens de Administração de Lote
- Mensagens de Fornecimento de Certificado
- Mensagens de Investigação de Certificado
- Mensagens de Erro

Muitas destas mensagens são usadas mais de uma vez através dos protocolos, e algumas devem ser usadas mais de uma vez dentro de um protocolo em particular. Alguns pares de mensagens são opcionais – usados para investigação de status e processamento tipo exceção, como reversões e estornos de crédito de bens.

4.2.1. Protocolos de Fornecimento de Certificados

Os três processos discutidos nesta seção definem o protocolo do SET para o pedido e fornecimento de certificados para Portadores de cartão, Comerciantes, e Portais de pagamento. Variações nesses fluxos são possíveis, baseadas nas implementações do Emissor

ou Adquirente do SET. Quaisquer variações irão ser descritas no acordo com o Adquirente do SET.

Protocolo de Pedido de Certificado de Portador de cartão

O processamento de renovação de certificados deve seguir um fluxo similar, porém pode variar de Emissor para Emissor.

Este processo corresponde ao estabelecimento da preparação do Portador de cartão para a Fase 0, ou seja a fase anterior daquela primeira estabelecida na Tabela 3.1, ou seja, o processamento do par de mensagens de aquisição de Certificado Digital de Portador de cartão. O fluxo é o seguinte.

De acordo com uma mensagem de invocação (não definida pelo SET) da Autoridade de Certificado do Portador de cartão (CCA), a carteira eletrônica, *software* da estação de trabalho do Portador de cartão que opera com o SET, prepara um Pedido de Iniciação de Certificado de Portador de cartão (***Cardholder Certificate Initiation Request*** **– CardCinitReq**) contendo uma lista de certificados, CRLs e Identificadores das CRLs que a carteira atualmente guarda , na forma de impressões digitais daqueles itens. O pedido é então retornado para a CCA. A CCA responde com uma mensagem de Resposta de Iniciação de Certificado de Portador de cartão (***Cardholder Certificate Initiation Response*** **– CardCInitRes**) contendo quaisquer certificados, CRLs, e os Identificadores de CRLs que o portador de cartão irá necessitar para verificar as assinaturas e a informação cifrada de certificado mais tarde no fluxo.

Uma recepção bem sucedida do **CardCInitRes** pela carteira eletrônica cria uma mensagem de Pedido de Formulário de Registro (***Registration Form Request*** **– RegFormReq**) que é enviada para a CCA. A CCA responde com mensagem de Resposta de Formulário de Registro (***Registration Form Response*** **– RegFormRes**) que tem um modelo de formulário de informação para o Portador de cartão completar, em acordância às regras do Emissor do cartão para autenticação do Portador de cartão. O Portador do cartão completa o Formulário e a carteira cifra-o (junto com outras informações) dentro de uma mensagem de Pedido de Certificado (***Certificate Request*** **– CertReq**) indicando a prontidão do portador de cartão para receber seu certificado digital. Se validado com sucesso, usando as diretrizes do Emissor para

validação, a CCA prepara uma mensagem Resposta de Certificado (***Certificate Response* – CertRes**), contendo o Certificado Digital do Portador de Cartão preparado, pronto para armazenagem pela carteira eletrônica.

Nos casos onde o certificado digital não é retornado na **CertRes** , a carteira do Portador de cartão deve requisitar o status do pedido através do uso da mensagem de Pedido de Investigação de Certificado (***Certificate Inquiry Request* – CertInqReq**). Com a mensagem de resposta da CCA, a Resposta de Investigação de Certificado (***Certificate Inquiry Response* - CertInqRes**) irá conter o certificado preparado ou fornecer a informação que diz ao Portador de cartão quando ele estará pronto. Este par de mensagens é opcional no SET.

Uma vez que essa transação é bem sucedida, o Portador de cartão irá ter a posse do certificado digital requerido para cada crédito ou débito que ele estiver apto para usar *on-line*, e está pronto para a Fase 0 de uma transação do SET. Este protocolo irá ser usado para cada cartão que um consumidor decide registrar.

Protocolo de Pedido de Certificado de Comerciante

A exemplo do Portador de cartão, o processo corresponde ao estabelecimento da preparação do Comerciante para a Fase 0 – processamento do par de mensagens de aquisição de Certificado digital de Comerciante. O fluxo é o seguinte.

A Iniciação de Certificado de Comerciante começa o processo de obtenção de Certificados Digitais de Comerciante do SET. O componente **POS** (***Point Of Sale***), *software* de suporte ao SET do Servidor do Comerciante, irá começar o processo (sob a ativação do administrador do sistema) pela preparação de uma messagem de Pedido de Iniciação de Certificado de Comerciante-Adquirente (***Merchant-Acquirer Certificate Initiation Request* – Me-AqCInitReq**) contendo uma lista de certificados, CRLs, e Identificadores de CRL que o sistema POS atualmente possui. Ele também contém informações de banco, os tipos de certificados sendo requeridos e um Formulário de registro para o banco indicado ou companhia de cartão. Se o Comerciante solicitar o Formulário de registro, a mensagem de resposta irá contê-lo. O pedido é então retornado para a MCA. A MCA responde com uma

mensagem de Resposta de Iniciação de Certificado de Comerciante-Adquirente (***Merchant-Acquirer Certificate Initiation Response* – Me-AqCInitRes**). A Mensagem contem:

- Um Formulário de registro (modelo) para o Comerciante completar (se necessário).
- Quaisquer certificados, CRLs, e os identificadores de CRL que o Comerciante irá usar para verificar assinaturas e certificados cifrados mais tarde.
- Uma declaração da política para o Comerciante ler e concordar.
- URLs de marcas e logos de cartões.

Se o pedido não é bem sucedido, a resposta irá retornar a razão e/ou a URL ou endereço do *e-mail* que conduzirá o comerciante para informação adicional.

Uma recepção bem sucedida do **Me-AqCInitRes** pelo *software* POS do SET do Comerciante cria uma mensagem de Pedido de Certificado (***Certificate Request* – CertReq**) que é enviada para a MCA. Se validada com sucesso usando as diretrizes do Adquirente para validação, a MCA prepara um mensagem Resposta de Certificado (***Certificate Response* – CertRes**), contendo o Certificado Digital do Comerciante preparado, pronto para armazenamento pelo *software* POS.

Nos casos onde o certificado digital não é retornado na **CertRes**, o *software* POS do Comerciante deve solicitar o status do pedido pelo uso da mensagem de Pedido de Investigação de Certificado (**Certificate Inquiry Request – CertInqReq**). A mensagem resposta da MCA, Resposta de Investigação de Certificado (***Certificate Inquiry Response* - CertInqRes**), irá conter o certificado preparado ou fornecer informação que diz ao Comerciante quando ele estará pronto. Este par de mensagem é opcional sob o SET.

Uma vez que a transação é bem sucedida, o Comerciante irá estar de posse dos certificados requeridos que representam a marca da loja para todos os cartões de pagamento que ela aceita, e está pronto para a Fase 0 de uma transação do SET. Este protocolo será repetido para cada Marca que o Comerciante aceita para pagamento.

Protocolo de pedido de certificado de portal de pagamento

Este processo também corresponde ao estabelecimento da preparação do Portal de pagamento do Adquirente para a Fase 0 da transação – processamento do par de mensagens de fornecimento de Certificado de Adquirente. Este protocolo segue o mesmo fluxo do protocolo de Pedido de Certificado de Comerciante, sendo as diferenças somente quanto à origem ou destino das mensagens. Este fluxo de protocolo é entre o Portal de Pagamento e a Autoridade de Certificação de Portal de Pagamento (PGA), com todos os pares de mensagem operando como descrito para a transação equivalente do Comerciante – MCA.

Uma vez que a transação é bem sucedida, o sistema de Portal de pagamento estará de posse dos certificados requeridos que representam sua legitimidade para os Portadores de cartão e Comerciantes, no processamento de débito para uma dada marca. Isto prepara o sistema do Portal de pagamento para a Fase 0 de uma transação do SET. Este protocolo será repetido para cada marca de cartão que ele suporta para os serviços de Portal de pagamento do(s) Banco(s) Adquirente(s).

4.2.2. Protocolos do Sistema de Pagamentos

Os próximos três modelos de processos ilustram as mensagens que tomam lugar no processamento de cartões de pagamento entre Portadores de cartão, Comerciantes, e Portais de pagamento.

Protocolo de Compra Normal

Este protocolo corresponde às Fases 4, 5, 6 e 8 de uma transação de cartão de pagamento *on-line*, descrita na Tabela 3.1- Processamento da Compra Eletrônica. Cabe lembrar que não fazem parte do escopo do SET as fases 1, 2, 3 e 7, da transação de pagamento.

O fluxo para o processamento de pagamento normal é o seguinte:

Dependendo da conclusão da Experiência de Compra (Fase 1 da Tabela 3.1), o Processo de Seleção de Item (Fase 2 da Tabela 3.1) e o Processo de *Check-out* (Fase 3 da Tabela 3.1), o Portador de cartão selecionará uma forma de pagamento. Se ele seleciona um cartão de pagamento dotado do SET (Fase 4 da Tabela 4.1), o SET é iniciado via a mensagem de Pedido de Inicialização de Compra (***Purchase Inicialization Request* – PInitReq**), preparada pela carteira eletrônica do Portador de cartão (sinalizando o início da Fase 5 da Tabela 3.1). Esta mensagem fornece ao Comerciante o identificador da marca do cartão de pagamento, um ID local criado pelo Portador de cartão para a transação, um *nonce* para verificação de quão atual é a resposta, e as impressões digitais dos certificados, CRLs e Identificadores de CRL para a marca que a carteira eletrônica tem em sua memória. A mensagem de Resposta de Pedido de Compra (***Purchase Initialization Response* – PInitRes**) do Comerciante contém os dados do pedido original, os certificados, CRLs, os Identificadores da CRLs que o Portador de cartão irá necessitar para processamento adiante, uma data, um ID de Transação (***Transaction ID* – XID**) criado pelo *software* POS do Comerciante, uma resposta ao desafio (*nonce*) enviado no pedido do Portador de cartão, e um desafio (*nonce*) recente do próprio Comerciante.

O SET permite a omissão destas mensagens quando usado em ambientes não interativos (tais como catálogo de compra em CD-ROM). Os certificados, CRLs e Identificadoes de CRLs providos, são obtidos por qualquer mecanismo *off-line* (por exemplo, cópias armazenadas em CD). Para nossas finalidades, vamos assumir que a transação é inicialmente conduzida via a Internet num ambiente de processamento *on-line*.

Com uma recepção bem sucedida do **PInitRes** pela carteira eletrônica do Portador de cartão, a mensagem de Pedido de Compra (***Purchase Order Request* – PReq**) é preparada. A mensagem **PReq** consiste de duas partes: Instruções do Pedido (***Order Instructions* – OI**) destinadas ao Comerciante e Instruções de Pagamento (***Payment Instructions* – PI**), destinadas ao Portal de Pagamento.

Quando o software POS do Servidor do Comerciante recebe a **PReq** ele possui todos os dados necessários para iniciar um Pedido de Autorização (***Authorization Request* – AuthRes**) ao Portal de Pagamento. A porção de ***Payment Instructions*** (**PI**) do **PReq** é copiada no **AuthReq** , mas ele é indecifrável pelo Sistema POS do Comerciante. Assim, o sistema somente o canaliza do Portador de cartão para o Portal de Pagamento. O **AuthReq** contém

dados assinados e cifrados a respeito da compra, junto com o **PI**. Isto inicializa a Fase 6 da Tabela 3.1 de uma transação SET *on-line*.

Após o processamento da autorização através da rede financeira estar completa, o Portal de Pagamento prepara uma mensagem de Resposta de Autorização (***Authorization Response* - AuthRes**) contendo o resultado.

O resultado deve indicar uma das três possíveis situações: aprovado, recusado, ou recusado condicionalmente. Uma resposta condicionalmente recusada corresponde a um indicador **callIssuer** no campo **AuthCode**. Quando um Comerciante recebe apenas esta resposta, usando um processamento fora do sistema SET, ele deve contactar o Banco Adquirente para tratar do assunto com o Emissor. Se o Emissor aprova o pedido, ele deve prover ao Comerciante um **AprovalCode** (código da aprovação) via telefone. O *software* POS do SET permite a entrada deste código antes da criação da mensagem de Resposta de Compra (***Purchase Response* – PRes**) que é retornada ao Portador de cartão. Deve existir algum atraso entre a mensagem **PReq** do Portador de cartão e a subsequente mensagem de resposta **PRes** do comerciante, dependendo do resultado do **AuthReq**. Uma conclusão bem sucedida destes passos finaliza a Fase 6 da Tabela 3.1.

Com uma conclusão bem sucedida do **PReq** através da **PRes**, o Comerciante, para todas as intenções e finalidades, possui um legítimo pedido do consumidor que ele irá necessitar atender. A Fase 7 da Tabela 3.1 cobre o processo de entrega de bens. Uma vez que a remessa ou execução dos serviços esteja completa, o Comerciante pode solicitar a captura da venda ao Portal de Pagamento do Adquirente (Fase 8 da Tabela 3.1).

Usando a informação do **AuthRes**, o sistema POS do Comerciante prepara a mensagem de Pedido de Captura (***Capture Request* – CapReq**) que deve conter uma ou diversas transações previamente autorizadas. Pedidos de captura incluem as informações do Comerciante que o Portal de pagamento necessita para produzir mensagens claras de pedidos a banco, para que o Banco Adquirente processe ou envie para a rede financeira para processamento (não definida pelo SET). A mensagem Resposta de Captura (***Capture Response* – CapRes**) do Portal de Pagamento indicará os resultados para cada transação representada dentro do **CapReq**, então concluindo a Fase 8 da transação de cartão de pagamento do SET *on-line*.

Quando completo, este protocolo corresponde a assinatura do Portador de cartão de um registro de autorização de débito produzido por um terminal POS num ambiente não Internet. No evento de recusa de uma transação, a venda é cancelada e o Portador de cartão deve apresentar outro cartão se desejar, reiniciando-se o processo.

Outros Protocolos de Mensagens do Sistema de Compra

Eventualmente outros processamentos devem ser solicitados para transações de compras de formas diferentes das anteriormente descritas. As condições que requerem fornecimento de crédito, reversões de autorizações bem sucedidas ou capturas, e mensagens de investigação, são a seguir descritas.

O par de mensagens de Pedido de Autorização de Reversão (***Authorization Reversal Request* – AuthRevReq**) e a Resposta de Autorização de Reversão (**Authorization Reversal Response – AuthRevRes**) deve ser usado entre o Comerciante e o Portal de Pagamento para adequar um pedido por causa da impossibilidade de remetê-lo totalmente em uma única vez. Isto deve dividir um pedido únic previamente autorizado em remessas parciais, ou modificá-lo através da remoção de itens do pedido. Esta mensagem não é diretamente executada dentro de qualquer fase formal de uma transação do SET, ao invés é usada para corrigir informações da Fase 6 da Tabela 3.1.

O Pedido de Captura (***Capture Request* – CapReq**) e a Resposta de Captura (***Capture Response* - CapRes**) são um par de mensagens adicional entre Comerciantes e Portais de pagamento porque o SET permite um processamento fora do sistema SET para realizar o mesmo trabalho. Isto é considerado um fluxo de par de mensagens de processo de pagamento normal como descrito anteriormente, porém não é absolutamente essencial para o próprio SET. Entretanto, deve ser desejável pelo Comerciante querer o seu *software* POS de Comerciante num ambiente onde seja permitido colher autorizações de processamentos de captura e liquidação automáticos, através das próprias mensagens do SET.
O par de mensagens Pedido de Investigação (***Inquiry Request* – InqReq**) e Resposta de Investigação (***Inquiry Response* – InqRes**) entre o Portador de cartão e o Comerciante é opcionalmente usado para investigar acerca do status de um pedido. Isto é uma das mensagens

indempotentes (ver seção 5.5) do SET. Uma ou mais múltiplas cópias da mensagem **InqReq** pode ser enviada a qualquer tempo. O **InqRes** é similar a mensagem **PRes**, e somente deveria ser usada na ausência de uma mensagem **PRes**, que indica ao Comerciante a disposição final de uma transação.

Mensagens de Reversão de Captura e Crédito são idênticas na sintaxe e executam as mesmas funções. As estruturas de dados definidas pelo SET são idênticas para ambos os conjuntos de mensagens.

A mensagem de Pedido de Reversão de Captura (***Capture Reversal Request* – CapRevReq**) é enviada pelo Comerciante ao Portal de pagamento para trocar ou eliminar uma transação de captura previamente realizada com sucesso. A mensagem deve ser enviada em qualquer tempo depois da conclusão de um pedido de captura para reduzir ou remover a quantidade de capturas. A mensagem Resposta de Captura Reversa (***Capture Reversal Response* – CapRevRes**) indica a exclusão de um pedido de processamento. Esta mensagem não é diretamente executada dentro de qualquer fase formal da transação do SET, ela é usada para corrigir informação da Fase 8.

As mensagens Pedido de Crédito (***Credit Request* – CredReq**) são enviadas do Comerciante para o Portal de pagamento solicitando um retorno de crédito de uma transação previamente capturada. Este par é usado quando a informação **CapReq/CapRes** para a transação original tenha envelhecido e não está mais disponível nos *logs* de transação do Comerciante ou Portal de Pagamento. Entretanto, se aquela informação ainda está disponível, o par de mensagem **CapRevReq/CapRevRes** deve ser usada em seu lugar.

O processamento do Pedido de Reversão de Crédito (***Credit Reversal Request* – CredRevReq**) e da Resposta de Reversão de Crédito (***Credit Reversal Response* – CredRevRes**) é usado entre o Comerciante e Portal de Pagamento para reverter uma transação de crédito concedido previamente. Isto deve ocorrer quando um Portador de cartão opta por retornar os bens ao Comerciante, porém os bens de fato não retornam, ou quando uma carga sob disputa é determinada como legítima e os bens permanecem na posse do Portador de cartão.

4.2.3. Protocolos de Pedido de Certificado de Portal e Administração de Lotes

O Pedido de Certificado de Portal de Pagamento (***Payment Gateway Certificate Request* – PcertReq**) é enviado pelo Comerciante para pedir as cadeias mais atuais de certificados de seu Portal de pagamento adquirente para as marcas de cartão que ele aceita. A Resposta de Certificado de Portal de Pagamento (***Payment Gateway Certificate Response* – PCertRes**) retorna o status do certificado solicitado e impressões digitais que correspondem à Marca e as sequências **BIN** (***Bank Information Number***) solicitadas. Os Certificados de Portal de pagamentos são requeridos pelo Comerciante para cifrar todas as comunicações pretendidas para futuro processamento. Este par de mensagens irá normalmente ser usado no início de cada dia de negócios, mas poderia também ser outras vezes durante o dia.

O processamento do Pedido de Administração de Lote (***Batch Administration Request* – BatchAdminReq**) e da Resposta de Administração de Lote (***Batch Administration Response* – BatchAdminRes**) é usado para gerenciar lotes de transações aguardando captura e liquidação de processamento. A Administração de Lote inclui pedidos para abrir um lote, fechar um lote (liquidando-o), eliminar um lote e retornar os totais, status, e/ou os detalhes de um lote para reconciliação e finalidades de balanceamento de lote. Estes pedidos e respostas ajudam ambos Comerciantes e Portais de Pagamentos a identificar quaisquer discrepâncias entre seus subconjuntos de dados de transação.

4.3. Protocolo de Mensagens de Erro

Algumas vezes as mensagens de pedido encontram mensagens de resposta de Erro devido a diversos problemas relacionados ao SET, em condições que as Especificações do SET identificam como erros que ocorrem quando respondedores de mensagens não podem confiantemente identificar uma mensagem de pedido que chega. As mensagens de Erro são em resposta às mensagens de entrada definidas pelo SET, e nunca em resposta a outras mensagens de Erro. Além disso, elas tipicamente são advindas de erros no nível de comunicação – não no nível de negócios, tais como recusas de autorizações ou erros identificados do sistema de *hosts* das entidades.

As categorias de mensagens de Erro incluem:

- Mensagens analisáveis, mas mal formadas que não seguem os requisitos do formato de mensagens do protocolo SET.
- Valores ilegais de componentes de mensagens.
- Falhas na criptografia.

O componente ***ErrorCode*** de uma mensagem de erro pode conter qualquer dos valores mostrados na Tabela 4.2 abaixo

Error Code	**Descrição**
UnspecifiedFailure	A razão para a falha não aparece em outro lugar nesta lista.
MessageNotSupported	Este tipo de mensagem válida não é aceita pelo receptor.
DecodingFailure	Um erro foi encontrado durante o processo de decodificação DER da mensagem.
InvalidCertificate	Um certificado necessário para processar esta mensagem não estava válido (por uma razão não especificada em outro lugar nesta tabela). O campo ***ErrorThumb*** identifica o certificado inválido.
ExpiredCertificate	Um certificado necessário para processar esta mensagem expirou. O campo ***ErrorThumb*** identifica o certificado inválido.
RevokedCertificate	Um certificado necessário para processar esta mensagem foi revogado. O campo ***ErrorThumb*** identifica o certificado inválido.
MissingCertificate	Um certificado necessário para processar esta mensagem não estava disponível na memória de certificados do receptor e não foi incluído na mensagem.
SignatureFailure	A assinatura digital da mensagem não pôde ser verificada.
BadMessageHeader	O cabeçalho de mensagem não pôde ser processado.
WrapperMsgMismatch	Os conteúdos da envoltória de mensagem são incompatíveis com o conteúdo interno da mensagem, por exemplo, o **RRPID** não corresponde.
VersionTooOld	O número de versão da mensagem é muito velho para o receptor processar.
VersionTooNew	O número de versão da mensagem é muito novo para o receptor processar.
UnrecognizedExtension	A mensagem ou um certificado contém uma extensão crítica que o receptor não pôde processar. O campo ***ErrorOID*** identifica a extensão. Se a extensão aparece em um certificado, o campo ***ErrorThumb*** identifica o certificado.
MessageTooBig	A mensagem é muito grande para o recipiente processar.
SignatureRequired	A versão não assinada desta mensagem não é válida.
MessageTooOld	A data da mensagem é muito nova para o receptor processar.
MessageTooNew	A data da mensagem é muito velha para o receptor processar.

ThumbsMismatch	***Thumbs*** enviados em um pedido não assinado não corresponde àqueles retornados ao requisitantes para verificação de ataque de substituição.
UnknownRRPID	Um **RRPID** desconhecido foi recebido.
UnknownLID	Um identificador de local desconhecido foi recebido.
UnknownXID	Um identificador de transação desconhecido foi recebido.
ChallengeMismatch	Um desafio enviado numa mensagem de pedido não corresponde ao desafio na mensagem de resposta. Indica a falha de um desafio recente para a mensagem.

Tabela 4.2: Códigos para as mensagens de erro do SET

4.4. Constituição do Par de Mensagens do SET

As mensagens do SET são constituídas por diversas estruturas de dados que sustentam os itens de informação que transitam de forma recorrente, de mensagens para mensagens, que representam estruturas de controle, dados de aplicações, etc.

Aqui estão exemplificadas as estruturas de dados que compõe as mensagens do SET, através da descrição dos itens componentes do par de Pedido e Resposta de Autorização (**AuthReq / AuthRes**), que é utilizado pelo Comerciante com a mensagem **AuthReq** para solicitar uma autorização de venda do Portal de pagamento, que em resposta recebe do Portal de pagamento a mensagem **AuthRes** (Se refira a notação de sintaxe de mensagens descrita no capítulo 6 para melhor entendimento). Os tipos em negrito indicam a notação utilizada na sintaxe e os componentes de nível de topo, que são constituídos de subníveis de estruturas de dados. Os pontos (.) são usados para indicar subníveis aninhados.

Constituição do Par de Mensagens AuthReq/AuthRes

Porção de Pedido: **Conteúdo de Mensagem**

AuthRes **EncB (M, P, AuthReqData, PI)**

Onde:

AuthReqData	**{ AuthReqItem, [Mthumbs], CaptureNow, [SaleDetail]}**
. AuthReqItem	{ AuthTags, [CheckDigests], AuthReqPayload}
.. AuthTags	{AuthRRtags, TransIDs, [AuthRetNum] }
... AuthRRtags	RRTags (contém dados de identificação da mensagem de pedido-resposta RR que servem como identificador único para um par de menssagens).
... TransIDs	Copiado do *Order Information Data* (OIData)
... AuthRetNum	Identificação de pedido de autorização usado dentro da rede financeira.
.. CheckDigests	*Hash* do *Order Information Data* computado duas vezes, uma vez pelo portador de cartão e mais uma vez pelo Comerciante; usado pelo Portal de Pagamento para comparar e verificar a ligação entre as *Payment Instruction* e *Order Information*.
.. AuthReqPayload	{ SubsequentAuthInd, AuthReqAmt, [AVSData], [SpecialProcessing], [CardSuspect], RequestCardTypeInd, [InstallRecurData], [MarketSpecAuthData],MerchData, [ArqExtensions]}
... SubsequentAuthInd	Expressão Booleana indicando que o Comerciante requer autorização adicional para divisão de remessas.
... AuthReqAmt	Pedido de autorização de quantidade . Deve diferir do PurchAmt por causa das taxas , taxas de remessas, etc.
... AVSData	{ [StreetAddress], Location }
.... StreetAddress	Endereço de remessa do Portador de Cartão.
.... Location	Dados de localização do componente SaleDetail; inclue código do país, nome da cidade, estado ou província, código postal, e um código de locação para o Comerciante.

... SpecialProcessing	Expressão Booleana indicando qualquer tipo de processamento especial requisitado pelo Portal de Pagamento.
... CardSuspect	Um código enumerado que indica que o Comerciante suspeita do Portador de cartão junto com a razão da suspeita.
... RequestCardTypeInd	Indica que o Negociante está requisitando o tipo do cartão que está sendo usado e deseja que ele seja retornado na mensagem de resposta do Portal de Pagamento.
... InstallRecurData	Indica pagamentos periódicos. Um exemplo de uso é no pagamento de uma assinatura de periódico, onde deve ser cobrado cada volume após o fornecimento separado de cada volume da série.
... MarketSpecAuthData	Dados de autorização específicos de mercado (hotel, locadoras de autos, etc).
.... MerchCatCode	código de 4 bytes descrevendo o tipo de negócio do Comerciante, produtos, ou serviços. Definido pelo ANSI X9.10.
. Mthumbs	Impressão digital dos certificados, CRLs, e Identificadores de CRL possuídos pela memória do Comerciante.
. CaptureNow	Expressão Booleana indicando que a captura deveria ser executada se a autorização é aprovada.
.SaleDetail	Contém todos os dados relacionados a uma transação. Os usos dos campos SaleDetail são ditados pelas marcas de cartão, não pela especificação do SET.
PI	A estrutura de dados mais sensível sob o SET. Ele implementa o conceito de Assinaturas Duais para esconder o número de conta de pagamento (PAN) do Comerciante, enquanto permitindo ao Portal de pagamento processar o pedido de autorização enviado pelo Comerciante usando os dados ocultos que somente o Portal de Pagamento pode decifrar.

Porção de Resposta:	**Conteúdo de Mensagem**

(Nota: Conforme notação <...> da mensagem AuthRes são permitidas duas formas de mensagens, porém somente uma ou outra é usada na transação)

AuthRes	**< EncB (P, M, AuthResData, AuthResbaggage), EncBX (P, M, AuthResData, AuthResBaggage, PANtoken >**
AuthResData	**{AuthTags, [BrandCRLIdentifier], [PEThumb], AuthResPaylooad }**

onde:

. AuthTags	Copiado do correspondente na porção de pedidos AuthReq.
. BrandCRLIdentifier	Lista das atuais CRLs para todas as CAs sob a marca.
.PEThumb	Impressões digitais dos certificados do Portal de pagamento que são indicados conforme requerido pelo Comerciante.
. AuthResPayload	{ AuthHeader, [CapResPayload], [ArsExtensions]}
.. AuthHeader	{ AuthAmt, AuthCode, ResponseData, [BatchStatus], [CurrConv]}
... AuthAmt	Copiado da autorização de quantidade do AuthReq Payload .
... AuthCode	Código numérico indicando o resultado do processamento do pedido
... ResponseData	{[AuthValCodes], [RespReason], [CardType], [AVSResult, [LogRefID]}
.... AuthValCodes	{[ApprovalCode], [AuthCharInd], [ValidationCode], [MarketSpecDataID]}
.... ApprovalCode	Código de aprovação assinado pelo Portador de cartão para a transação.

.... AuthCharInd — Código numérico indicando a condição sob a qual a autorização foi executada.

.... ValidationCode — Código de 4 bytes computados para assegurar que os campos requeridos na mensagem de autorização também aparecem em sua subsequente mensagem de esclarecimento.

.... MarketSpecDataID — Código numérico que indica o tipo de dados específicos de mercado fornecidos na autorização como determinado pela rede financeira.

.... RespReason — Código numérico que identifica a entidade de serviço de autorização e a razão para a recusa se a autorização foi recusada.

.... CardType — Código numérico indicando o tipo de cartão usado na autorização.

.... AVSResult — Código numérico de resposta de Serviço de Verificação de Endereço

.... LogRefID — Dados alfanuméricos assinalados à autorização para verificar uma reversão, se requisitada.

... BatchStatus — Componente usado para retornar o status de um lote de transações do Comerciante ou para reconciliar o valor de um lote entre Comerciante e o Portal de Pagamentos.\

... CurConv — { CurrConvRate, CardCurr }

.... CurConvRate — Taxa de conversão de moeda para multiplicar pelo AuthReqAmt para converter para a moeda usada pelo Portador de cartão.

.... CardCurr — Código de moeda para o Portador de cartão conforme definido pela ISO 4217.

.. CapResPayload — { CapCode, CapAmt, [BatchID], [BatchSequenceNum], [CRsPayExtensions] }

... CapCode	Código numérico indicando o status da captura.
... CapAmt	Copiado da correspondente mensagem CapReq.
... BatchID	Identificação de fechamento do lote para finalidades de contabilidade.
... BatchSequenceNum	Número de sequência de um item dentro do lote.
... CRsPayExtensions	Dados financeiros estendidos relacionados aos itens individuais na resposta de captura.
.. ArsExtensions	Dados estendidos usados somente para processamento de resposta de autorização ou uma subsequente reversão de autorização ou pedido de captura.
AuthResBaggage	**{ [CapToken], [AcqCardMsg], [AuthToken]}**
. CapToken	Contém dados que foram requeridos pelo Portal de Pagamento para uma autorização previamente autorizada em tempo de fechamento do lote. Ele é gerado pelo Portal de Pagamento e retornado ao Comerciante.
. AcqCardMsg	Um mecanismo que permite aos Portais de Pagamento enviarem uma mensagem de volta ao Portador de cartão sem permitir que o Comerciante veja o conteúdo. Isto é usado para implementar o conceito de canalização de informação entre o Portal e o Portador de cartão. Isto é um campo opcional no SET. Seu uso é definido pelas políticas contidas no Certificado de Cifragem do Portal de Pagamento.
. AuthToken	Contém dados que foram requeridos pelo Portal de Pagamento para subsequentes autorizações para uma transação. Usadas para remessas parciais de bens ou débitos parcelados.

PANToken	Usada em instâncias onde a supercifragem dos dados do Número de Conta de Pagamento (Payment Account Number – PAN) não é um requisito para esconder os dados do Comerciante.

Capítulo 5

Padrões Criptográficos Aplicados pelo Set

Este capítulo fornece uma visão geral dos padrões criptográficos em que se baseia o SET e expõe a sintaxe para aplicação de operadores criptográficos de assinatura, cifragem e encapsulamento das mensagens do SET.

5.1. Características da criptografia utilizada no SET

Algoritmos de chaves simétricas

O *Data Encryption Standard* (*DES*) é o algoritmo simétrico normalmente usado no SET para proteger os dados financeiros sensíveis (por exemplo, instruções de pagamento).O *Commercial Data Masking Facility* (CDMF) é outro algoritmo de chave simétrica usado para proteger mensagens de Adquirente para o titular de cartão de crédito.

O *DES* é o algoritmo de cifragem de dados simétrico normalmente usado para proteger a informação financeira. Originalmente publicado em 1977 para uso pelo governo dos Estados Unidos para proteger dados valiosos e sensíveis, mas não classificados, este padrão foi adotado subseqüentemente pelo *American National Standard Institute* (ANSI) como o Algoritmo de Cifragem de Dados padrão (*Data Encryption Algorithm* - DEA).

O *DES* especifica um algoritmo de cifragem para cifrar e decifrar blocos de dados de 64 bits sob o controle de uma chave única. O algoritmo está definido pelo *Federal Information*

Processing Standard (FIPS) 46-2, publicado pelo *National Institute of Standards and Technology* (NIST) [12].

O SET usa o modo de Encadeamento de Bloco de Cifra (*Cipher Block Chaining* - CBC) do *DES* , como definido na FIPS 81[16]. A chave tem comprimento de 8 bytes, com cada byte tendo um bit de paridade na posição 0 que rende um comprimento de chave efetivo de 56 bits. A regra de enchimento padrão será usada com o modo *DES* -CBC como descrito abaixo.

A regra de enchimento do SET para *DES CBC* requer que uma fila de enchimento sempre seja anexada ao bloco de texto claro que está sendo cifrado. Este bloco final deve ser um bloco de dados completo, ou um bloco parcial de dados cujo comprimento não seja um múltiplo inteiro do comprimento do bloco. Uma fileira de enchimento é usada no SET independentemente se o bloco final é um bloco de dados parcial ou completo.

A fila de enchimento anexada ao bloco de dados final faz seu comprimento um múltiplo inteiro de oito octetos. Se BL (*Block Lenght*) representa o comprimento em octetos do bloco de dados final, então a fila de enchimento consiste em 8 - (| BL | mod 8) octetos. Cada octeto na fila de enchimento tem como seu valor 8 - (| BL | mod 8).

Quando o comprimento da fila de enchimento é um único octeto, o valor daquele octeto é 01. Quando o comprimento da fila for dois octetos, o valor do dois octetos é 02, e a fila de enchimento usado é '0202'. Quando o comprimento for três, o valor é 03, e a fila de enchimento é '030303', e assim por diante.

O algoritmo CDMF é o algoritmo simétrico para prover confidência dos dados destinados principalmente a guiar a informação do Adquirente para o Titular de cartão através do Comerciante.

A chave CDMF transmitida no protocolo SET é a chave antes de ser transformada para o uso em uma máquina de cifragem/decifragem *DES* . Em outras palavras, uma chave CDMF é tratada apenas como uma chave do *DES* normal.

Algoritmos de Resumo

O Algoritmo de *Hash* Seguro (*Secure Hash Algorithm* - SHA-1) será usado para todos os *hashes* nesta versão do SET, inclusive para os *hashes* usados em assinaturas. Todas as referências aos algoritmos de *hash* serão interpretadas como usando algoritmo de hash SHA-1 definido na FIPS 180-1[23]. O mecanismo de *hash* com chave (HMAC) também usará o SHA-1.

Algoritmos de Chaves Assimétricas

O SET usa algoritmos de cifragem assimétrica de chave pública para assinaturas digitais e envelopes digitais.

Comprimentos de Chave Assimétricas

O algoritmo de chave pública RSA é usado para todas as operações de chaves públicas e certificados, com os seguintes comprimentos de chave. Os comprimentos do quadro que se segue satisfazem os regulamentos de exportação americanos e estão sujeitos a mudança.

Comprimentos de chaves assimétricas utilizadas pelo SET				
Entidade	**Assinatura**	**Cifragem**	**Certificado Assinado**	**CRL Assinada**
Titular de cartão	1024			
Comerciante	1024	1024		
Portal de Pagamento	1024	1024		
CA de Titular de cartão	1024	1024	1024	
CA de Comerciante	1024	1024	1024	
CA de Portal de Pagamento	1024	1024	1024	1024
CA de Marca Geo-política			1024	1024
CA de Marca			1024	1024
CA Raiz			2048	2048

Envelope Digital

Um envelope digital [27] é uma técnica criptográfica genérica para cifrar dados e enviar a chave de cifragem junto com os dados. Geralmente, um algoritmo simétrico é usado para cifrar os dados, e um algoritmo assimétrico é usado para cifrar a chave de cifragem.

O SET usa o método Bellare-Rogaway *Optimal Asymmetric Encryption* (OAEP) [27] em conjunto com seus operadores de encapsulamento criptográfico. Além disso, o SET usa a técnica de *hash* de dados desenvolvida por Matyas e Johnson como uma melhoria para a construção de Bellare-Rogaway básica. As ferramentas do SET e aplicações usarão o processamento OAEP da chave do *DES* e dos dados opcionais antes de sua a cifragem assimétrica usando a chave pública do receptor.

5.2. Padrões criptográficos de chave-pública

A família de padrões de Criptografia de Chave Pública (*Public keys Cryptography Standandards* - PKCS) usada pelo SET inclui:

- Cifragem RSA para construção de assinaturas digitais e envelopes digitais.
- O acordo de chave de Diffie-Helman que define como duas pessoas, sem arranjos prévios, possam concordar sobre uma chave secreta compartilhada que é conhecida somente entre elas e usada em comunicações cifradas.
- Sintaxe estendida de certificado para permitir a adição de extensões do SET (adicionando informações tais quais políticas de uso de certificados ou informação de identificação posterior) do padrão X.509 de certificados digitais.
- Sintaxe de mensagem criptográfica descrevendo como aplicar criptografia de dados relacionados ao SET, incluindo assinatura digital e envelopes digitais.
- Sintaxe de informação de chave privada descrevendo como incluir uma chave privada junto com a informação do algoritmo e um conjunto de atributos para oferecer um modo simples de estabelecimento de confiança na informação fornecida.

- Sintaxe de pedido de certificação descrevendo as regras e conjuntos de atributos necessários para o pedido de certificado da Autoridade de Certificação do SET.

Os seguintes padrões definem como usar os algoritmos e técnicas listadas neste trabalho. Alguns deles descrevem regras de sintaxe, enquanto outros descrevem os protocolos para uso.

- PKCS #1 descreve a sintaxe para a construção de chaves públicas e privadas do RSA.

- PKCS #3 descreve os protocolos usados para estabelecer conexões seguras com acordo de chaves.

- PKCS #5 descreve como derivar uma chave secreta de uma senha usando o *DES* . Isto é usado para implementar técnicas de cifragem baseadas em senha.

- PKCS #6 descreve a sintaxe para estender os usos dos certificados digitais do padrão X.509.

- PKCS #7 descreve a sintaxe para dados que são cifrados para formar posteriormente assinaturas digitais ou envelopes digitais.

- PKCS #8 define uma alternativa para o uso de assinaturas digitais no estabelecimento de confiança da informação através dos usos de informações cifradas com chaves privadas. Ele trabalha em conjunto com o PKCS # 5.

- PKCS #9 define a assinatura de mensagens, pedidos de certificados, certificados estendidos, informação cifrada com chave privada .

- PKCS #10 define os pedidos de certificação em suporte ao uso destes certificados para criar assinaturas digitais e envelopes digitais.

Os padrões PKCS #2 e PKCS #4 foram incorporados ao PKCS #1, e não estão mais ativos. O PKCS fornece as bases para interoperabilidade e compatibilidade significativa com os padrões industriais existentes tais como *Open Systems Interconnect* (OSI). Uma das metas do projeto do PKCS foi a eventual incorporação ao OSI para uso em escala global.

O PKCS #7, em particular, é usado como a base para o encapsulamento de mensagem para todas as mensagens do SET. Desse modo, as regras de codificação ASN.1 são preservadas,

orientando para um formato único de definição de todos os pares de mensagens do SET através da especificação.

5.3. Os formatos do PKCS #7

Para assegurar a interoperabilidade e a habilidade de atualização, os Padrões de Criptografia de Chave Pública (PKCS) #7, Padrão de Sintaxe de Mensagem Criptográfica [27], é usado como a base para os métodos de encapsulamento criptográfico usados nas mensagens do SET.

Os formatos PKCS #7 são usados para representar os dados envelopados em mensagens do SET. A ASN.1 e sua regras de codificação, um conjunto de padrões internacionais, é usado em conjunto com a especificação do PKCS #7. Usando a ASN.1 para definir as mensagens do SET , um único formato é usado ao longo da especificação inteira do SET.

O SET usa os seguintes métodos de encapsulamento de dados do PKCS #7:

- ***SignedData***, para encapsulamento de dados assinados,
- ***EnvelopedData***, para encapsulamento de dados codificados,
- ***EncryptedData***, para encapsulamento de dados cifrados com chaves simétricas,
- ***DigestedData***, para encapsulamento de dados resumidos.

As mensagens assinadas contêm todos os certificados e CRLs necessários para o receptor verificar as suas assinaturas; o receptor pode indicar os certificados que ele tenha previamente validado e armazenado (*cached*) usando as impressões digitais na mensagem de pedido correspondente.

As CRLs e os certificados de assinatura estão implícitos nos tipos de mensagem assinadas. Seguindo o PKCS #7 , estes estão contidos no campo ***Certificates*** do tipo ***SignedData***. Além disso, o SET inclui certificados de troca de chave em blocos ***SignedData***; estes estão implícitos no protocolo. A linguagem PKCS #7 expressamente permite este uso.

Nos casos onde certificados ou CRLs requerem autenticação de origem ou proteção de integridade mas não são encapsulados no ***EnvelopedData***, eles são transportados nos tipos ***SignedData*** do PKCS #7 recursivamente encapsulados .

5.3.1. *SignedData*

O tipo ***SignedData*** do PKCS #7 é mostrado a seguir para auxiliar no entendimento do processo de assinatura do SET. São permitidas ocorrências múltiplas de informações de assinantes (***SignerInfos*****)** dentro de um único bloco ***SignedData***; porém, uma mensagem é assinada por não mais que duas partes no SET.

Atributos Autenticados

O ***SignedData*** no SET sempre inclui dois atributos autenticados: o ***contentType*** e o ***messageDigest***. O tipo de conteúdo que está sendo assinado deverá ser incluído no campo do atributo de tipo de conteúdo (***contentType***) e o resumo dos dados que estão sendo assinados é incluído no campo do atributo resumo de mensagem (***messageDigest***).

Os identificadores de objeto para o atributo ***contentType*** são definidos para identificar unicamente cada um dos tipos da ASN.1 do SET que podem aparecer no ***SignedData***.

Por exemplo, considere a assinatura de uma estrutura de dados do tipo ASN.1 denominado ***Stuff.*** O resumo SHA-1 da codificação DER deste tipo é computado. A estrutura de dados assinada (***envelopedData***) conterá o identificador do objeto ***id-set-stuff*** no atributo ***contentType*** e o resumo de dados da estrutura ***Stuff*** no atributo ***messageDigest*** como mostrado a seguir.

Tipo de Conteúdo	*id-set-stuff*
Resumo de Mensagem	*SHA-1(Stuff)*

Considerando ainda o exemplo da estrutura de dados ***stuff*** que deverá ser autenticada, pois é identificada pelo conteúdo do tipo de conteúdo ***id-set-stuff***. Então, o resumo SHA-1 destes dados é computado e o resultado é cifrado usando a chave privada do signatário; e isto é o resumo assinado que é colocado no campo Resumo Cifrado (***messageDigested***).

A Figura 5.1 detalha os campos componentes da estrutura ***SignedData***. Na figura os valores informados entre parênteses sob os campos são os conteúdos desse bloco quando utilizado pelo SET.

5.3.2. *EnvelopedData*

O tipo ***EnvelopedData*** do PKCS #7 é mostrado abaixo para ajudar no entendimento do processo de cifragem do SET. São permitidas ocorrências múltiplas de Informações de Receptor (***RecipientInfos***) no ***EnvelopedData*** do PKCS #7, ou seja um único envelope pode ter vários destinatários. Entretanto, nas mensagens do SET somente um Receptor é utilizado nesta estrutura.

A Figura 5.2 demonstra os componentes do tipo ***EnvelopeData***. Na figura os valores informados entre parênteses sob os campos são os conteúdos desse bloco quando utilizado pelo SET.

SignedData

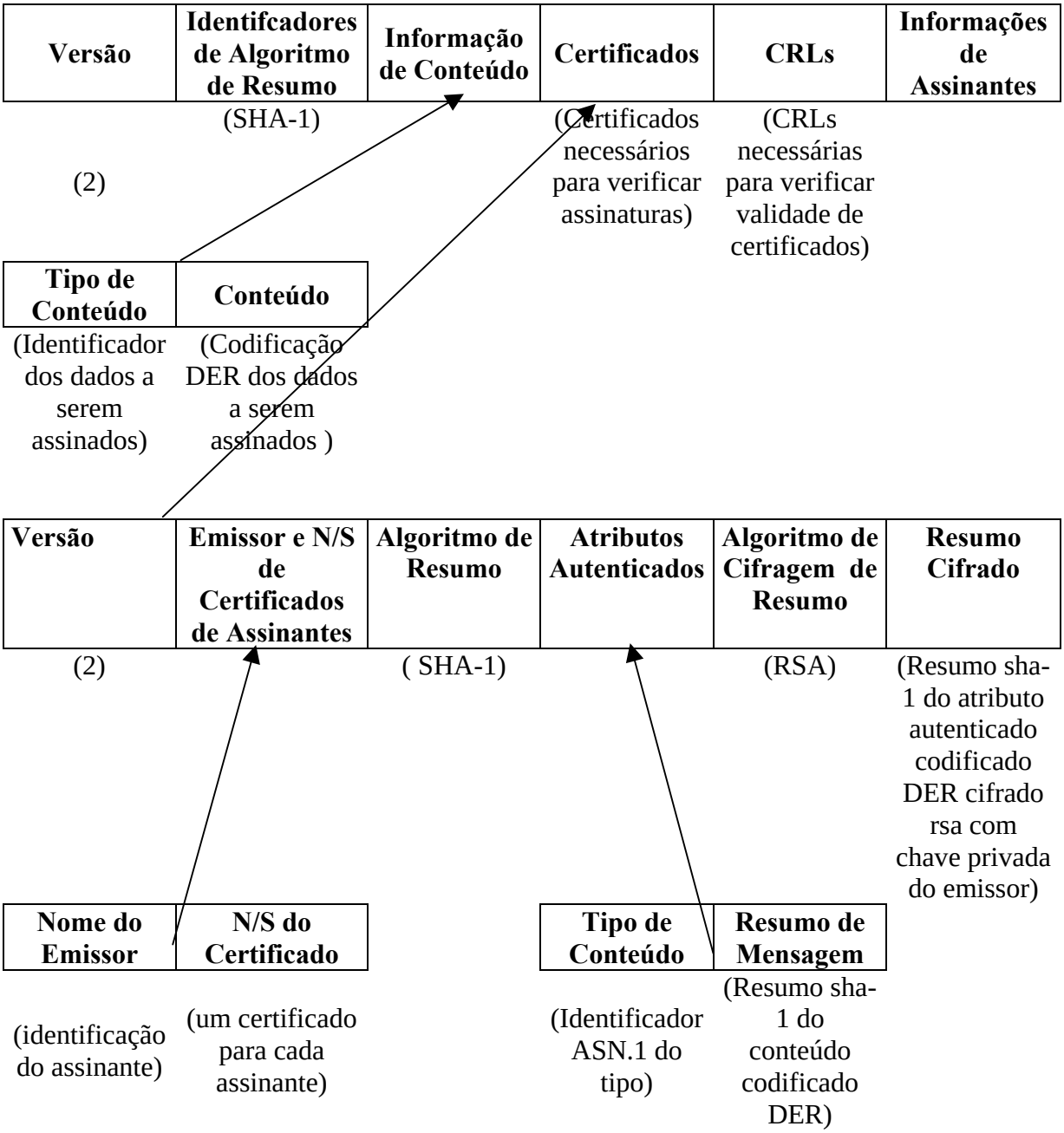

Figura 5.1: *SignedData*

5.3.3. *EncryptedData*

A Figura 5.3 demonstra os componentes do tipo ***EncryptedData***. Os valores informados entre parênteses sob os campos são os conteúdos desse bloco quando utilizado pelo SET.

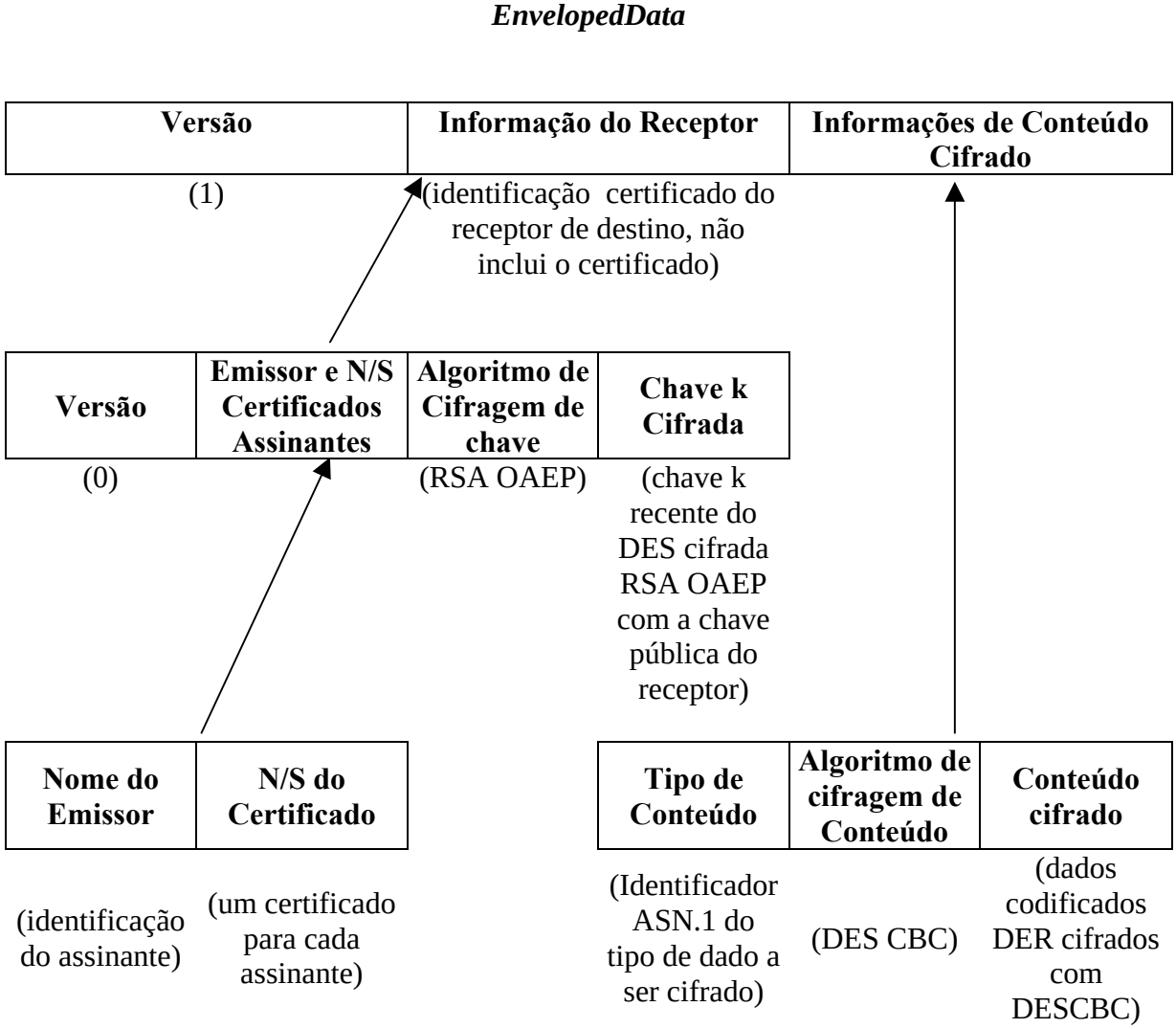

Figura 5.2: *EnvelopedData*

5.3.4. *DigestedData*

A construção ***DigestedData*** do PKCS #7 é mostrada na Figura 5.4 para ajudar a definição do processo *hashing*. Os valores informados entre parênteses sob os campos são os conteúdos desse bloco quando utilizado pelo SET.

EncrypetedData

Versão	**Informações de Conteúdo Cifrado**	
(0)		
Tipo de Conteúdo	**Algoritmo de Cifragem de Conteúdo**	**Conteúdo Cifrado**
(Identificador ASN.1 do tipo de dado a ser cifrado)	(DES CBC ou DES CDMF)	(dados codificados DER cifrados DES CBC ou DES CDMF com chave)

Figura 5.3: *EncryptedData*.

DigestedData

Versão	**Algoritmo de Resumo**	**Informações de Conteúdo**	Resumo
(0)	(SHA 1)		(resumo SHA-1 dos dados codificados DER, incluindo etiquetas e octetos de comprimento)
		Tipo de Conteúdo	**Conteúdo**
		(identificador ASN.1 do tipo de dado a ser resumido)	(vazio neste tipo parametrizado DD {} utilizado pelo SET)

Figura 5.4: *DigestedData*

5.4. Notação Abstrata de Composição de Mensagens

A notação apresentada a seguir é utilizada na representação sintática das mensagens do SET, incluindo a representação dos processamentos criptográficos aplicados sobre elas.

Conceito	Notação	Conceito
Tupla	{A, B, C}	Um agrupamento de zero ou mais elementos de dados, que representam as mensagens e as condições ocasionalmente usadas na sua intercambialidade. As Tuplas são denotadas através de identificadores que são símbolos alfanuméricos. Esta notação significa "a tupla contendo A, B, e C" que devem , ser eles próprios, também tuplas.
Componente	T = {A, B, C}	Pode ser dado um nome às tuplas, no caso T.A, T.B, e T.C se referem às respectivas tuplas componentes de T.
Concatenação Ordenada	A | |B | |C	Esta notação significa que uma explícita concatenação ordenada dos itens A, B, e C é necessária.
Opcional	[A]	Esta notação significa que um item A é opcional.
Seleção	<A, B, C>	Esta notação significa que exatamente um de A, B, e C tem que aparecer. Esta é uma notação de seleção.
Seleção Opcional	[<A, B, C>]	Esta notação significa que a seleção é opcional; quer dizer, que nada ou exatamente um de A, B, e C deve aparecer.
Instâncias Múltiplas	{A +}	Esta notação significa uma tupla contendo um ou mais instâncias de A.
	{A *}	Esta notação significa uma tupla contendo zero ou mais instâncias de A.
	{[A] +}	Esta notação significa uma tupla contendo: uma ou mais instâncias de A, em um arranjo ordenado, onde cada instância de A é opcional (quer dizer, pode ser nula).
Ou-exclusivo	⊕	Este símbolo denota uma operação ou-exlusivo bit a bit (XOR).

Tabela 5.1: Notação abstrata de mensagens

Notação de tratamentos criptográficos

São apresentadas as notações para os tratamentos criptográficos de *hashing*, assinatura, assinatura dual, cifragem, e os tratamentos de encapsulamento dos elementos de dados que são utilizados pelo SET.

5.4.1. Notação de *Hashing*

O *hashing* transforma um elemento de dados num único valor ou impressão digital para aquele elemento de dados. O SET usa algoritmos tais que as chances de duas computações resultarem num mesmo valor de *hash* são 1 em 10^{48}. O *hashing* é usado pelo SET para criar resumos de mensagem que permitem a verificação da integridade dos dados recebidos.

Usando o algoritmo SHA-1 de 160 bits para fazer o *hash* de uma tupla t, a notação de *hash* aparece como $H(t)$.

Usando um algoritmo HMAC- SHA-1 de 160 bits sobre a tupla t usando a chave K, a notação de *hash* com chave aparece como $HMAC(t,K)$.

A operação *Digest Data* (DD) sobre a tupla t corresponde a uma operação *hash* SHA-1 de 160 bits embutida no formato definido como *DigestedData* do PKCS. Sua notação aparece como $DD(t)$.

A ligação entre duas mensagens é indicada via uma referência ou um ponteiro entre os elementos $t1$ e $t2$. Isso aparece como $L(t1,t2)$.

A notação que corresponde aos operadores baseados em *hash* usados pelo SET é resumida no quadro que segue.

Notação	Operador	Descrição dos Operadores de *Hash*
$H(t)$	*Hash*	Hash SHA-1 de 160 bits da tupla t.
$HMAC(t,k)$	Mecanismo *Hash* com Chave	*Hash* com chave de 160 bits da tupla t usando a chave k baseado em $1HMAC-SHA-1$. $HMAC(t,k)=H((k\oplus opad)\|\|H((k\oplus ipad)\|\|t))$ onde: *ipad* é o byte 0×36 repetido 64 vezes; *opad* é o byte $0\times 5C$ repetido 64 vezes; e $\oplus$ é a função XOR.
$L(t1,t2)$	Ligação	Uma referência.ponteiro, ou ligação para $t2$ é incluida com $t1$; equivalente a tupla $\{t1,H(t2)\}$

5.4.2. Notação de Assinatura

A assinatura de uma mensagem envolve a computação de um resumo para a porção de dados e sua afixação aos dados. Uma vez recebidos, o valor de *hash* para a porção de dados é computado e comparado ao valor de *hash* recebido. Se eles casarem, os dados devem ter chegado inalterados.

A ilustração abaixo se refere ao conceito de uma mensagem assinada digitalmente

Conteúdo da Mensagem		Resumo da Mensagem

Uma mensagem assinada usa a assinatura da entidade **s** numa tupla **t** através de um algoritmo RSA com um *hash* SHA-1, indicado como $S(s,t)$.

Os operadores de Somente Assinatura (*Signature Only – SO*) usam a assinatura de uma entidade s numa tupla t sem a porção de texto claro da tupla t. O SO corresponde às assinaturas externas do PKCS #7 para dados envelopados. Eles são indicados como $SO(s,t)$. Estes operadores são utilizados para assinaturas duais usada pelo SET.

A notação que corresponde aos operadores de assinatura usada pelo SET é resumida no quadro seguinte.

Notação	Operador	Descrição dos Operadores de Assinatura
$S(s,t)$	Assinatura	A assinatura pela entidade s na tupla t, incluindo o texto claro t. O algoritmo normal de assinatura do SET é o RSA com *hash* SHA-1. Corresponde ao *SignedData* do PKCS #7.
$SO(s,t)$.	Somente Assinatura	A assinatura pela entidade s na tupla t, porém não incluindo o texto claro t no envelope. SO corresponde a assinatura externa do PKCS # 7. O algoritmo de assinatura normal do SET é o RSA com *hash* SHA-1.

5.4.3. Notação de Cifragem

A Cifragem assimétrica ou envelopes digitais são criados pela cifragem da tupla t usando uma chave DES recente que é subsequentemente inserida num envelope PKCS #7 destinado ao receptor r, usando sua chave pública obtida de seu certificado para cifragem RSA. A Cifragem Assimétrica corresponde ao *EnvelopedData* do PKCS #7 da tupla t cifrada pela entidade r. Isto é indicado como $E(r,t)$.

A Cifragem com Integridade, similar ao operador de cifragem E, fornece um grau extra de integridade pela inclusão de um *hash* da tupla t com a tupla t cifrada . Sua notação é $EH(r,t)$. A Cifragem Extra é também similar ao operador de cifragem E, exceto que t e p são as partes de uma mensagem de duas partes. A tupla t é cifrada usando a cifragem assimétrica ordinária, e p é um parâmetro ou a porção que é submetida a um processamento como dado extra do OAEP. A Cifragem Extra serve como um mecanismo de ligação entre as duas mensagens. Sua notação é $EX(r,t,p)$.

A Cifragem Extra com Integridade é similar ao operador de cifragem extra EX, porém inclui também um *hash* da tupla t. Sua notação é $EXH(r,t,p)$.

A Cifragem Simétrica com Chave Fornecida é uma instância do *EncryptedData* do PKCS # 7 e indica a operação para cifrar o texto plano da tupla t com a chave K fornecida usando o DES ou CDMF. Sua notação é $EK(k,t)$.

A notação que corresponde aos operadores de cifragem usada pelo SET é resumida no quadro a seguir:

Notação	Operador	Descrição dos Operadores de Cifragem
$E(r,t)$	Cifragem Assimétrica (Envelope digital)	Cifragem da tupla t com uma chave simétrica k recente gerada. Após então, cifragem assimétrica da chave k com OAEP $(OAEP(k))$, usando a chave pública da entidade r, obtida do seu certificado numa transação prévia. O resultado é inserido num envelope PKCS #7 a ser enviado para a entidade r. O *DES* é algoritmo normal de cifragem simétrica. Corresponde ao *EnvelopedData* do padrão PKCS #7.
$EH(r,t)$	Cifragem com Integridade	Similar ao operador E exceto que o envelope PKCS #7 contém $OAEP(\{k, H(t)\})$ para uma garantia de integridade quando a assinatura não está disponível.
$EX(r,t,p)$	Cifragem Extra	Similar ao operador E exceto que t e p são as partes de uma mensagem de duas partes; t é a tupla a ser ligada a p e sujeita a usual cifragem simétrica, e p é um parâmetro, ou a parte a ser sujeita a " processamento OAEP extra". O t é ligado à p. O $OAEP(\{k, p\})$ é inserido no envelope PKCS #7 para a entidade r. O slot de t é chamado slot ordinário de EX, e o slot de p é chamado o slot extra de EX.
$EXH(r,t,p)$	Cifragem Extra com Integridade	Similar ao operador EX exceto que $OAEP(\{K, H(t), p\})$ é inserido no envelope do PKCS # 7. E com a exigência que o processamento do receptor verifique $H(t)$, como em EH.
$EK(h,t)$	Cifragem Simétrica com uma chave k fornecida	A cifragem simétrica da tupla **t** usando a chave secreta k. Corresponde a uma instância do *EncryptedData* do PKCS #7.

5.4.4. Notação de Encapsulamento

O Encapsulamento combina o uso de cifragem e assinaturas para assegurar o mais alto grau de integridade e autenticação da entidade fim. O Encapsulamento é executado em quase todas as mensagens do SET .

O Encapsulamento Simples com Assinatura é uma instância do *SignedData* no *EnvelopedData* do PKCS #7: uma mensagem assinada, e então cifrada. Sua notação é $Enc(s,r,t)$. O Encapsulamento Simples com Assinatura e Chave Fornecida cria mensagens assinadas que são cifradas com uma chave secreta compartilhada que foi fornecida pelo emissor numa mensagem prévia. Sua notação é $EncK(k,s,t)$. O Encapsulamento Extra com Assinatura cria a cifragem da mensagem de duas partes, com a primeira parte da mensagem,

t, num *slot* ordinário do OAEP e a segunda parte da mensagem , p, num *slot* extra OAEP. Sua notação é $EncX(s,r,t,p)$. O Encapsulamento Simples com Assinatura e Bagagem cria mensagens que contem "bagagem extra". Exemplos de bagagem extra incluem *token* que são retornados para posterior processamento com uma resposta bem sucedida a um pedido de autorização de carga, ou outra informação sensível (por exemplo números de conta de cartão) que é retornada ao Comerciante para gravação de registro ou processamento de contestação. Isto é desejável para super cifrar tais dados para adicionar níveis de segurança e proteção. A notação para o operador é $EncB(s,r,t,b)$. O Encapsulamento Extra com Assinatura e Bagagem é usada para mensagens de duas partes com bagagem externa. Sua notação é $EncBX(s,r,t,b,p)$.

Os operadores que combinam operadores de assinatura e operadores de cifragem estão mostrados no quadro:

Notação	Operador	Descrição dos operadores de encapsulamento
$Enc(s,r,t)$	Encapsulamento Simples com assinatura	Mensagem assinada e então cifrada. Corresponde a um exemplo da *SignedData* encapsulada no *EnvelopedData* do PKCS #7.
$EncK(k,s,t)$	Encapsulamento Simples com Assinatura e Chave Fornecida	Mensagem assinada e cifrada com uma chave secreta fornecida. Corresponde a uma instância do *EncryptedData* do PKCS 7.
$EncX(s,r,t,p)$	Encapsulamento Extra com Assinatura	Mensagem de duas partes t e p, assinada, cifrada com a primeira parte da mensagem cifrada no *slot* ordinário e a segunda parte da mensagem no *slot* extra. O slot de t é chamado slot ordinário de E, e o slot de p é chamado o slot extra de E.
$EncB(s,r,t,b)$	Encapsulamento Simples com Assinatura e Bagagem	Mensagem contendo bagagem externa assinada e cifrada.
$EncBX(s,r,t,b,p)$	Encapsulamento Extra com Assinatura e Bagagem	Mensagem de duas partes com bagagem externa assinada, cifradas com operador E, cifrada com a primeira parte da mensagem cifrada no *slot* ordinário e a segunda parte da mensagem no *slot* extra.

5.5. Outras Implicações Criptográficas

Uma área de consideração especial para os desenvolvedores de ferramentas do SET e aplicações é a implementação de geração de número aleatório usado para chaves e *nonces*. Embora uma definição precisa de aleatoriedade esteja fora do escopo da especificação do SET, os desenvolvedores de produtos precisam estar cientes da importância deste aspecto na sua implementação. A geração pobre de chaves e métodos de envio devido ao uso de números aleatórios fracos são causas de quedas comuns de implementações criptográficas.

Para propósitos criptográficos, uma vez que uma semente forte é escolhida, ou ela será usada uma vez somente ou será usada exclusivamente em um gerador de número aleatório criptograficamente seguro. Também, cada instância do algoritmo de geração de número aleatório terá sua própria semente de geração de chave independente.

O SET define diversos campos como *nonces, salts* ou *freshness challenges* como técnicas usadas para derrotar “ataques de dicionário”. Juntamente com outras técnicas como ID de transações, *freshness challenges* ajudam na implementação da característica de idempontência do SET. Um exemplo deveria ser onde múltiplas cópias da mesma mensagem de pedido são recebidas, porém somente uma deveria ser processada. Se um receptor recebe cópias duplicadas de uma mensagem (mesmo ID de transação, mesmo *nonce*) , eles podem verificar que a mensagem já foi processada. Essencialmente o *nonce* é um número aleatório que é gerado com a finalidade de ser copiado na mensagem de retorno, ajudando a assegurar que ela veio de volta do receptor original da mensagem e não de qualquer outro. Cada mensagem que requer *nonces* irá usar valores únicos, então garantido um nível maior de integridade das mensagens e processamento do aquele que é possível em sua ausência.

Independência de Algoritmo

Embora esta versão atual do SET seja explícita sobre os algoritmos criptográficos que serão processados pelo Titular de cartão, Comerciante, e sistemas de Portal de pagamento, os operadores de encapsulamento criptográficos do protocolo foram projetados para ter algoritmos independentes. Todos os conjuntos de objetos de informações de algoritmo ASN.1

são cifrados com o marcador de extensão (…) para permitir que algoritmos de objetos adicionais sejam adicionados a versões futuras da especificação, desde que mantendo a compatibilidade das versões anteriores com esta versão do SET. Os algoritmos simétricos para proteção das mensagens do Adquirente para o titular de cartão são outro exemplo de como o SET estará se movendo para este objetivo de longo alcance.

Tokens de *Hardware*

Dependendo das políticas estabelecidas pelo Adquirente e Marca, os *tokens* de *hardware* também podem ser usados por sistemas que dão suporte ao SET. Um *token* de *hardware* é definido como um módulo de *hardware* criptográfico que não permite revelação da chave privada. É previsto que os *tokens* de *hardware* devam ser integrados aos sistemas que dependem de um nível mais alto de garantia de confiança, como o Portal de Pagamento ou a CA.

Execução de funções criptográficas em *tokens* de *hardware*:

- As CAs usarão um *token* de *hardware* para todas as operações de chaves privadas.
- Os Portais de Pagamento irão adotar o uso de *token*s de *hardware*; o seu uso deve ser designado pela política do Adquirente ou da Marca.
- Os Comerciantes deveriam adotar o uso de *tokens* de *hardware*; o seu uso deve ser designado pela política do Adquirente ou da Marca.
- Titulares de cartão de crédito deveriam adotar o uso de *tokens* de *hardware*.

Idempotência

Quando uma operação pode ser executada em qualquer número de vezes, sem causar dano, é dito que é idempotente. Da perspectiva do SET, a idempotência é uma propriedade de como um receptor responde a uma mensagem.

Qualquer pedido no SET que não recebe uma resposta se ressentirá disto desde que é impossível ao remetente saber se o pedido ou a resposta se perderam. A mensagem re-

transmitida será na forma de bits idêntica à mensagem de pedido original. Em geral, uma mensagem duplicada não é uma condição de erro.

O protocolo do SET é projetado para trabalhar em ambientes onde a entrega de mensagem não é garantida. Se uma aplicação do SET não recebe uma resposta em um período razoável de tempo (como definido pela aplicação ou possivelmente em resposta a uma pesquisa de usuário), re-envia a mensagem. Quando a aplicação SET receptora determina que já processou a mesma mensagem previamente, recupera a resposta prévia e envia aquela resposta prévia novamente.

Quando o emissor de uma mensagem não receber uma resposta, é impossível determinar se o pedido foi perdido ou a resposta foi perdida. Para exacerbar mais esta condição, é bastante possível que o pedido original pode ter estado em algum lugar na rede e então eventualmente esteja simplesmente atrasado no processamento e um novo pedido idêntico já tenha sido retransmitido. Por isso o protocolo permite que o emissor repita o pedido com a garantia que o resultado será o mesmo ainda que o pedido estivesse perdido ou a resposta estivesse perdida.

Nem todas as mensagens do SET requerem idempotência. O pedido de compra requer idempotência. Por outro lado, o pedido de investigação, por exemplo, foi projetado para ser enviado a qualquer hora e assim não é necessário que um comerciante armazene cada pedido de investigação para determinar se aquele pedido já foi recebido; ele simplesmente devolve o estado atual da transação na resposta de investigação.

Os produtos do SET garantirão idempotência do protocolo pelo exame da transação (XID) e identificadores do par pedido/resposta (RRPID). Por exemplo, um portal de pagamento rejeitará tentativas de repetição de pedidos de autorização dos comerciantes. Ele detectará estas tentativas examinando o RRPID do pedido de autorização e XID da instrução de pagamento embutido, separadamente assinado e cifrado pelo titular de cartão.

Se uma aplicação do SET detecta que está sendo sujeito a um ataque malicioso envolvendo um ou mais tipos de mensagens idempotentes do SET, não é necessário responder a estas mensagens nesta situação.

Capítulo 6

Processamento Criptográfico das Mensagens do Set

Este capítulo provê uma visão geral do processamento criptográfico aplicado às mensagens pelo protocolo SET. Para tanto, inicialmente veremos o conjunto de mensagens usadas pelo SET para condução dos protocolos de gerenciamento de certificados e de pagamentos, com uma breve discussão da funcionalidade dos pares de mensagens envolvidos.

6.1. Conjunto de Mensagens do SET

Protocolo de Mensagens de Gerenciamento de Certificados:

O SET possui os seguintes pares de mensagens, de pedido e de resposta, no protocolo de gerenciamento de certificados:

- **CardCInitReq / CardCInitRes**: Mensagens de inicialização de certificados de Portador de cartão.
- **Me-AqCInitReq / Me-AqCInitRes**: Mensagens de inicialização de certificados de Comerciante ou Adquirente.
- **RegFormReq / RegFormRes**: Mensagens de Formulário de Registro de Portador de Cartão. O par de mensagem de Pedido e Resposta de Formulário de Registro é usado pelos Portadores de Cartão no protocolo de Pedido de Certificado.

- **CertReq / CertRes**: Mensagens de Pedido e Resposta de Certificados. Este par é usado por todas as entidades finais na última fase do fornecimento de certificado e processo de renovação.

- **PCertReq / PCertRes**: Mensagens de pedido e resposta de Pesquisa de Certificados.

- **CertInqReq / CertInqRes**: Mensagens de Investigação de Certificados. Este par deve ser usado pelas entidades finais para determinar o status dos certificados que eles haviam pedido as CAs.

Protocolo de Mensagens do Sistema de Pagamentos

O SET possui os seguintes pares de mensagens, de pedido e de resposta, no protocolo do sistema de pagamentos:

- **PInitReq / PInitRes**: Mensagens de inicialização de Pagamento. Este par de mensagens é usado pelo Portador de cartão para iniciar um pedido de compra de um Comerciante.

- **PReq / PRes**: Mensagens de pedido e resposta de Compra. Este par de mensagens é usado pelo Portador de cartão para enviar os dados do Pedido de Compra para um Comerciante e é considerado o coração do protocolo de pagamento do SET. Ele consiste de duas partes: uma parte de informações do pedido (*Order Instructions* –OI) e uma outra parte de informações de pagamento (*Payment Instructions* – PI). As informações de instruções de pagamento são destinadas ao processamento pelo Portal de Pagamento e são transmitidas somente ao Portal, através do Comerciante. Cada item OI e PI é assinado separadamente usando assinatura dual .

- **InqReq / InqRes**: Mensagens de Investigação. As mensagens deste par são usadas pelo Portador de cartão para pedir o status de uma transação ao Comerciante.

- **AuthReq / AuthRes**: Mensagens de pedido e resposta de Autorização. O par de mensagens é usado pelo Comerciante (M) para obter a autorização para uma venda do Portal de Pagamento (P). Ele é usado para transações de somente autorização, onde o

pedido de captura é feito posteriormente, e para transações de autorização com captura, quando permitidas pelas regras do Adquirente.

- **CapReq / CapRes**: Mensagens de pedido e resposta de Captura. Estas mensagens são enviadas do Comerciante ao Portal de Pagamento para completar aquelas transações que tiveram uma prévia resposta de pedido de autorização bem sucedidas.

- **AuthRevReq / AuthRevRes**: Mensagens de pedido e resposta de Reversão de Autorização. O par de mensagens de Reversão é usado para alterar quantidades de uma autorização previamente ou cancelar totalmente uma resposta de autorização aprovada previamente. O uso deste par de mensagens é opcional sob o SET. Ele pode ser enviado pelo Comerciante para o Portal de Pagamento em qualquer tempo depois da autorização aprovada, porém antes do pedido de captura da transação. Também pode ser usado para parcelar um pedido previamente não parcelado, como também deve ocorrer com situação de estorno do pedido.

- **CapRevReq/ CapRevRes**: Mensagens de pedido e resposta de Reversão de Captura. O par de mensagens de Pedido e Resposta de Reversão de Captura, é utilizado pelo Comerciante ao Portal de pagamento, para reverter um Pedido de Captura prévio bem sucedido. Elas devem ser usadas para cancelar uma venda capturada onde um consumidor tenha mudado de idéia antes da aceitação da entrega dos bens, ou os tenha devolvido dentro de um pequeno período de tempo após a venda. Quando uma Reversão de Captura não pode ser usada, depois de ocorrido um período de tempo após a captura original, o par de Pedido/Resposta de Crédito deve ser usado em seu lugar .

- **CredReq / CredRes**: Mensagens de pedido e resposta de Crédito. Este par de mensagens de Pedido de Crédito e Resposta é usado pelo Comerciante para pedir um crédito para uma transação previamente autorizada e capturada, quando os bens foram devolvidos pelo portador de cartão e um crédito para esse cartão de pagamento é desejado.

- **CredRevReq / CredRevRes**: Mensagens de pedido e resposta de Reversão de Crédito. Este par de mensagens é usado pelo Comerciante para pedir uma reversão de um Pedido de Crédito previamente aprovado pelo Portal de Pagamento. Ele deve ser necessário quando um Portador de Cartão recebe o crédito por uma compra por conta de um

compromisso de devolução, porém a devolução dos bens ao Comerciante de fato não ocorre.

- **BatchAdminReq / BatchAdminRes**: Mensagens de pedido e resposta de Administração de Lotes. As mensagens deste par são usadas para os pedidos de Administração de Lote, enviados do Comerciante ao Portal de Pagamento. Tais solicitações são usadas para permitir a administração de lotes de transações esperando captura. As instruções devem indicar a abertura de um novo lote, eliminação de um lote, fechamento de um lote, transferir uma cópia do conteúdo do lote, ou retornar o status do lote. A porção de resposta da mensagem indica o status do fechamento do lote relativo ao pedido, junto com os demais resultados que existam.

6.2. Processamento da Envoltória de Mensagem

Consulte a Tabela 4.1: Descrição dos Campos da Envoltória de Mensagens do SET no capítulo 4 deste trabalho para melhor entendimento do procedimento que se segue.

A entidade emissora da mensagem irá assegurar que os conteúdos de mensagem foram formatados corretamente e foram encapsulados baseados no tipo de mensagem. Os dados adicionais como certificados, CRLs, e BrandCRLIdentifiers serão incluídos se qualquer porção da mensagem está sendo assinada pela entidade emissora .

As aplicações do SET, antes do envio da mensagem, implementarão o procedimento de composição de envoltória de mensagem descrito no quadro abaixo, ou procedimentos funcionalmente equivalentes, para todas as mensagens enviadas. Ele representa o processamento padrão requerido a cada vez que uma mensagem é enviada. Note que a criptografia será diferente baseada no tipo de cifragem requerida e se a mensagem é assinada; isto é especificado para cada mensagem.

Passo	Procedimento de Composição da Envoltória de Mensagem
1	Geração da mensagem do SET como apropriado. As mensagens são aquelas descritas anteriormente, agrupadas funcionalmente em transações dos protocolos de sistema de pagamento ou de gerenciamento de certificados.
2	Inserção dos números da versão atual e revisão no **MessageWrapper** (atualmente 1 e 0 respectivamente).
3	Inserção da data (inclusive hora) . Nota: A data deve ser precisa para assegurar à entidade receptora a capacidade de verificar a idade das mensagens corretamente.
4	Preenchimento das informações *MessageIDs*. Se não há *MessageIDs* na Mensagem (por exemplo, mensagens de certificado), então este campo deve ser omitido.
5	Inserção do RRPID. Se este for um pedido, o RRPID será gerado, e armazenado para comparar com a resposta. Se esta for uma mensagem de resposta, o RRPID será copiado do pedido.
6	Inserção do *SWIdent*. Este é um sequência que identifica o vendedor e a versão do software do vendedor.
7	Inserção da Mensagem (como um tipo aberto da ASN.1).
8	Execução da codificação DER da mensagem envolvida.
9	Passagem da mensagem codificada do passo 8 para o mecanismo de transporte. Dependendo do mecanismo de transporte, a mensagem ainda pode ser envolvida posteriormente por outro protocolo do mecanismo de transporte (tal qual com um cabeçalho MIME ou HTTP).

A entidade receptora da mensagem executará o procedimento de decomposição da envoltória de mensagem do quadro que se segue. Ela se assegurará que os conteúdos da mensagem foram formatados corretamente e foram encapsulados baseados no tipo de mensagem. Os dados adicionais como certificados, CRLs, e BrandCRLIdentifiers serão extraídos da mensagem para autenticar qualquer assinatura digital aplicada pela entidade emissora. A memória do sistema da entidade receptora deve ser atualizada para refletir os novos certificados, CRLs e BrandCRLIdentifiers.

Passo	Procedimento de Decomposição da Envoltória de Mensagem
1	Se o mecanismo de transporte envolveu uma mensagem do SET antes de transmití-la, remoção da envoltória como requerido pelo mecanismo de transporte.
2	Validação do formato e do conteúdo dos campos da envoltória da mensagem: versão, revisão, data/hora e tipo da mensagem. No caso de falha: A. Retorno de Erro com o conjunto *ErrorCode* apropriado para o erro. B. Parar o processamento da mensagem.
3	Usando o RRPID, comparação e atualização do log do sistema de mensagem duplicada manipulando-o conforme as diretrizes da marca operadora.
4	Decodificação DER da mensagem.
5	Se a mensagem contém dados assinados, então execução das seguintes ações: A. Atualização da memória do sistema com quaisquer CRLs recebidas. B. Para cada certificado recebido, execução do processamento de Validação da Cadeia de Certificado . C. Verificação da assinatura da mensagem.
6	Se a mensagem contem dados encapsulados, execução da operação inversa do encapsulamento (decifragem), de acordo com o tipo de encapsulamento dos conteúdos da mensagem, inclusive o Passo 5 acima, se o dados encapsulados contém dados assinados.
7	Extração de quaisquer BrandCRLIdentifiers incluídos na mensagem e atualização da memória do sistema, com a verificação de que todas as CRLs identificadas no BCI (*BrandCRLIdentifier*) estão na memória do sistema; caso contrário aborto do processamento da mensagem.
8	Processamento da mensagem.
9	Atualização do *log* do sistema para refletir o estado desta transação.

6.3. Validação da Cadeia de Certificados

A validação da cadeia de certificado pela entidade receptora requer que cada certificado no caminho seja verificado e que cada certificado mapeie corretamente a CA que o emitiu. Os procedimentos de validação serão obrigatórios para todos os níveis da cadeia. Por exemplo, uma aplicação de Portador de cartão validará o Comerciante, a CA do Comerciante, a CA de Marca, e os certificados da CA Raiz e marcas de cartão de pagamento relacionadas. O processo de validação é constituído dos seguintes componentes:

- Validação de certificados X.509.
- Validação de certificados SET.
- Lista de Revogação de Certificados (CRL).
- Processamento do BrandCRLIdentifier (BCI).

Na prática, é assumido que o processo de validação parará em um nível que previamente tenha sido validado. Todo *software* SET validará datas de certificado como parte do processo de validação da cadeia de certificados. O *software* do SET proverá um mecanismo de advertência para certificados em expiração, para prevenir a tentativa de seu uso após o vencimento. A entidade receptora executará o procedimento de validação da cadeia de certificados.

Passo	Procedimento de Validação da Cadeia de Certificados
1	Validação de cada certificado na cadeia de acordo com as regras especificadas na Seção 12.4.3 do padrão X.509, usando os passos de validação de cadeia do SET como descritos anteriormente.
2	Verificação se as extensões de certificado *KeyUsage, CertificatePolicies*, *PrivateKeyUsage*, e *AuthorityKeyIdentifier* estão sendo usadas conforme o X.509.
3	Se um BCI novo foi recebido: a. Validação de sua assinatura usando o certificado assinado da CRL da CA de Marca. b. Verificação se o *BrandName* no BCI coincide com aquele na cadeia de certificado sendo validada. c. Verificação se a data do *NotAfter* é menor que a data presente. d. Conferência do *SequenceNum*. Se é maior que o *SequenceNum* no BCI da memória , armazenagem do BCI e verificação de que todas as CRLs contidas no BCI estão guardadas na CRL da memória. Armazenagem de quaisquer CRLs que estão no BCI mas ainda não estão na memória .
4	Para cada CRL nova que foi recebida, execução da validação da CRL.
5	Conferência de cada certificado contra a CRL da CA que assinou.

Impressões Digitais

As impressões digitais são geradas como se segue:

As impressões digitais são computadas executando o *hash* SHA-1 das seguintes codificações DER das estruturas ASN.1:

- *UnsignedCertificate.*
- *UnsignedCertificateRevocationList.*
- *UnsignedBrandCRLIdentifier.*

O *hash* é computado sobre do valor-de-comprimento-de-etiqueta (*tag-length-value*) da estrutura codificada DER. A Impressão digital é o mesmo *hash* que é usado para assinar ou verificar um certificado ou CRL ou BCI.

As impressões digitais são enviadas por uma entidade em uma mensagem de pedido do SET e sempre podem ser ignoradas pelo receptor correspondente. Da entidade que envia não é exigido enviar todas as impressões digitais de todos os certificados, CRLs e *BrandCRLIdentifiers* que existem atualmente em sua memória, mas só aquelas que são pertinentes a um par de mensagem de pedido/resposta particular. Por exemplo, o *software* do comerciante não precisará enviar as impressões digitais para os outros portadores de cartão de crédito ou para outras marcas. As impressões digitais podem ser listadas em qualquer ordem. O quadro seguinte descreve o procedimento de envio das impressões digitais.

Passo	Procedimento de envio de impressões digitais
1	Inicialização do buffer para armazenamento das impressões digitais.
2	Anexação da impressão digital (*hash*) correspondente: • Para cada certificado que exista na memória do sistema que envia que é pertinente ao processamento da mensagem de resposta e para validação da cadeia de certificado, anexação da impressão digital (*hash*) correspondendo a este certificado. • Para cada CRL que exista na memória do sistema que envia que é pertinente ao processamento da mensagem de resposta e para validação da cadeia de certificado, anexação da impressão digital (*hash*) correspondendo a esta CRL. • Para cada BrandCRLIdentifier que exista na memória do sistema que envia que é pertinente ao processamento da mensagem de resposta e para validação da cadeia de certificado, anexação da impressão digital (*hash*) correspondendo a este BrandCRLIdentifier.

O receptor irá se assegurar que o emissor da mensagem possui todos os certificados, CRLs, e o *BrandCRLIdentifier* necessários para completar o processamento da mensagem. O receptor pode escolher ignorar as impressões digitais e enviar esta informação ao solicitante. O quadro seguinte descreve o procedimento de recebimento das impressões digitais.

Passo	**Procedimento de recebimento de impressões digitais**
1	Inicialização do *buffer* para armazenamento das impressões digitais.
2	Para cada: • Certificado que é pertinente para o processamento da mensagem de resposta ou para validação da cadeia de certificado, conferência se a impressão digital do certificado (*hash*) coincide com uma das impressões digitais recebidas na mensagem de pedido. Se a impressão digital coincidir, o certificado existe na memória do sistema remoto e não há necessidade de ser enviado com a mensagem de resposta. Se a impressão digital não casa ou a lista está vazia, então deve ser feita a inclusão do certificado na mensagem de resposta. • CRL que é pertinente para o processamento da mensagem de resposta ou para validação da cadeia de certificado, conferência se a impressão digital da CRL (*hash*) coincide com uma das impressões digitais recebidas na mensagem de pedido. Se a impressão digital coincidir, a CRL existe na memória do sistema remoto e não há necessidade que a CRL seja enviada com a mensagem de resposta. Se a impressão digital não casa ou a lista está vazia, então deve ser incluída a CRL na mensagem de resposta. • BrandCRLIdentifier que é pertinente para o processamento da mensagem de resposta ou para validação da cadeia de certificado, conferência se a impressão digital da identificador de CRL (*hash*) coincide com uma das impressões digitais recebidas na mensagem de pedido. Se a impressão digital coincidir, o identificador da CRL existe na memória do sistema remoto e não há necessidade que seja enviado com a mensagem de resposta. Se a impressão digital não coincide ou a lista está vazia, então inclua o identificador de CRL na mensagem de resposta.
3	Retorno do resultado do passo 2 com lista de certificados, CRLs e BrandCRLIdentifiers a serem transferidos com mensagem de resposta.

6.4. Processamento Criptográfico das Mensagens do SET

Os operadores de assinatura, *hash*, cifragem e encapsulamento vistos no capítulo anterior, estão listados a seguir. Agora veremos como eles são aplicados pelo SET para realizarem a criptografia de suas mensagens.

- Assinatura (S)
- Assinatura Somente (SO)
- Hash (H)
- DigestedData (DD)
- Ligação (L)
- Hash-Comutado (HMAC)
- Cifragem Assimétrica (E)
- Cifragem Assimétrica com Integridade (EH)
- Cifragem Assimétrica Extra (EX)
- Cifragem Assimétrica Extra com Integridade (EXH)
- Cifragem Simétrica (EK)
- Encapsulamento Simples com Assinatura (Enc)
- Encapsulamento Simples com Assinatura e Chave Fornecida (EncK)
- Encapsulamento Extra com Assinatura (EncX)
- Encapsulamento Simples com Assinatura e Bagagem (EncB)
- Encapsulamento Extra com Assinatura e Bagagem (EncBX)
- Optimal Asymmetric Encryption Padding (OAEP)

Com as convenções e notações vistas, e para entendimento da forma de como está explicitada a aplicação dos operadores de cifragem utilizados pelo SET, vamos detalhar como exemplo, em nível macro, as estruturas de dados que compõem o par de mensagens do SET de pedido **AuthReq** e resposta **AuthRes** do par de mensagens Pedido / Resposta de Autorização de Compra. Este par de mensagem foi totalmente detalhado no capítulo 4. Por ora revemos as macro-estruturas de dados que a compõe.

O formato da porção do pedido do par, **AuthReq,** é:

EncB (M, P, AuthReqData, PI)

O formato da porção de resposta do par, **AuthRes** é :

< EncB (P, M, AuthResData, AuthResBaggage), EncBX (P,M, AuthResData, AuthResBaggage, PANToken >

O operador **EncB** da mensagem **AuthReq** diz-nos que a mensagem é codificada usando Encapsulamento Simples com Assinatura e Bagagem, e contém quatro componentes, o **M** indica que a mensagem é do Comerciante **(*Merchant*)** e o **P** que o receptor é o Portal de pagamento **(*Payment Gateway*)**. Os outros dois elementos são os componentes de dados: **AuthReqData**, (*Authorization Request Data* - Dados de Pedido de Autorização) e **PI** (*Payment Instructions* - Instruções de Pagamento). O **AuthReqData** é uma tupla consistindo do **AuthReqItem**, um **MThumbs** opcional, um componente **CaptureNow** e o componente opcional **SaleDetail**.

A mensagem de resposta de autorização (**AuthRes**) é uma seleção de uma de duas formas: Encapsulamento Simples com Assinatura e Bagagem (**EncB**) ou Encapsulamento Extra com Assinatura e Bagagem (**EncBX**), onde a bagagem extra é indicada pelo componente **PANToken** (*Token do Payment Account Number* - PAN). Ambas as formas contem **AuthResData** e **AuthResBagage** . Ela é enviada da entidade ***Payment Gateway* (P)** para a entidade ***Merchant* (M)**.

Assim, para o restante das mensagens de gerenciamento de certificados e do sistema de pagamentos do SET, as tabelas abaixo exibem as estruturas de dados que as compõe e os operadores de cifragem relacionados a cada mensagem.

Pares de Mensagens de Gerenciamento de Certificados	**Mensagem**	**Operador Cifragem**	**Estruturas de dados da Mensagem**
1-Inicialização Certificados Portador de Cartão	CardCInitReq		{RRRID, LID-EE, Chall-EE,BrandID, [Thumbs]}
	CardCInitRes	S (s,t)	S (CA, CardCInitResTBS)
2-Inicialização Certificados Comerciante-Adquirente	Me-AqCInitReq		{RRRPID, LID-EE, Chall-EE, RequestType, IDDDataBrandID, Language, [Thumbs]}
	Me-AqCInitRes	S (s,t)	S (CA,Me-AqCInitResTBS)
3-Formulário Registro Portador de Cartão	RegFormReq	EXH (r,t,p)	EXH (C, RegFormReqData, PANOnly)
	RegFormRes	S (s,t)	S (CA, RegFormResTBS)
4-Pedido e Resposta Certificados	CertReq	EncX (s,r,t,p)	EncX (EE, CA, CertReqData, AcctInfo)
		Enc (s,r,t)	Enc (EE, CA, CertReqData)
	CertRes	S (s,t)	S (CA, CertResData)
		EncK (k,s,t)	EncK (CABackKeyData, CA, CertResData)

5- Pedido Resposta Pesquisa Certificado	PCertReq	S (s,t)	S (M, PcertReqData)
	PCertRes	S (s,t)	S(P, PcertResTBS)
6- Investigação de Certificados	CertInqReq	S (s,t)	S (EE, CertInqReqTBS)
	CertInqRes	S (s,t)	S (CA, CertResData)
		Enc k (k,s,t)	EncK (CABackKeyData, CA, CertResData)

Tabela 6.1: Operadores de cifragem de mensagens de gerenciamento de certificados.

Pares de Mensagens do Sistema de Pagamentos	**Mensagem**	**Operador Cifragem**	**Estruturas de dados da Mensagem**
1-Inicialização de Pagamento	PInitReq		{RRPID, language, LID-C, [LID-M],Chall-C, BrandID, BIN, [Thumbs], [PIRqExtensions]}
	PInitRes	S (s,t)	S (M,PinitResData)
2- Pedido e Resposta de Compra	PReq		PreqDualSigned
			PreqUnsigned
	PRes	S (s,t)	S (M, PresData)
3-Investigação de Transação	InqReq	S (s,t)	InqReqSigned = S (C, InqReqData)
	InqRes	S (s,t)	S (M , PresData)
4- Pedido e Resposta de Autorização de Compra	AuthReq	EncB (s,r,t,b)	EncB (M, P, AuthReqData,PI)
	AuthRes	EncB (s,r,t,b)	EncB (P, M , AuthResData, AuthResBaggage)
		EncBX (s,r,t,b,p)	EncBX (P, M, AuthResData, AuthResBaggage, PANtoken)
5- Pedido e Resposta de Captura	CapReq	EncB (s,r,t,b)	EncB (M, P, CapReqData, CapTokenSeq)
		EncBX (s,r,t,b,p)	EncBX (M, P, CapReqData, CapTokenSeq, PanToken)
	CapRes	Enc (s,r,t,)	Enc (P, M, CapResData)
6- Pedido e Resposta Reversão Autorização	AuthRevReq	EncB (s,r,t,b)	EncB (M, P, AuthRevReqData, AuthRevReqBaggage)
	AuthRevRes	EncB (s,r,t,b)	EncB (P, M, AuthRevResData, AuthRevResBaggage)
		Enc (s,r,t)	Enc (P, M, AuthRevResData)

7- Pedido e Resposta Reversão Captura	CapRevReq	EncB (s,r,t,b)	EncB (M, P, CapRevData, CapTokenSeq)
		EncBX (s,r,t,b,p)	EncBX (M, P, CapRevData, CapTokenSeq, PABToken)
	CapRevRes	Enc (s,r,t)	Enc (P, M, CapRevResData)
8- Pedido e Resposta de Crédito	CredReq	EncB (s,r,t,b)	EncB (M, P, CredReqData, CapTokenSeq)
		EncBX (s,r,t,b,p)	EncBX (M, P, CredReqData, CapTokenSeq, PANToken)
	CredRes	Enc (s,r,t)	Enc (P, M, CredResData)
9- Pedido e Resposta de Reversão Crédito	CredRevReq	EncB (s,r,t,b)	EncB (M, P, CredRevReqData, CapTokenSeq)
		EncBX (s,r,t,b,p)	EncBX (M, P, CredRevReqData, CapTokenSeq, PANToken)
	CredRevRes	Enc (s,r,t)	Enc (P, M, CredRevResData)
10- Pedido e Resposta de Admnistração Lote	BatchAdminReq	Enc (r,s,t)	Enc (M, P, BatchAdminReqData)
	BatchAdminRes	Enc (r,s,t)	Enc (M, P, BatchAdminResData)

Tabela 6.2: Operadores de cifragem de mensagens do sistema de pagamentos

6.5. Operadores de Cifragem Aplicados às Mensagens do SET

6.5.1. Assinatura – Operador $S(s,t)$

O operador $S(s,t)$ é a assinatura da tupla t com a chave privada s do emissor incluindo o texto claro t **.** O SET utiliza esse operador para as seguintes mensagens: CardCInitRes, Me-AqCInitRes, RegFormRes, CertRes, CertInqReq, CertInqRes, PinitRes, PRes, InqReq, PcertReq, PcertRes.

Como exemplo detalhamos o conteúdo do pacote *SignedData* do PKCS # 7 para a mensagem de Resposta (CardCInitRes) do Par de Mensagens de Pedido e Resposta Certificados de Portador de Cartão: S (CA, CardCInitResTBS).

Composição do *SignedData* do PKCS #7 na mensagem S (CA, CardCInitResTBS)				
Campos	**Sub-Campos Nível 1**	**Sub-campos Nível 2**	**Conteúdos dos campos**	**Mensagem assinada**
Versão			2	2
Identificador de Algoritmo de Resumo			SHA-1	SHA-1
Informação de Conteúdo	Tipo de Conteúdo		Identificador do Conteúdo	**CardCInitResTBS**
	Conteúdo		Dado a ser assinado (codificado DER)	Estrutura de dados **CardCInitResTBS**
Certificados			Certificados necessários para verificar assinatura	Certificado da CA
CRLs			CRLs	Uma ou várias
Informação de Assinantes	Versão		2	2
	Emissor e Número de Série (Uma ou mais Instâncias)	Nome Emissor Certificados		Nome da CA
		Número de Série de Certificado		Número de série do Certificado da CA
	Algoritmo de Resumo		SHA-1	SHA-1
	Atributos Autenticados		Identificador do conteúdo	**CardCInitResTBS**
			Resumo de Mensagem	**SHA-1(CardCInitResTBS)**
	Algoritmo de Cifragem de Resumo		Cifragem RSA	*RSAEncyption*
	Resumo Cifrado			**Resumo SHA-1 da estrutura CardCInitResTBS cifrada RSA com a chave privada do emissor CA**

6.5.2. Somente Assinatura – Operador $SO(s,t)$

O operador $SO(s,t)$ é a assinatura da tupla t com a chave privada **s** do emissor, mas não inclui o texto claro t. A diferença para o operador $S(s,t)$**,** que inclui o texto claro, é que os dados codificados DER não são incluídos no pacote *SignedData* do PKCS # 7.

Como exemplo da aplicação desse operdor examinemos a estrutura de dados de Instrução de Pagamento (**PI – *Payment Instruction***) que é a estrutura de dados mais central e sensível do SET. Ela é usada para passar os dados necessários para autorizar um pagamento de cartão do Portador de cartão para o Portal de Pagamento, que irá usar os dados para iniciar uma transação de cartão de pagamento através da rede financeira tradicional. O dado é cifrado pelo Portador de cartão e enviado via Comerciante, de forma que o dado seja omitido do Comerciante, a não ser que o Adquirente os retorne ao Comerciante. A estrutura **PIDualSigned**, a opção de assinatura dual da Instrução de Pagamento, é constituída das estruturas de dados **{PISignature, EX(P, PI-OILink, PANData)}**. A estrutura de dados **PISignature**, por sua vez, é constituída das estrutura **SO(C, PI-TBS)**. A representação da estrutura **SO (C, PI-TBS)** nos diz que a estrutura de dado **PI-TBS** é assinada com o operador de Somente Assinatura, utilizando a chave privada do assinante, ou seja, Portador de cartão (C – *Cardholder*). A composição do *SignedData* do PKCS #7 na mensagem **SO (C, PI-TBS)** seria semelhante a do exemplo anterior, sendo que o conteúdo a ser assinado seria a estrutura de dados **PI-TBS** e o campo contendo o dado a ser assinado (codificado DER) fica vazio.

6.5.3. *Hash* – Operador H(t)

O operador $H(t)$ é o *hash* **SHA-1** de comprimento 160 bits da tupla t. Este operador corresponde ao tipo parametrizado H{} da ASN.1. Entretanto este operador nunca é diretamente utilizado nas estruturas de dados das mensagens do SET, ele é apenas aplicado no processamento interno do **OAEP**. Para exemplo de aplicação desse operador veja o Fluxo de Processamento de Cifragem Extra **OAEP** exposto mais adiante neste capítulo.

6.5.4. Resumo de Dados – Operador DD(t)

O operador **DD(t)** ou *DigestedData* corresponde ao *hash* **SHA-1** de 160 bits de comprimento da tupla **t** envolvido num pacote *DigestedData* do PKCS # 7. O SET usa o tipo parametrizado **DD{}** para especificar resumos desanexados, no qual o conteúdo que é resumido não é incluido no componente conteúdo de Informação de Conteúdo. No ***DigestedData*** tipo **DD{}** o campo conteúdo reservado ao texto a ser resumido (codificado DER) fica vazio. O receptor então não disporá da mensagem no pacote, tendo que obtê-la em outro lugar para proceder a verificação.

Para exemplicar a aplicação desse operador, continuemos decompondo a estrutura de dados **SO(C, PI-TBS)**, vista no exemplo do operador de Somente Assinatura. A estrutura de dados **PI-TBS** é **{HPIData, HOIData}**, onde por sua vez, a estrutura de dados **HPIData** é o resumo de dados **DD(PIData)**.

Detalhamos em seguida o conteúdo do pacote ***DigestedData*** do PKCS # 7 para a estrutura de dados **DD(PIData)**.

<table>
<tr><th colspan="4">Composição do DigestedData do PKCS #7 tipo DD{} na estrutura de dados DD(PIData)</th></tr>
<tr><th>Campos</th><th>Sub-Campos
Nível 1</th><th>Conteúdos
dos campos</th><th>Estrutura de dados</th></tr>
<tr><td>Versão</td><td></td><td>0</td><td>0</td></tr>
<tr><td>Algoritmo de resumo</td><td></td><td>SHA-1
e parâmetros
associados aos dados
a serem resumidos</td><td>SHA-1,
Endereço do dado
PIData,
Comprimento do
dado PIData</td></tr>
<tr><td rowspan="2">Informação de
Conteúdo</td><td>Tipo de Conteúdo</td><td>Identificador do
Conteúdo</td><td>PIData</td></tr>
<tr><td>Conteúdo</td><td>Dado a ser resumido
(codificado DER)</td><td>Vazio</td></tr>
<tr><td>Resumo</td><td></td><td></td><td>Resumo SHA-1 de 160 bits da estrutura de dados PIData, do endereço dos dados PIData, do comprimento do dado PIData</td></tr>
</table>

6.5.5. Ligação – Operador $L(t1,t2)$

O operador de ligação $L(t1,t2)$ corresponde à uma sequência resultante da cocatenação da tupla $t1$ e um ***DigestedData*** parametrizado – **DD{}** da tupla $t2$. É uma estrutura com dois campos, sendo o primeiro campo preenchido com a tupla $t1$e o segundo campo preenchido com o resultado de $DD(t2)$. O operador de ligação não é simétrico, ou seja, não une $t2$ a $t1$.

Como exemplo da aplicação temos a estrutura de dados **PI-OILink** que compõe a estrutura de dados **PIDualSigned**, que é **{PISignature, EX(P, PI-OILink, PANData)}**, uma opção de assinatura dual da Instrução de Pagamento. A estrutura **PI-OILink** é constituída por **L(PIHead, OIData)**, ou seja ligação da estrutura de dados **PIHead** (identificação de Instrução de Pagamento) à estrutura de dados **OIData** (dados de Instrução de Pedidos).

6.5.6. *Hash* com Chave Fornecida – Operador HMAC(t,k)

O operador de *hash* com chave $HMAC(t,k)$ corresponde ao *hash* $HMAC-SHA-1$ de 160 bits da tupla t, usando o segredo k. A operação $HMAC(t,k)=H((k\oplus opad)\,||\,H(k\oplus ipad\,||\,t))$, onde k é a chave secreta fornecida, **ipad** é o byte 0x36 repetido 64 vezes, **opad** é o byte 0x5C repetido e **H()** é o operador de *hash* **SHA-1** de 160 bits visto anteriormente.

A evidência que um Portador de cartão tenha participado de uma transação é dada pela assinatura digital do Portador de cartão e certificado. Como exemplo da aplicação deste operador, verificaremos como essa evidência pode ser obtida de uma outra forma.

A estrutura de dados componente da Instrução de Pagamento, **PIHead** é constituída como **{TransIDs, Inputs, MerchantID, [InstallRecurData], TransStain, SWIdent, [AcqBackKeyData], [PIExtensions]}**. A estrutura de dados **TransStain** é **HMAC(XID, CardSecret)**. Na transação de registro de certificado, o Portador de cartão gera **XID** (identificador único de cada da transação) e **CardSecret** é o valor secreto que o Portador de cartão envia a CA numa transação. A partir daí, o Portador de cartão e a CA conhecem o valor **CardSecret**. O Portador de cartão executa o processamento **HMAC(XID, CardSecret)** e o resultado deste processamento é enviado como parte da Instrução de Pagamento pelo Portador

de cartão, como visto acima. Em função de ambos, CA e Portador de cartão conhecerem o valor **CardSecret**, ou seja **k**, o **HMAC(t,k)** = **HMAC(XID, CardSecret)** é um *hash* que é gerado pelo Portador de cartão e que somente a CA pode verificar. Isto permite que o Emissor possa efetivamente comprovar a participação do Portador de cartão na transação.

6.5.7. Cifragem Assimétrica – Operador $E(r,t)$.

O operador $E(r,t)$ executa a cifragem simétrica da tupla t com a chave k do *DES* recentemente gerada pelo emissor e a cifragem RSA da chave k processada OAEP (cifragem RSA OAEP) com a chave pública do receptor r. A cifragem assimétrica corresponde ao *EnvelopedData* do PKCS # 7 da tupla t cifrada para a entidade r. No *EnvelopedData* do PKCS # 7, o campo de chave cifrada conterá $RSAOAEP(k)$ e o campo de dados cifrados conterá $DESCBC(t)$.

O exemplo de aplicação deste operador será visto mais adiante na composição envelopada resultante da operação do Encapsulamento Simples com Assinatura – $Enc(s,r,t)$, onde o resultado da aplicação do operador $E(r,t)$ aparece no campo de chave cifrada do envelope.

6.5.8. Cifragem Assimétrica com Integridade – Operador $EH(r,t)$

O operador $EH(r,t)$ executa a cifragem simétrica DES CBC da tupla t e cifragem RSA OAEP da chave k do *DES* e *hash* da tupla t com chave pública r do receptor. Este operador é similar ao operador E acima, exceto que o campo de dados cifrados do *EnvelopedData* do PKCS # 7 conterá um *hash* da tupla t como garantia de integridade. No *EnvelopedData* do PKCS # 7, o campo de chave cifrada conterá $RSA\ OAEP(\{k, SHA-1(t)\})$ e o campo de dados cifrados conterá $DESCBC(t)$.

6.5.9. Cifragem Assimétrica Extra – Operador $EX\,(r.t, p)$

O operador $EX\,(r.t, p)$ executa a a cifragem $DES\,CBC$ da tupla t ligada ao parâmetro p, ou seja, $L(t, p)$, onde p é um *nonce* recente de 20 bytes, para evitar-se ataques de dicionário. Este *nonce* é denominado **EXNonce.** Após, é executada a cifragem RSA OAEP da chave k do DES concatenada com o parâmetro p. No *EnvelopedData* do PKCS # 7, o campo de chave cifrada conterá $RSA\,OAEP(\{k, p\})$ e o campo de dados cifrados conterá $DES\,CBC(\{t, SHA-1(p)\})$.

O exemplo da aplicação deste operador se encontra no exemplo do operador de Encapsulamento Extra com Assinatura – $EncX\,(s, r, t, p)$ à mensagem $EncX\,(EE, CA, Cert\,\mathrm{Re}\,qData, AcctInfo)$, visto adiante.

6.5.10. Cifragem Assimétrica Extra com Integridade – Operador $EXH\,(r, t, p)$

O operador $EXH\,(r, t, p)$ executa a cifragem $DES\,CBC$ da tupla t ligada ao parâmetro p, isto é, $L(t, p)$, concatenado com um *hash* de integridade da tupla t. O parâmetro p é um *nonce* de 20 bytes denominado **PANOnly**. Após isto, é executada a cifragem RSA OAEP da chave k do DES concatenada com o parâmetro p e o *hash* da tupla t. No *EnvelopedData* do PKCS # 7, o campo de chave cifrada conterá $RSA\,OAEP\,(\{k, p, SHA-1(t)\})$ e o campo de dados cifrados conterá $DES\,CBC\,(\{t, SHA-1(p),\quad SHA-1(t)\})$.

Composição do *EnvelopedData* do PKCS #7 na mensagem EXH (C, RegFormReqData, PANOnly) $EHX(r,t,p) \rightarrow RSA\ OAEP(\{k, p, SHA-1(t)\}) \mid\mid DES\ CBC(\{t, SHA-1, (p), SHA-1(t)\})$				
Campos	**Sub-Campos Nível 1**	**Sub-campos Nível 2**	**Conteúdos dos campos**	**Mensagem cifrada**
Versão			1	1
Informação de Receptor	Versão		0	0
	Emissor e Número de Série (identifica o certificado do receptor pretendido, o certificado não está incluido)	Nome do Emissor		Nome da Comerciante
		Número de Série de Certificados		Número de série do Certificado do Comerciante
	Algoritmo de cifragem de chave		RSA OAEP Encryption SET	RSA OAEP *Encryption* SET
	Chave cifrada			**Chave k do DES | | PANOnly | | SHA-1(RegFormReqData)**
Informação de conteúdo cifrado	Tipo de conteúdo			*EncryptedData*
	Alagoritmo de cifragem de conteúdo		DES CBC	DES CBC
	Conteúdo cifrado			**RegFormReqData | | SHA-1 (PANOnly) | | SHA-1 (RegFormReqData)**

Como exemplo de composição de *envelopedData* do PKCS # 7, vimos no quadro acima a aplicação do operador de cifragem *EXH* na mensagem **EXH (C, RegFormReqData, PANOnly)**, que é a porção de pedido (**RegFormReq**) do Par de Mensagens Pedido Resposta de Formulário de Registro de Portador de Cartão

6.5.11. Cifragem Simétrica – Operador $EK(h,t)$

O operador $EK(h,t)$ executa a cifragem simétrica do texto claro da tupla t com uma chave secreta k fornecida pelo receptor em uma transação prévia. Os algoritmos *DES* ou CDMF podem ser usados.

O exemplo de aplicação deste operador será visto mais adiante na composição do *EncryptedData* do PKCS #7 resultante da operação do Encapsulamento Simples com Assinatura e Chave Fornecida – $EncK(k,s,t)$ na mensagem $EncK(CABackKeyData, CA, Cert\,\mathrm{Re}\,sData)$. Esta mensagem é a porção de Resposta (**CertRes**) do Par de Mensagens Pedido Resposta de Certificados. O resultado da aplicação do operador $EK(k,t)$ aparece no campo de conteúdo cifrado.

6.5.12. Encapsulamento Simples com Assinatura – Operador $Enc(s,r,t)$

O operador $Enc(r,s,t)$ executa a assinatura das mensagens com o operador $S(s,t)$ e então cifra com o operador $E(r,t)$. Ele corresponde a uma instância do *SignedData* encapsulado no *EnvelopedData* do PKCS # 7, ou seja, $E(r,(S(s,t))$. No *EnvelopedData* do PKCS # 7, o campo de chave cifrada conterá $RSAOAEP(k)$ e o campo de dados cifrados conterá $DES\,CBC(S(s,t))$.

As mensagens utilizadoras desse tipo de operador do SET são as mensagens: CertReq, AuthRevReq, CapRevRes, CredRes, BatchAdminReq, BatchAdminRes. No quadro a seguir, é detalhada a mensagem $Enc(EE, CA, Cert\,\mathrm{Re}\,qData)$, que é a porção do Pedido (**CertReq**) do Par de Mensagens Pedido e Resposta Certificados, numa composição *EnvelopedData*.

Composição do *EnvelopedData* do PKCS #7 na mensagem Enc (EE, CA, CertReqData) $Enc(r,s,t) \rightarrow RSAOAEP(k) \,||\, DES\,CBC\big(S(s,t)\big)$				
Campos	**Subcampos Nível 1**	**Subcampos Nível 2**	**Conteúdos dos campos**	**Mensagem assinada e cifrada**
Informação de Receptor	Chave cifrada			cifragem RSA OAEP da Chave *k* do *DES* da Entidade Final (EE) com chave pública da CA
Informação de conteúdo cifrado	Conteúdo cifrado			cifragem DES CBC da estutura assinada **S (EE, CertReqData)** com chave secreta *k* da Entidade Final (EE)

Como anteriormente já haviam sido exemplificados os outros campos componentes do *envelopedData*, a guisa de objetividade, no quadro deste e dos próximos exemplos, estarão exibidas a composição somente com os campos Chave cifrada e Conteúdo cifrado.

6.5.13. Encapsulamento Simples com Assinatura e Chave Fornecida – Operador $EncK(k,s,t)$

O operador $EncK(k,s,t)$ executa a assinatura das mensagens com o operador $S(s,t)$ e então cifra com o operador $E(K,t)$. Ele corresponde a uma instância do *SignedData* cifrada no *EncryptedData* do PKCS # 7, ou seja, $EK(k,S(s,t))$. No *EncryptedeData* do PKCS # 7, o campo de dados cifrados conterá $DES\,CBC\,(S(s,t))$.

As mensagens utilizadoras desse tipo de operador do SET são as mensagens: $Cert\,\mathrm{Re}\,s, CertInq\,\mathrm{Re}\,s$. O quadro abaixo detalha a mensagem $EncK(CABackKeyData, CA, Cert\,\mathrm{Re}\,sData)$, que é a porção de Resposta (**CertRes**) do Par de Mensagens Pedido Resposta de Certificados, numa composição *EncryptedData*.

Composição do *EncryptedData* do PKCS #7 na mensagem EncK (CABackKeyData, CA, CertResData) $EncK(k,s,t) \rightarrow DES\,CBC\left(k, S(s,t)\right)$			
Campos	**Subcampos Nível 1**	**Conteúdos dos campos**	**Mensagem assinada e cifrada com chave k fornecida**
Versão		0	0
Informação de Conteúdo	Tipo de Conteúdo	Identificador do Conteúdo	*EncryptedData*
	Algoritmo de cifragem de conteúdo	DES CBC	DES CBC
	Conteúdo Cifrado		**cifragem DES CBC da estrutura assinada pela entidade CA S (CA, CertResData), com chave secreta CABackKeyData fornecida previamente pelo receptor**

6.5.14. Encapsulamento Extra com Assinatura – Operador $EncX(s,r,t,p)$.

O operador $EncX(s,r,t,p)$ executa a assinatura da tupla t, concatenada com o dado extra a ser cifrado, p, com o operador $SO(s,t)$. Após, o resultado da assinatura é concatenado com a tupla t, e esta concatenação é cifrada usando o operador de Cifragem Assimétrica Extra – $EX(r,t,p)$. No *EnvelopedData* do PKCS # 7, o campo de chave cifrada conterá $RSA\,OAEP\,(\{k,p\})$ e o campo de dados cifrados conterá. $DES\,CBC\left(\{t, SO(s,\{t,p\})\}\right)$.

A mensagem utilizadora desse tipo de operador do SET é: CertReq. O quadro que se segue detalha a mensagem $EncX(EE,CA,Cert\,\mathrm{Re}\,qData,AcctInfo)$, que é a porção de Pedido (**CertReq**) do Par de Mensagens Pedido Resposta de Certificados, numa composição *EnvelopedData*.

Composição do *EnvelopedData* do PKCS #7 na mensagem EncX (EE, CA, CertReqData, AcctInfo) $EncX(s,r,t,p) \rightarrow RSAOAEP(\{k,p\}) \,||\, DESCBC(\{t,SO(s,\{t,p\})\})$				
Campos	**Subcampos Nível 1**	**Subcampos Nível 2**	**Conteúdos dos campos**	**Mensagem assinada e cifrada com dado extra**
Informação de Receptor	Chave cifrada			**cifragem RSA OAEP da Chave k do DES da Entidade Final (EE) | | AcctInfo, com chave pública da CA**
Informação de conteúdo cifrado	Conteúdo cifrado			**cifragem DES CBC de (CertReqData| | SO(EE,(CertReqData | |AcctInfo)), com chave secreta da EE**

6.5.15. Encapsulamento Simples com Assinatura e Bagagem - Operador $EncB(s,r,t,b)$

O operador $EncB(s,r,t,p)$ executa a assinatura da tupla t ligada com o *hash* da bagagem externa b com o operador $S(s,t)$. Após, o resultado da assinatura é concatenado com a tupla t, e esta concatenação é cifrada usando o operador de Cifragem Assimétrica – $E(r,t)$. No *EnvelopedData* do PKCS # 7, o campo de chave cifrada conterá $RSAOAEP(k)$ e o campo de dados cifrados conterá $DESCBC(\{S(s,\{t,SHA-1(b)\}),p\})$.

As mensagens utilizadoras desse tipo de operador do SET são as mensagens: **CertReq**, **AuthReq**, **AuthRes**, **CapReq**, **AuthRevReq**, **AuthRevRes**, **CapRevReq**, **CredReq**, **CredRevReq**. O quadro abaixo exibeo detalhamento da mensagem **EncB (M, P, AuthReqData, PI)**, que é a porção do Pedido (**AuthReq**) do Par de Mensagens Pedido e Resposta Autorização de Venda, numa composição *EnvelopedData*.

Composição do *EnvelopedData* do PKCS #7 na mensagem EncB (M, P, AuthReqData, PI) $EncB(s,r,t,p) \rightarrow RSA\ OAEP(k) \mid\mid DES\ CBC(\{S(s,\{t,SHA-1(b)\}),p\})$				
Campos	**Subcampos Nível 1**	**Subcampos Nível 2**	**Conteúdos dos campos**	**Mensagem assinada e cifrada com bagagem externa**
Informação de Receptor	Chave cifrada			**cifragem RSA OAEP da Chave *k* do *DES* do Comerciante (M) , com chave pública do Portador de cartão (P)**
Informação de conteúdo cifrado	Conteúdo cifrado			**cifragem DES CBC de (S(M,(AuthReqData | | SHA-1(PI) | | PI) ,com chave secreta de M**

6.5.16. Encapsulamento Extra com Assinatura e Bagagem - Operador $EncBX(s,r,b,p)$.

O operador $EncBX(s,r,t,b,p)$ executa a assinatura da tupla t ligada à bagagem externa b e concatenada com p com o operador $SO(s,t)$. Após, o resultado da assinatura é concatenado com a tupla t ligada com a bagagem externa b e concatenado com p, e esta concatenação é cifrada usando o operador de Cifragem Assimétrica – $EX(r,t,p)$. No *EnvelopedData* do PKCS # 7, o campo de chave cifrada conterá $RSAOAEP(\{k,p\})$ e o campo de dados cifrados conterá $DES\,CBC(\{\{SO(s,\{t,SHA-1(b),p\}),\{t,SHA-1(b)\}\})$.

As mensagens utilizadoras desse tipo de operador do SET são as mensagens: **AuthRes**, **CapReq**, **CapRevReq**, **CredReq**, **CredRevReq**. O quadro abaixo detalha a mensagem **EncBX (M, P, CredReqData, CapTokenSeq, PANToken)**, que é a porção do Pedido (**CredReq**) do Par de Mensagens Pedido Resposta de Crédito, numa composição *EnvelopedData*.

Composição do *EnvelopedData* do PKCS #7 na mensagem EncBX (M, P, CredReqData, CapTokenSeq, PANToken) $EncBX(s,r,t,b,p) \rightarrow RSA\ OAEP(\{k,p\}) \,||\, DES\ CBC(\{\{SO(s,\{t,SHA-1(b),p\}),\{t,SHA-1(b)\}\}),$				
Campos	**Sub-Campos Nível 1**	**Sub-campos Nível 2**	**Conteúdos dos campos**	**Mensagem assinada cifrada com bagagem externa e dado extra**
	Chave cifrada			**cifragem RSA OAEP da Chave *k* do *DES* do Comerciante (M) || PANToken, com chave pública do Portador de cartão (P)**
	Conteúdo cifrado			**cifragem des CBC de SO(M,(CredReqData || SHA-1(CapTokenSeq) || PANToken) || CredReqData || SHA-1(CapTokenSeq), com chave secreta de M**

6.5.17. *Optimal Asymmetric Encryption Padding* – Operador OAEP

Ao longo deste capítulo, foram vistos exemplos da aplicação nas mensagens do SET do processamento OAEP antes da cifragem assimétrica da chave secreta e dados extras adicionais na cifragem. O processamento OAEP se encontra expresso graficamente na figura que se segue.

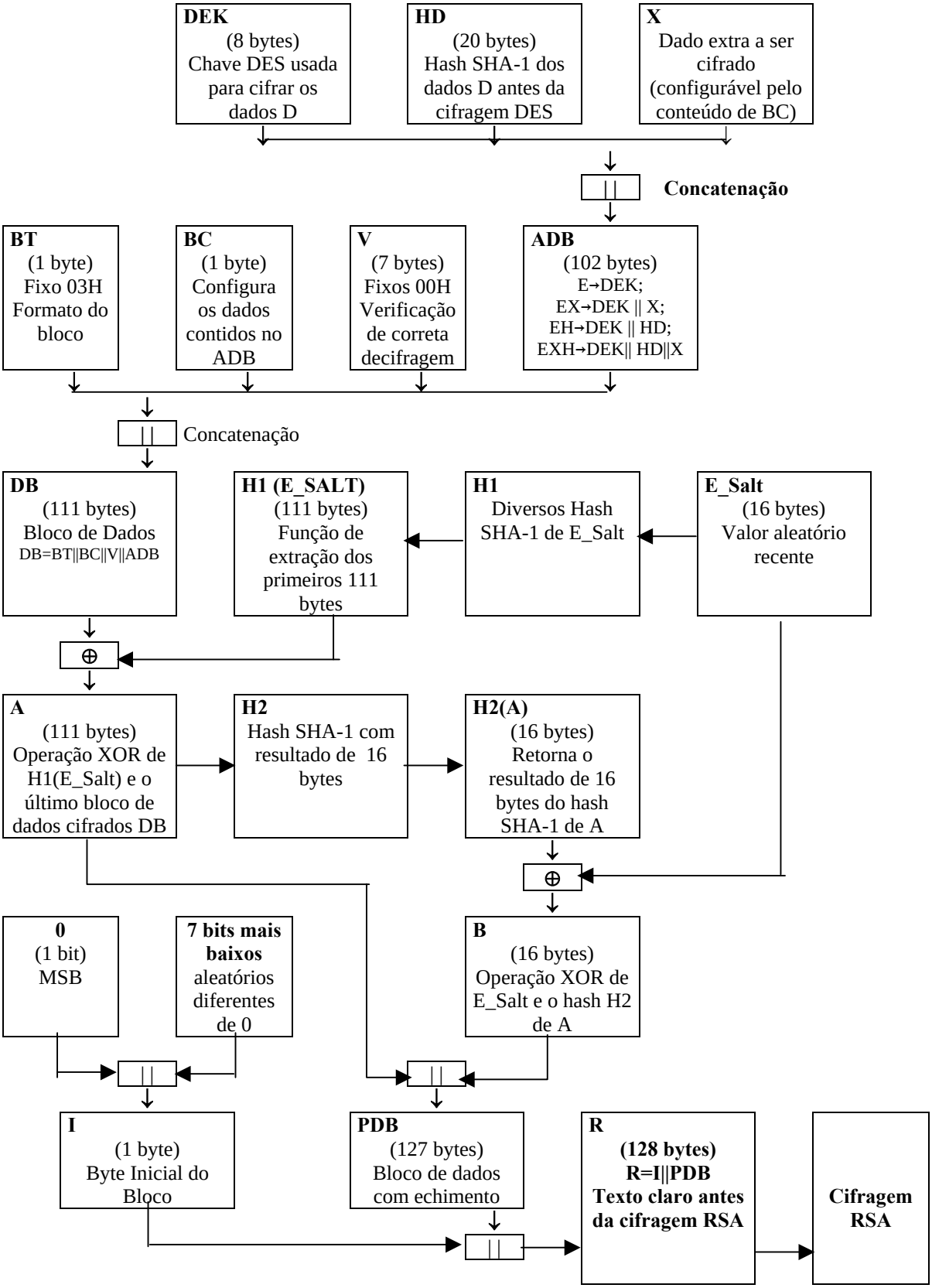

Figura 6.1: Fluxo de Processamento OAEP

Foi visto que o processamento OAEP é utilizado pelo SET com as primitivas de cifragem: **E** (Cifragem Assimétrica), **EX** (Cifragem Assimétrica Extra), **EH** (Cifragem Assimétrica com Integridade) e **EXH** (Cifragem Assimétrica Extra com Integridade) definidas no Sistema.

Concluindo este capítulo, com base na figura anterior, examina-se como essas primitivas de cifragem são ativadas, de acordo com o tipo de mensagem do SET. No OAEP, o Bloco de Dados Atual (**ADB**), que após processamento será cifrado pelo RSA, contem um ou mais campos **DEK**, **HD**, e **X**: o campo **DEK** é a chave *DES* de 8 bytes de cifragem simétrica dos dados; **HD** é um *hash* de 20 bytes dos dados antes da cifragem; **X** é o dado extra a ser cifrado. Os blocos que estão contidos no **ADB** dependem da primitiva de cifragem utilizada, que é indicada pelo byte de conteúdo **BC**, conforme a tabela abaixo:

Valor de BC	**Primitiva de Cifragem**	**Campos no ADB**
0	**E** (Cifragem Assimétrica)	**DEK**
> 0 e < 80H	**EX** (Cifragem Assimétrica Extra)	**DEK | | X**
80 H	**EH** (Cifragem Assimétrica com Integridade)	**DEK | | HD**
> 81 H	**EXH** (Cifragem Assim. Extra c/ Integridade)	**DEK | | HD | | X**

O tipo de dado extra **X** também é definido pelo valor de **BC**, a saber:

00 (80) – sem dado extra, isto é, **X** não está presente

01 (81) – **PANData** (*Primary Account Number* - é transportado no formulário de compra assinado na mensagem de Pedido de compra – **PReq**).

02 (82) – **PANData** (é tranportado na mensagem de Pedido de Certificado – **CertReq**).

03 (83) – **PANToken** (é transportado no formulário não assinado do Pedido de Compra – **PReq**)

04 (84) – **PANOnly** (é transportado na mensagem **RegFormReq**)

05 (85) – **AcctData** (contém informação de identificação sobre o Comerciante ou Portal de Pagamento numa mensagem **CertReq**).

Capítulo 7

SOFTWARE E *HARDWARE* PARA O SET

Este capítulo expõe o esforço para manutenção da segurança pelo tratamento criptográfico e os aspectos técnicos de *software* e *hardware* disponíveis para tal tratamento. Alguns produtos comerciais de empresas desenvolvedoras para tecnologia SET são detalhados.

7.1. O esforço para o tratamento criptográfico

Como vimos ao longo deste estudo, o SET requer um tremendo poder de processamento para manusear a complexidade de sua natureza criptográfica. Enquanto 10 transações SSL (*Secure Sockets Layer*) por segundo podem ser processadas num processador Pentium de 100 a 200 Mhz, este número cai para uma ou duas por segundo para o SET. Os cálculos relacionados à segurança sempre absorvem até 95 % da capacidade de processamento de um servidor [4], deixando pouco espaço para outro trabalho. Estes gargalos são responsáveis pela sobrecarga do servidor, negativa de conexões e perdas de negócios.

O SET é altamente dependente de uma cifragem robusta, e o desafio é fornecer a potência necessária para atender a demanda. Dependendo do *site*, do orçamento, do número histórico e previsto de transações de venda, e da atual tecnologia de servidor *Web*, deve haver a necessidade de uma potência de processamento apropriada para manusear o tráfego, ou correr o risco de perda de negócios.

Como amplamente discutido, o SET usa a cifragem para uma variedade de características, incluindo:

- Autenticação
- Privacidade
- Integridade de Mensagem

O SET requer acesso contínuo ao processamento criptográfico para implementar estas características. Como um exemplo, o SET usa operações tipo assinatura, *hashing*, e verificação de Certificado no mínimo 15 vezes durante um simples processamento de par de mensagens de um pedido de compra.

A criptografia para o SET pode ser implementada através de rotinas de *software* (processador intensivo) ou operações de *hardware* (processador assistente). As ferramentas de *software* oferecem estas rotinas, invocadas como cálculos que são solicitadas pelos programas. As *Crypto-boxes* de *hardware* se apresentam como cartões adicionais (*add-on boards*) ou computadores separados que operam como servidores de descarga do trabalho no modo cliente servidor. Cada uma destas abordagens fornece uma completa transparência de rendimento para os programas ou usuários afetados. Um Servidor de Comerciante que é baseado na cifragem por *software* se comunica tão bem com o Portal de Pagamento como outro que é baseado em cifragem por *hardware*, independentemente de como o Portal de Pagamento implementa sua criptografia.

Alguns dos principais fatores que irão ajudar na decisão de se utilizar a criptografia somente por *software* ou por *hardware*, por ser mais apropriada, incluem:

- Custos dos produtos e disponibilidade de capital de investimento para o *site*.
- Requisitos de desempenho para servir os clientes num tempo razoável.
- Número de pedidos por unidade de tempo que requerem processamento criptográfico.

O SET requer cálculos criptográficos para cada par de mensagens que ele usa para processar pedidos de débito, pedidos de captura, e para finalidades de administração. Dentro da própria experiência de *shopping*, a apresentação da porção de pagamento confia no processamento criptográfico da carteira eletrônica (*electronic wallet*) do consumidor, no sistema POS (*Point of Sale*) do Comerciante, e no Portal de Pagamento do Adquirente. Tipicamente, maior o volume de transações, maior a necessidade de processamento criptográfico robusto, e maior o custo deste processamento.

O sistema POS do Comerciante do SET deve confiar nas ferramentas do software desenvolvedor que integram as bibliotecas comerciais de cifragem de servidores do Comerciante existentes. Um exemplo disto é o POS GlobeSet, que usa a ferramenta S/PAY da RSA. Os programadores tipicamente usam software "fora–de-prateleira" na forma de Programas de Interfaces de Aplicação (Application Programs Interfaces- API) que devem ser chamadas quando necessário.

As Carteiras Eletrônicas (*E-wallet*) sempre contém aqueles componentes de ferramentas necessárias para o processamento local que podem ser baixadas por navegadores da *Web* compatíveis com um "click" de poucos botões. Para muitos consumidores, as rotinas de *software* dentro da carteira eletrônica irão ser suficientes para o uso pessoal, desde que as demandas sejam relativamente leves. Para os Comerciantes, níveis muito maiores de processamento devem ser necessários, determinados pelo demanda do *site* comercial.

O trabalho criptográfico do SET pode também ser implementado via *hardware* através de componentes de cifragem especialmente projetados (*crypto-boxes* e *crypto-cards*). Os Comerciantes podem adicionar cartões de cifragem/segurança nos mesmos servidores que fornecem acessos a compras na *Web*. O atual arranjo de cartões de cifragem suporta muito mais sistemas de operação de computadores. Estes cartões *add-on* operam em faixa similar de compatibilidade com os cartões coprocessadores que estão presentes onde alto poder computacional é requerido.

A criptografia vem ser mais complicada do que rápida. Aqui está uma visão superficial de como o processamento criptográfico do SET é conduzido desde que tratado como uma série de camadas abstratas:

- Processamento Criptográfico Primitivo (computando um *hash*, gerando números aleatórios, etc).
- Processamento Criptográfico em nível-de-mensagem do SET (*hashes* SHA-1, cifragem DES, criação de envelopes PKCS, etc).
- Software de aplicação SET de invocação da criptografia em nível-de-mensagem do SET.

Cada uma dessas camadas é entregue em forma de *software* que chamamos de ferramentas. Estas ferramentas provêem acesso aos seus serviços via APIs, que são chamadas pelos

programas da camada mais alta (como visto pela movimentação para cima desde o processamento primitivo até as camadas de processamento da aplicação).

A Microsoft CryptoAPI é uma ferramenta de criptografia de finalidade geral que os desenvolvedores podem usar em vez do desenvolvimento desde da linha de partida de seus próprios programas criptográficos. O S/PAY da RSA executa o processamento criptográfico que é específico para o SET pela combinação de passos criptográficos primitivos nos algoritmos padrão que o SET usa (SHA-1, envelopes PKCS, cifragem *DES* , etc). Finalmente o *software* da camada de aplicação POS usa o S/PAY (ou uma ferramenta similar) na preparação e processamento corrente do atual par de mensagens do SET.

Além disso, as operações executadas pela camada mais baixa (primitiva) devem ser implementadas via somente *software* ou a combinação de *software* e *firmware*. Este *firmware* consiste de algoritmos criptográficos em memórias PROM que operam muito rapidamente por serem circuitos eletrônicos.

7.2. APIs de criptografia primitiva

A camada mais baixa do processamento criptográfico instrui o *hadware* para executar algum trabalho (computa um *hash*, gera um número aleatório, etc). Desenvolvedores de APIs de criptografia incluem Microsoft (CryptoAPI), RSA, e outros. Estas bibliotecas são requeridas pelos desenvolvedores de *software* de aplicação do SET. Elas são tipicamente parte de algum sistema SET disponível no mercado.

As ferramentas do SET como a S/PAY da RSA são o próximo nível mais alto de camada do *software* que usa APIs de criptografia primitiva como necessárias para executar o trabalho de segurança. O S/PAY conhece como transportar o trabalho que as mensagens do SET necessitam, sem que o desenvolvedor especifique cada passo. Por exemplo, com sua API, um desenvolvedor pode simplesmente fornecer o conteúdo e chamar uma operação de cifragem DES. O resultado pode ser colocado diretamente dentro de uma mensagem enquanto ela está sendo construída. Estas ferramentas não somente preservam o desenvolvedor de esforço considerável, elas também ajudam a assegurar que o processamento criptográfico é consistente, removendo quaisquer relações de implementações proprietárias. A RSA, como

um dos arquitetos do SET, removeu uma das mais formidáveis barreiras para o desenvolvimento de *software* do nível de aplicação do SET com o S/PAY, e muitos produtos comercias do SET se beneficiam disso. Novamente, estas ferramentas do SET serão tipicamente um componente de qualquer sistema SET disponível comercialmente.

7.3. Ferramentas da camada-de-aplicação do SET

O *software* de Aplicação do Comerciante do SET (POS) é aquele que será comprado e instalado pelo Comerciante. Tais aplicações residem sobre as duas ferramentas de menor nível, adicionando negócios específicos do SET e regras de processamento. Elas também oferecem uma serie de APIs que serão usadas para customizar a instalação para o negócio. Estas APIs fornecem flexibilidade para executar a programação relativamente simples para se ligar a sistemas legados (sistemas já existentes de processamento de dados do comerciante) de fontes de dados, para requisitar informação em processamento do sistema, e assim por diante. Essas ferramentas de biblioteca de funções podem ser modificadas ou expandidas para atender algumas necessidades específicas. Funções adicionais podem ser adicionadas através de linguagens de programação tais como $C++$ ou Java. O quadro a seguir lista alguns sistemas POS comerciais para o SET.

Sistemas POS comerciais para o SET			
Item	**GlobSet**	**IBM**	**VeriFone**
Nome do Produto	GlobSet POS	CommercePOINT eTill	vPOS
Suporte para criptografia assistida por *hardware*	Sim	Sim	Sim
Web URL	“www.globeset.com”	“www.ibm.com”	“www.verifone.com

Os sistemas POS para o SET disponíveis no mercado (*software*) irão transportar uma implementação do tipo somente *software* de processamento criptográfico. Após um tempo, caso o processamento se mostre muito lento, criando gargalos, ou conduzindo o servidor para uma queda de desempenho, que pode inclusive acarretar uma desgastante paralização, a criptografia assistida por *hardware* deve ser considerada.

7.4. Criptografia assistida por *hardware*

Como em qualquer outra decisão de investimento, decisões sobre *hardware* criptográfico deveriam ser feitas usando uma análise de custo-benefício. Diversos fatores influem na análise, porém os itens primários que deveriam ser considerados são a utilização do processador e o número de processos de pagamento que estão sendo executados nos horários de pico .

A execução de uma análise de custo-benefício é suficientemente inteligente quando todos os parâmetros são conhecidos e mensuráreis, porém é mais inteligente quando se tenta determinar os custos das perdas de negócios devido ao incremento no tempo de processamento nos seus servidores de *Web*.

Quando o trabalho de cifragem do SET é adicionado ao servidor, tempos de processamento podem baixar de várias centenas de transações por segundo para um pouco mais de algumas transações por segundo. O *hardware* especificamente projetado para a cifragem vem em salvamento quando o processamento do sistema começa a caminhar para uma parada.

O *hardware* de cifragem é fornecido desde os servidores *stand-alone* (servidores de segurança) ou como cartões *add-on*, que são instalados dentro dos *slots* disponíveis do servidor do sistema POS.

Os dispositivos de cifragem podem também monitorar a segurança dos dados que eles processam e alertar os administradores do sistema quando os problemas ocorrem. Eles podem também monitorar as atividades de chaves privadas. Se uma brecha de segurança é detectada, os dados em risco devem ser anulados, tornando a brecha não efetiva. O *hardware* de cifragem reduz ou elimina o risco de chaves privadas serem roubadas pela colocação das chaves privadas dentro do próprio *hardware*. Um estorno de chaves privadas armazenadas em *hardware* é relativamente difícil ou, em alguns casos impossível. Se o *hardware* tiver sido estampado pelo fabricante com uma chave privada pré-assinada, há a necessidade de mudar o cartão para obter uma chave diferente. Esta forma requer que se tenha em mão um cartão reserva.

Cartões A*dd-on* e Servidores de Segurança

Os cartões criptográficos *add-on* e servidores funcionam usando a mesma abordagem em camadas que é usada pela criptografia de somente-*software*. Alguns fabricantes de cartões fornecem ferramentas que se interfaceiam com a ferramenta S/PAY para as APIs usadas para controlar as operações dos dispositivos. Os fornecedores de *hardware* devem suprir suas próprias bibliotecas, porém sempre usam as mesmas ferramentas que os *software* de criptografia usam.

As APIs criptográficas são armazenadas na memória (PROM) dos cartões *add-on* e servidores, junto com algumas chaves privadas que devem ser armazenadas. Eles tipicamente possuem sua própria RAM de armazenamento para elevar a velocidade de processamento. Pela colocação de chaves e certificados diretamente no *hardware*, você está garantindo que os dados sensíveis do SET não estão vulneráveis à ataques externos, então se atingindo as premissas iniciais de segurança do SET, como vistas neste estudo.

Os servidores de segurança seguem o modelo de arquitetura cliente-servidor, com o sistema POS do SET como um cliente. Considerando que esses servidores de segurança contem seus próprios processadores, eles provêem uma console de interface do operador para controlar a conectividade, gerenciamento de chave, e serviços de gerenciamento de certificados.

As implementações de *hardware* são dispositivos de finalidade especial que oferecem ambas seguranças física e lógica. Os Módulos de Hardware de Criptografia (*Hardware Cryptography Modules* – HCMs) são conectados a outras plataformas de servidores através de conexões físicas de I/O (direto ou via rede). Eles usam suas próprias bases de dados para armazenar as chaves que não estão dentro do próprio *hardware*. Uma API similar aos softwares APIs permite que múltiplas aplicações acessem os serviços dos dispositivos.

Algumas vantagens no uso de HCM incluem:

- Segurança incrementada, uma vez que se está descarregando o processamento para o *hardware* que pode ser protegido na forma que se julgar melhor. O armazenamento em *hardware* de informações de chave desencorajam sempre as mais tenazes tentativas.

- Melhoria de desempenho através da descarga de processamento, removendo quaisquer requerimentos de compartilhamento de ciclos de processamento de outros servidores na rede.

- Redução dos custos de desenvolvimento, desde que múltiplas aplicações podem compartilhar o mesmo dispositivo. Futuramente, uma simples API pode ser usada para comutar do processamento somente por *software* para o processamento assistido por *hardware*, eliminando a necessidade de modificar programas existentes que requeiram isto.

Baseado em circunstâncias específicas um cartão *add-on* ou um servidor dedicado de criptografia pode eliminar quaisquer problemas relacionados ao SET que se pode experimentar.

O quadro em seguida lista alguns atributos para alguns produtos assistidos por *hardware*, cartões *add-on* e servidores de segurança comumente encontrados no mercado.

Produtos criptográficos comerciais assistidos por hardware				
Produtos	**ATALLA**	**SPYRUS**	**RAINBOWN THECNOLOGIES**	**IBM**
Tipo de Hardware	Cartões ou servidores	Cartões	Cartões	Cartões ou Servidores
APIs Suportadas	Microsoft CryptoAPI	Cartão-Especifico e Microsoft CryptoAPI	Cryptoki CDSA Microsoft CryptoAPI NSAPI	Cartão-Específico Microsoft CryptoAPI Compatível
Compatibili-dade com Sistemas Operacionais	Windows NT Unix	Dos 3.0 Windows 3.x Win95 Windows NT SunOS AIX Outros sistemas UNIX	Windows NT Linus BSDI Free BSD Solaris MacOS Outros sistemasUNIX	AIX OS/2 Windows NT
URL *Web* para suporte completo do SET	Sim “www.attala.com”	Sim “www.spyrus.com”	Sim “isg.rnbo.com”	Sim “www.ibm.com/security/ criptocard.html”

As implementações de criptografia do SET devem requerer que se considere um ambiente de processamento mais robusto que aquele que está disponível através de criptografia somente por *software*. Como os problemas começam na superfície do seu ambiente, deve-se estar apto para remover o *software* que causa problema e substituí-lo por *hardware* que resolva o problema. Desde que o desenvolvedores do SET não podem antever todos os ambientes nos quais seus produtos irão operar, eles desenvolveram seus sistemas para permitir estes tipos de crescimento com uma quantidade razoável de esforço de sua parte.

7.5. Software de carteiras eletrônicas e certificados digitais

O SET espelha o mundo físico da aceitação de cartão de pagamento no processamento dentro do mundo digital do comércio eletrônico.

Aqui, são enfocados os elementos do consumidor requeridos para transações sob o SET. Embora seja importante que os operadores do Comerciante e proprietários de negócios entendam como os consumidores se interfaceiam com seus Servidores de Comerciante, não é responsabilidade direta do Comerciante assegurar que os consumidores estejam prontos para interoperar. Este é o trabalho dos bancos fornecedores de cartão e companhias de cartão de crédito. Será examinada então a informação concernente a *E-wallets*, Certificados digitais de portadores de cartão, e como ambos se interfaceiam com o *software* POS do Comerciante

Para se preparar para comprar, os consumidores devem possuir no mínimo dois componentes necessários para uma transação SET:

- Uma carteira eletrônica (*E-wallet*) – *software* que acompanha um navegador de *Web* que armazena e gerencia contas de cartão de crédito.
- Um certificado digital (ID Digital) – identificação que serve como um suporte para ambas as partes, o equivalente eletrônico da peça física de plástico (cartão) e a assinatura na sua parte de trás.

As carteiras eletrônicas permitem ao consumidor armazenar informações privadas – cartões de crédito, cartões de débito, nome e informação de endereço – no seu PC e recuperar aquelas informações rápida e seguramente. Uma vez que uma *E-wallet* esteja disponível no seu navegador *Web*, um portador de cartão pode começar o processo de obtenção do Certificado

digital SET. Na forma mais simples, para cada número de cartão que um usuário da *Web* detenha e deseje usar no mundo *on-line*, os seguintes passos devem ser executados:

1. O usuário visita o *site* apropriado de fornecimento de certificado do Fornecedor de Cartão.

2. Uma aplicação de registro (formulário de autenticação) é completada para cada crédito específico ou carga de cartão.

3. Uma verificação de segurança para a autenticação é executada pelo fornecedor na data apresentada.

4. Uma resposta aprovada do Fornecedor começa o processo de geração de certificado. Uma vez pronto, o certificado é transmitido de volta ao portador de cartão para armazenamento e gerenciamento pela *e-wallet*.

A *E-wallet* endereça dois importantes assuntos relacionados ao SET: elas oferecem ambos segurança e conveniência. O consumidor pode abrir sua *e-wallet* protegida por senha somente quando a senha estiver correta, e uma vez que faça uma compra, sua identidade é validada através do uso do certificado digital.

As *E-wallets* podem ser adquiridas via Internet. Qualquer *software* de *E-wallet* compatível com o SET deve ter estas características mínimas para ser considerado útil:

- Ele deveria possuir uma interface de usuário que o consumidor ache fácil de usar.
- Ele deve estar apto para gerenciar múltiplos certificados para vários cartões e tipos de cartões (cartões de crédito, cartões de débito, etc).
- Ele deve oferecer proteção de acesso (senha).
- Ele deveria criar um ambiente amigável no qual os consumidores possam facilmente gerenciar seus certificados.
- Ele deve estar apto a se comunicar com Servidores do Comerciante compatíveis com o SET.

No mundo físico, o Fornecedor de Cartões verifica a identidade de um consumidor antes do fornecimento de um cartão de crédito físico. O consumidor então usa uma carteira convencional para guardar os cartões de crédito que ele recebeu de um ou mais bancos de

fornecimento. Quando o consumidor faz uma compra numa loja de departamentos, ele escolhe seu cartão de crédito da carteira e entrega-o para o Comerciante. O Comerciante coloca o cartão do consumidor em um dispositivo POS (*Point of Sale*), espera pela autorização de pagamento, e entrega ao consumidor seu *slip* eletrônico, que ele então assina. O Comerciante compara a assinatura com aquela atrás do cartão físico e, se elas conferem, entrega a compra ao consumidor, seu cartão e seu recibo.

A Carteira Virtual, entretanto, reside no PC do consumidor. Em vez de fazer uma viagem ao shopping para as suas compras, o consumidor vai comprando na Internet. Atravessando o seu *site* favorito na *Web*, que o cumprimenta com sugestões personalizadas de presentes baseadas na história passada de suas compras, ele seleciona os bens que deseja comprar e os coloca em sua cesta de compras. Para completar a porção de pagamento da transação, o consumidor abre sua *E-wallet* e seleciona o certificado já armazenado para completar sua compra. Sem sair do seu PC, o usuário assina sua transação com o seu Certificado Digital e o fim da transação do SET ocorre. O Servidor do Comerciante dotado do SET então recebe a transação, decifra a mensagem, executa a autenticação, e envia um conhecimento para o usuário que seu pedido foi recebido e está indo para processamento.

Um número de companhias tem se comissionado para fornecimento dos serviços e produtos do SET incluindo Servidores de Comerciantes, Portais de Pagamento, produtos de CA e bibliotecas de *software* do SET (ferramentas) – tantos, de fato, que a decisão de como diferenciar entre várias ofertas pode conduzir à confusão e frustração. As *E-wallets* não são exceção. IBM, Microsoft, and GlobeSet , todos oferecem *E-wallets*, assim como outros vendedores como BankGate e Mairthen. Este tópico descreve diversas dessas *E-wallets*, incluindo suas funções e características.

A carteira GlobeSet foi usada pela American Express e Wal-Mart em junho de 1997 para completar a primeira transação Internet dos EUA na qual bens foram vendidos usando o protocolo SET. A GlobeSet Wallet é uma carteira eletrônica *plug-in* para navegadores Web e outras aplicações. Entre suas funções estão a geração de pedidos de compras, recebimento de respostas de compras e a amostragem dos detalhes da transação de compra e histórico. A GlobeSet descreve sua carteira como “totalmente expansível” significando que ela pode suportar múltiplos tipos de instrumentos financeiros e transações através de módulos *plug-in*. A carteira trabalha com outros produtos GlobeSet tais como o POS da GlobeSet e Servidor de

Comerciante, mas mais importantemente ela interopera com sistemas de pagamento de outros vendedores de *software* do SET.

A GlobeSet cita as seguintes características da sua carteira eletrônica :

- Uma interface de usuário “intuitiva” para fazer compras na Internet
- Um perfil de portador de cartão configurável, incluindo endereço de faturamento, endereço de entrega e instruções, informação de conta para os múltiplos cartões de crédito, a ligação de específicas contas de cartões para compras específicas, e a assinatura do Número de Identificação Pessoal (PIN) como um nível adicional de segurança.
- Uma mostra concisa de informação da transação de compra, incluindo contato de Comerciante e informação de endereço e números de suporte 0800, cotas de vendas incluindo quantidades, preços, taxas, e custos de transporte, mostragem em tempo real do acompanhamento das partes condutoras da transação e recibos detalhados de compra.
- Um *log* histórico de cotas em progresso bem como as pendentes, pagamentos, e compras canceladas.
- Suporte para gerenciamento de certificado, investigação de transação, reversão e crédito.
- Suporte para arquivamento de registros de transações para revisão do portador de cartão, auditoria e relatórios.

A carteira atua como ambas, uma aplicação *stand-alone* e um *plug-in* de navegador. Para administrar as características da carteira, o portador lança-a do desktop como uma aplicação. Para usar a carteira dentro da transação, o navegador *Web* ativa-a de dentro da “experiência de Compras”.

A versão 1.0 da Carteira GlobeSet foi liberada para o público em dezembro de 1997. Ela opera sob os sistemas operacionais Windows NT Server ou Workstation NT, Windows 95, e Sun Solaris . Ela também requer um dos seguintes navegadores de Web: Netscape Navigator 3.0X, Netscape Communicator 4.0 ou mais, ou Microsoft Internet Explorer 4.0 ou mais.

Como a carteira da GlobeSet, a carteira da Microsoft é um sistema de *software* de pagamento para armazenar informação privada tal como cartões de débito e crédito junto com informação de faturamento e para remessa. A carteira da Microsoft adicionalmente opera outros métodos de pagamento, incluindo dinheiro e micropagamentos (geralmente pagamentos menores que US$ 10) *on-line* .

Originalmente desenvolvida como complemento para a edição 1996 do Internet Information Server (IIS) da Microsoft, a carteira MS teve uma resposta ao crescimento do comércio na Internet e acompanhou as preocupações de segurança na Internet.

Se, por exemplo, o usuário do PC inclui o "Armazenamento Protegido", uma característica criptográfica de armazenamento do navegador Internet Explorer, a informação do cartão de crédito é armazenada lá. De outra forma, a Carteira armazena o número do cartão de crédito no *registry* da máquina do usuário. A despeito do método de armazenamento, o usuário deve entrar com a senha para acessar a informação cifrada.

A carteira cifra o número do cartão de crédito do usuário usando uma chave de *hashing*.

Diferente da carteira GlobeSet, a carteira da Microsoft usa o protocolo de cifragem SSL que muitos navegadores e servidores de Web suportam. Os desenvolvedores da Microsoft decidiram que, dada a falta de um padrão da indústria no tempo em que estavam projetando sua carteira, era seguro incorporar o SSL em seu projeto. Suporte ao SET deve ser adicionado através de terceiras partes desenvolvedoras de *plug-ins* para a carteira MS Wallet.

A carteira Microsoft usa uma interface programática para o desenvolvimento por terceiras partes de suporte de cartão de crédito de "etiqueta privada" e protocolos adicionais de cifragem incluindo o SET. Ela tem um Modelo de Objeto Componente (Component Object Model – COM) extensivo de arquitetura aberta que permite a operação com outros protocolos e métodos de pagamento.

A carteira CommercePOINT da IBM é um *plug-in* de navegador que provê segurança aumentada usando o protocolo SET. Ele, também, armazena o pagamento eletrônico de cartões e informações relacionadas e gerenciamento de atividades de cartão de pagamento eletrônico. O usuário pode facilmente adicionar e apagar cartões de pagamento de sua carteira e modificar a informação de faturamento e transporte.

Para usar a aplicação CommercePOINT, o usuário lança a carteira através do seu navegador de *Web*. Antes de fazer a compra, entretanto, ele deve primeiro entrar com a senha ou PIN, outra característica de segurança que é padrão com todas *E-wallets*.

Uma vez que o usuário seleciona o item que deseja comprar, os dados da transação (pedido de compra) são cifrados separadamente dos dados de conta do cartão e ambos envelopes digitais

são transmitidos via Internet. O *software* do servidor do Comerciante pode decifrar somente os dados do pedido de compra, e os dados de conta podem ser decifrados somente pelo Portal de pagamento. Uma vez que o portal processe a requisição do cartão de compra do usuário, ele notifica o Comerciante que o débito foi aprovado. Finalmente, o usuário vê uma janela de Recibo de Pedido, que o notifica que sua compra está completada.

Para sumarizar as funções e características da Carteira CommercePOINT , ela:

- Possui uma interface gráfica de usuário.
- Permite ao usuário gerenciar múltiplos cartões de crédito e certificados.
- Usa a proteção PIN, adicionando outra camada de segurança.
- Provê um ambiente direto para gerenciamento de cartões de crédito e certificados.
- Armazena registros de compras para gerenciamento de contas pessoais.
- Comunica-se com Servidores de Comerciante dotados do SET.

A PayPurse é outra carteira dotada do SET. Sempre que o usuário compra um produto em um *site* de Comerciante dotado do SET, a PayPurse é automaticamente ativada e usa a informação armazenada sobre o usuário de cartão de crédito. A Trintech provê a PayPurse para bancos de fornecimento para distribuição aos seus portadores de cartão. O consumidor pode fazer então o *download* do *software* de instalação PayPurse do *site* da *Web* de seu banco fornecedor. O software de instalação caminha através de passos necessários para carregar a aplicação em seu PC, criando um ícone no seu desktop para a PayPurse da Trintech .

A tela principal de pagamento da PayPurse da Trintech oferece quatro opções primárias:

- Adição de um cartão de crédito ou mudança de detalhes sobre o cartão.
- Adição ou mudança de detalhes do endereço.
- Revisão de transações de compra prévias.
- *Download* de um Certificado digital para um cartão.

A Verifone tem colaborado com as Associações de Marca no desenvolvimento do SET, usando suas experiências no mundo físico dos sistemas de pagamento seguros para ajudar desenvolver o protocolo SET para o mundo virtual. Em adição aos seus produtos de *software* dotados do SET está sua vWallet, que permite ao consumidor fazer e acompanhar compras usando certificados do SET.

7.6. Software POS para Servidores de Comerciantes dotados do SET

Peças de sistemas para operação com o SET tem começado aparecer em cena desde 1997. Estas peças mostram-se como soluções que provêem processamento *end-to-end*. Elas incluem *E-wallet* de portadores de cartão, *software* POS de Comerciantes, sistemas de Autoridades de Certificação (CA) e *software* de aplicação de Portal de pagamento para Adquirentes. A interoperabilidade entre estes *softwares* é garantida, desde que eles sejam desenvolvidos com um comum entendimento das especificações do SET.

A IBM oferece um produto chamado CommercePOINT eTilll que trabalha sob seus Servidores de Comerciante Net.Commerce e Lotus Domino. O produto também pode ser integrado a outros servidores de Comerciantes. O eTill provê uma utilidade para gerenciar seus certificados tão bem como todas cifragems de formatação de mensagens relacionadas ao SET, e serviços de decifragem necessários. Um sistema de administração e configuração baseada em navegadores é provido pelo eTill para *webmasters* de Comerciantes.

A solução da GlobeSet é chamada GlobeSet POS e é componente do sistema *GlobeSet Payment System* , liberado em dezembro de 1997. O sistema de pagamento é particularmente vendido através de terceiras partes e revendedores de valor agregado (*Vallue Added Resellers* -VARs) que o integram em sistemas fechados. De acordo com a GlobeSet , seus sistemas POS provêem:

- Interoperabilidade provada.
- Uso com múltiplos portais de pagamento.
- Um SDK (*Software Development Kit*) para facilitar integração das aplicações de lojas.
- Possibilidade de uso de *hardware* criptográfico.
- Capacidades de suporte para múltiplos Comerciantes (*shopping center*) e capacidades de somente-autorização.

O POS GlobeSet está disponível para o Windows NT 4.0, Solaris 2.5 ou superior, HP-UX 10.x, e IBM AIX 4.2. Os sistemas de base de dados que são suportados incluem Oracle, Microsoft SQL Server, Sybase, e Informix. A GlobeSet oferece SDK para simplificar a integração de uma aplicação de experiência de compras com o *software* POS. Seu Gerente de Administração provê uma interface tipo navegador para suas funções de administração incluindo:

- Iniciação, parada, rearme, e monitoração do status do Servidor POS.
- Exame dos certificados que estão disponíveis para uso.
- Configuração e monitoração dos Comerciantes atualizados.
- Processamento manual de lote e atividade de transação.

A VeriFone é um dos fornecedores líderes de terminais POS nas indústrias de varejo, e assim é natural que suporte as operações POS na Internet, também. O Sistema de Comércio VeriFone provê um componente de *software* obediente ao SET, chamado vPOS. Outros componentes na oferta incluem vGate para os Portais de pagamento vWallet para os Portadores de Cartão. Como os outros desenvolvedores de *software*, a VeriFone oferece uma solução end-to-end para o processamento do SET. O vPOS opera sob o Microsoft Internet Information Server (IIS), Netscape Enterprise Server, e Oracle Web Server. Ele também provê uma interface de navegador de Web para gerenciamento e administração do sistema.

Sistemas com o CommercePOINT eTill , GlobSet POS , e VeriFone vPOS, todos usam a ferramenta RSA S/PAY para manusear os processamentos de criptografia necessários do SET via APIs. Outras APIs que estes componentes de *software* POS adicionam permitem acesso às suas características sem ser necessária a intimidade de detalhes de sua implementação ou tenha que se negociar com o processamento a nível criptográfico. É possível escrever uma implementação do software POS do SET usando ferramentas como S/PAY da RSA e o Guia do Programador do SET, porém é mais prático a utilização de implementações comercialmente já existentes.

Estão ainda disponíveis outras opções de implementações para o SET. Atualmente a CyberCash oferece um sistema compatível com o SET, visto que originalmente ele difere dos tradicionais processamentos de cartão do SET. As soluções de dispositivos de pagamento da CyberCash tais como cartões de crédito, caixa, cheques eletrônicos e protocolo global de pagamentos são construídos sob os níveis de segurança associados com o Sistema de Pagamento proprietário da CyberCash.

7.7. Funções e Segurança do Software POS

Será utilizado o eTill da IBM como um exemplo de como os sistemas POS operam dentro de um servidor de Comerciante. O eTill, ele próprio, consiste dos seguintes componentes:

- Uma aplicação JAVA recebe mensagens de ambos Portador de Cartão e Portal de Pagamento e envia mensagens para Portais de Pagamento.
- Uma biblioteca de APIs em C permite acesso por qualquer *software* do Servidor do Comerciante que é escrito em C ou $C++$.
- A biblioteca compartilhada de saída modificável pelo usuário (*Modifiable User Exit Shared Library*) executa a implementação das saídas opcionais dos usuários ou funções de chamada de volta para dinamicamente recuperar informação enquanto uma transação está em processo.
- Configuração de Bases de dados são usadas para customizar o sistema para acesso flexível ao Portal de Pagamento e outras aplicações.
- Bases de dados de transação mantém informação acerca de pedidos e transações SET.
- Formas de amostra baseadas em navegador executam customização posterior de configuração, administração, relatórios, interfaces do usuário de geração de mensagens.
- Utilidade de Registro de Certificado é usada para obter Certificados do Comerciante.

O eTill usa Perfis de Configuração para sua própria instalação, dos componentes do SET, da configuração do sistema de pagamento, da configuração do(s) Adquirente(s), da configuração de Marca(s) e da configuração dos dias em que o Adquirente não está disponível para operação.

Enquanto o SET reduz o risco de roubo de informação de cartão de pagamento durante o trajeto entre entidades finais, ele em nada garante a segurança dos ambientes onde está instalado. É responsabilidade do Comerciante definir a política de segurança para qualquer *hardware* ou *software* que eles instalem. Aqui existem algumas coisas que deveriam ser consideradas em tal política:

- Deve-se dedicar um servidor e um *firewall* ao Servidor de Comerciante e *software* POS, isolando-os de ambos da Internet e de outros domínios dentro da organização. Deve-se remover todo *software* do servidor desnecessário que não tenha finalidades

especificamente operacionais. Isto deve incluir compiladores de linguagens, bibliotecas, utilitários administrativos, e *log-ins* e senhas fornecidos de fábrica.

- Deve-se somente abrir portas do protocolo definido pelo SET para computadores através do *firewall*.
- O *firewall* não deveria permitir operações de transferência de arquivos (FTP - *File Transfer Protocol*) ou comandos de acesso remoto por outros estações de trabalho (*telnet* ou xterm), ou permitir outros acessos por outras portas.
- Não se opera *software* de transferência de arquivos, comandos de acesso remoto via rede, ou sistemas de *e-mail* no Servidor do Comerciante e *hardware* do POS.
- Sempre que operações remotas (telnet, xterm, etc) forem necessárias, deve-se ter certeza que estão protegidas por protocolos de segurança, como por exemplo o SSH (Secure Shell), que permite acesso seguro entre computadores via rede, através do estabelecimento das conexões com proteções criptográficas [4].
- Nunca deveriam ser feitas conexões do *software* do Servidor do Comerciante diretamente ao *software* POS (ao invés, APIs deveriam ser utilizadas).

Em adição à segurança do *software* POS e do *software* do Servidor do Comerciante, os *webmasters* ou administradores de segurança deveriam estar seguros que toda informação relacionada à transação não está vulnerável a ataques externos.

Em muitas transações de compra, os Portais de Pagamento podem estar instruídos para retornar o número da conta dos Portadores de Cartão para reconciliação de pagamento, auditoria, e processamento de disputa. É crítico que estes dados estejam seguramente armazenados. As bases de dados deveriam ser protegidas por senhas, e o sistema deveria ser configurado para garantir que acessos não autorizados não sejam possíveis.

Capítulo 8

Considerações Finais e Conclusões

Concluímos este trabalho apresentando uma breve revisão da criptografia utilizada para a proteção de mensagens do SET, e uma breve discussão sobre os outros principais protocolos de segurança. Diversas considerações sobre o SET e o comércio eletrônico no mundo e no Brasil foram tecidas. São ainda relacionadas as principais vantagens e algumas dificuldades enfrentadas pelo protocolo no contexto do comércio eletrônico mundial para que o protocolo se estabeleça como o padrão de uso universal em sua categoria. Na busca de se dispor de mais elementos que permitam avaliar a possibilidade de efetivação da tendência do SET se tornar um padrão, são sugeridos alguns temas para continuação do estudo.

8.1. Considerações Finais

8.1.1. O protocolo SET e o comércio eletrônico

Com a crescente expansão da Internet as empresas estão descobrindo que esse espaço não serve apenas para a divulgação, mas também para a comercialização de seus produtos. Segundo a Forrester Research Inc., em aneiro de 1998, o custo operacional por venda era de U$ 12.00 no balcão, de U$ 5.00 por telefone e U$ 1.00 pela Internet. As vendas *on-line* no varejo foram em torno de: um quarto de bilhão de U$ em 1996, 2 bilhões de U$ em 1997, de 5 bilhões de U$ em 1998, 7 bilhões de U$ em 1999, e em torno de 12 bilhões de U$ em 2000 [11].

Por ser um ambiente aberto e de acesso anônimo, a *Web* oferece riscos na transmissão e armazenamento de dados sigilosos, como números e senhas de cartões de crédito. A

preocupação com a segurança na transmissão de dados tem aumentado a medida que cresce o interesse das empresas nessa área de comércio eletrônico.

Para garantir a segurança de dados sigilosos envolvidos numa compra eletrônica, é necessária a criptografia desses dados. Essa técnica garante que apenas o emissor e o receptor tenham conhecimento das informações enviadas.

Outra questão está relacionada ao repúdio. Uma vez efetuada a venda, o comerciante deve ter a segurança de que o verdadeiro dono do cartão efetuou a compra e que esse não poderá negar mais tarde o pagamento da mesma.

O padrão SET (*Secure Eletronic Transactions*) surgiu em 1997, através da iniciativa de um consórcio liderado pelas empresas VISA e MasterCard em 1996, como uma alternativa para resolver esses principais problemas relacionados ao comércio eletrônico. O SET é um protocolo de segurança desenvolvido para garantir a transmissão segura de informações financeiras em redes públicas [10].

Para atender as necessidades de segurança o SET usa criptografia para:

- prover confidencialidade da informação,
- assegurar a integridade de pagamento, e
- autenticar tanto comerciantes como portadores de cartão.

A confidencialidade é assegurada pelo uso de criptografia nas suas mensagens. A proteção da informação sensível é feita pelo protocolo SET utilizando processos de criptografia de chave secreta e criptografia de chave pública. O algoritmo de criptografia simétrica ou de chave secreta utilizado pelo SET é o DES (*Data Encryption Standard*). A criptografia de chave pública, também conhecida como criptografia assimétrica, usa duas chaves: uma chave pública do destinatário para cifragem da mensagem e sua própria chave privada para decifragem da mensagem. O algoritmo criptográfico de chave pública utilizado pelo SET é o RSA.

A integridade e autenticação são asseguradas no SET pelo uso de assinaturas digitais. A autenticação num sistema digital é um processo por meio do qual o receptor de uma mensagem digital pode estar confiante da identidade do remetente e/ou da integridade da mensagem. Quando combinada com criptografia usando chave privada, o resumo de

mensagens permite a assinatura digital pelos usuários. A necessidade de assinaturas digitais surgiu da proliferação das comunicações digitais. Logo, a cifragem e autenticação acontecem sem compartilhamento de chaves secretas: cada pessoa usa apenas as chaves públicas de outras pessoas e sua própria chave privada.

É necessário um prazo de validade adequado de uma chave para prevenção contra tentativas de “quebra” a longo prazo. Logo, o tempo de validade deve ser muito menor do que o tempo esperado para que se consiga sua “quebra”, ou por outro lado, o comprimento da chave deve ser suficientemente grande para tornar pequenas as chances de se conseguir sua “quebra” antes do término de sua validade. A data de validade de uma chave acompanha a chave pública num certificado. O programa de verificação de assinatura deve verificar a validade da chave e não deve aceitar uma mensagem assinada por uma chave fora da validade.

A Certificação digital é uma aplicação na qual uma autoridade de certificação "assina" uma mensagem especial *m* contendo o nome de algum usuário *A* e sua chave pública, de forma que qualquer pessoa possa "verificar" que a mensagem foi assinada pela autoridade de certificação e assim há a confirmação do crédito da chave pública de *A*. Uma implementação típica da certificação digital envolve um algoritmo de assinatura para assinar uma mensagem em especial, utilizada pelos certificados da X.509. Com uma assinatura digital, qualquer um pode verificar a qualquer hora, que a certificação foi assinada pela autoridade de certificação, sem acesso a informação secreta.

A Lista de Revogação de Certificados, ou CRL, é outro tipo de mensagem especial com uma assinatura. A mensagem especial CRL contém uma lista de certificados revogados, onde os certificados são tipicamente referenciados indiretamente por um número de série. Uma CRL permite à autoridade da certificação "desabilitar" suas assinaturas no certificado da entidade *A* ou certificados estendidos, caso seja necessário quando o nome de *A* é alterado ou sua chave privada é comprometida.

De forma resumida, o processo de cifragem usado pelo protocolo SET consiste das seguintes etapas:

- a entidade emissora gera o resumo de mensagem através de função *hash* unidirecional (SHA-1).

- cifra o resumo de mensagem com sua chave privada de assinatura para produzir a assinatura digital.
- cifra no modo *DES*, usando uma chave secreta, a mensagem a ser enviada, a sua assinatura e cópia de seu certificado (que contém sua chave pública de assinatura), gerando os dados cifrados.
- cifra no modo RSA, usando uma chave pública do receptor (previamente obtida do certificado do receptor) a chave secreta usada para cifrar os dados, gerando o envelope digital.
- envia ao receptor os dados cifrados (mensagem mais assinatura digital mais seu certificado) juntamente com o envelope digital.

O processo de decifragem então consiste das outras etapas:

- a entidade receptora decifra o envelope digital com sua chave privada para recuperar a chave secreta.
- usa a chave secreta para decifrar a mensagem recebida.
- decifra a assinatura digital do emissor com a chave pública do assinante (obtida do certificado recebido do emissor), recuperando o resumo de mensagem original da mensagem.
- a entidade receptora gera o resumo de mensagem através da mesma função *hash* unidirecional usada pela entidade emissora.
- compara o resumo de mensagem gerado com o recebido, verificando a assinatura digital, e caso sejam iguais obtém a certeza da autenticidade e integridade da mensagem recebida.

O SET altera a maneira como os participantes de um sistema de pagamento interagem. Em uma transação face-a-face pormenorizada ou em uma transação de um pedido por *e-mail*, um processamento eletrônico inicia-se com o comerciante ou com uma instituição financeira que processa as autorizações dos processos de pagamentos e os pagamentos propriamente ditos. Contudo, em uma transação SET, o processamento eletrônico inicia-se com o portador do cartão.

Em um ambiente de comércio eletrônico, os consumidores e os compradores corporativos interagem com os comerciantes através de computadores pessoais. Um portador usa o cartão que tenha sido emitido por uma instituição financeira - emissor. O SET assegura que nas

interações dos portadores de cartão com os comerciantes, as informações da conta usadas no pagamento permanecem confidenciais.

O SET utiliza um recurso denominado assinatura dual. O propósito da assinatura dual é ligar duas mensagens que são endereçadas a dois destinatários diferentes. Neste caso, o cliente quer enviar a informação de compra para o comerciante e a informação de pagamento para o banco. O comerciante não precisa saber o número do cartão de crédito do cliente e o banco não precisa saber os detalhes da compra. Ao cliente é oferecida uma proteção extra em termos de privacidade mantendo estes dois itens separados, entretanto estes dois itens precisam estar ligados de tal forma que possam ser usados para resolver qualquer dúvida, ou seja, essa ligação é necessária para que o cliente possa comprovar que aquela ordem de pagamento é destinada aquele pedido e não a qualquer outro bem ou serviço.[9] [4].

8.1.2. Outros protocolos de segurança

A possibilidade do SET aumentar sua utilização tem sido ampliada porque os Comerciantes tem experimentado algumas deficiências em outras soluções para condução da segurança do comércio eletrônico. Com o aumento das compras eletrônicas e disponibilidade de sistemas mais sofisticados de pedidos *on-line*, a indústria de *software* tem reconhecido que os outros sistemas de segurança não tem atendido suficientemente as preocupações dos consumidores e comerciantes com relação a um ambiente de compras completamente seguro. Para clarificação deste fato, convém um breve exame de algumas outras ferramentas para segurança do comércio eletrônico, como o *Secure Socket Layer* (SSL), *Pretty Good Privacy* (PGP), *Secure/ Multiporpose Internet Mail Extensions* (S/MIME), a seguir apresentado.

Numa camada mais baixa, o protocolo SSL fica situado no topo de um protocolo de transporte confiável tal como o *Transmission Control Protocol* (TCP). O Protocolo de Registro SSL é usado para encapsular outros protocolos de nível mais alto. Um destes protocolos, o Protocolo de *Handshake* SSL, autentica reciprocamente o cliente e o servidor e os habilita decidir sobre o algoritmo de cifragem e chaves criptográficas a serem usados antes que protocolo de nível mais alto envie ou receba dados. O SSL endereça algumas das mesmas preocupações do SET. Seus objetivos são assegurar a privacidade de conexão, autenticar a identidade do par de usuários em conexão, e estabelecer um mecanismo confiável

de transporte para a mensagem, usando verificações de integridade e funções *hash*. O Protocolo SSL foi projetado para aplicações clientes/servidor, prevenindo intrusões não desejadas em transmissões de dados, alterações de dados, ou falsificação de mensagens. Um benefício do SSL é que ele permite que os protocolos de mais alto nível se situem em seu topo e se comunica com eles sem ditar um protocolo específico para aplicação. O SSL usa criptografia de chave simétrica para cifragem de dados e autentica a identidade de pares usando técnicas criptográficas assimétricas. Um relevante problema com o SSL reside no fato de que o trabalho de verificação dos certificados na cadeia deve ser feito totalmente pela CA, ao invés de pelo próprio usuário. Segundo os especialistas, alguns outros problemas do SSL são: promoção de trabalho computacional adicional ao cliente e ao servidor, aumento de tráfego na rede para implementar seu protocolo de comunicação, não trabalhar bem com *tokens* criptográficos existentes, ter gerenciamento de chaves SSL mais caro, requerer Autoridade de Certificação com políticas próprias para atender suas necessidades, ser lento na transmissão das comunicações cifradas em dispositivos como *modems*, por estas não poderem ser comprimidas e também possui restrições de exportações. Até que estes problemas sejam trabalhados, o SSL não pode fornecer o nível de segurança encontrada no protocolo SET. Uma grande vantagem do SET sobre o SSL se refere ao nível de proteção dos números de cartão de crédito. O SSL cifra os dados durante o trânsito na rede, assim os números de cartão de crédito enviados pela Internet estão protegidos. Ao serem recebidos, são armazenados pelo Comerciante sem qualquer tratamento criptográfico. Então, a ameaça a estas informações não se situa somente quando em trânsito, mas também existe quando estes números já foram enviados e se encontram armazenados. Os ataques ocorrem então aos servidores que armazenam a informação sigilosa. O melhor procedimento é cifrar e armazenar os números de cartões. Assim, quando se utiliza o SET, este procedimento é atendido, pois os números de cartão se encontram dentro do envelope digital, cuja chave pública que cria o envelope pertence ao Adquirente, assim o Comerciante não pode acessar os números dos cartões em texto claro, e por isso não poderá armazená-los de forma não segura.

O Pretty Good Privacy (PGP) possui uma abordagem de gerenciamento de chave distribuído que não conta com Autoridades de Certificação. Os usuários podem assinar uns as chaves públicas dos outros, adicionando graus de confiança a uma validação de chave. O usuário que assina a chave pública de um outro usuário atua como introdutor deste último usuário, com base na premissa de que se o receptor confia no introdutor, ele também confiaria naquele que está sendo introduzido. O PGP foi desenvolvido por Phil Zimmerman em meados dos anos

80. Sua popularidade se deve à sua habilidade de cifrar *e-mail*. Zimmerman distribuiu sua primeira versão do PGP sobre a Internet como *freeware*, quando incorreu em problemas legais referentemente aos direitos de patente da criptografia de chave pública utilizada (mais especificamente, a patente RSA). A situação foi legalizada em 1993 quando a ViaCrypt, uma companhia com uma licença válida para a patente, operou com Zimmerman na distribuição da versão comercial do PGP. O PGP seguiu os passos dos outros sistemas usados na cifragem de *e-mails*. Um sistema chamado SECURE/32, desenvolvido por Charlie Merrit, usou criptografia de chave pública. Imediatamente após, a RSA Data Security desenvolveu um sistema chamado MailSafe, um sistema de cifragem de *e-mail* mais robusto. O PGP usa a cifragem e decifragem, assinaturas digitais, e chaves criptográficas para proteger *e-mails*. Alguns especialistas afirmam que o PGP foi copiado do MailSafe, sem aquiescência de seu desenvolvedor. Outros declaram que ambos os sistemas, PGP e MailSafe, se originaram como cópias do SECURE/32. Uma das principais críticas ao PGP é que sua confiança é baseada numa cadeia de confiança informal, em lugar de uma hierarquia de confiança estruturada de fato, como a do SET. Os críticos reclamam que não se pode, por exemplo, tomar uma chave pública PGP de um usuário da Internet e obter-se a segurança que a chave pública realmente deveria possuir. Estas limitações fazem o uso do PGP impraticável na condução do comércio eletrônico na Internet.

Baseado na tecnologia da RSA Data Security, o *Secure/Multipurpose Internet Message Extensions* (S/MIME), um protocolo de segurança no nível de camada de aplicativo, oferece outro padrão para cifragem e assinaturas digitais de *e-mail*. Ambos, o S/MIME e a versão do PGP denominada Open PGP, são implementados nos *browsers* da Web da Netscape Communications. Estes dois padrões de cifragem de *e-mails* estão criando problemas de ambiguidade para os usuários, enquanto se discute qual o único padrão deveria ser adotado. Originariamente projetado para uso com *e-mails*, a aplicabilidade do S/MIME para as mesmas finalidades do SET é muito limitada. O S/MIME e Open PGP usam as técnicas da cifragem proprietárias e manuseiam assinaturas digitais diferentemente. Simplesmente, se o usuário A usa um *browser* dotado do S/MIME e tenta se comunicar com o usuário B, que usa um *browser* dotado do PGP, os dois usuários não obtém sucesso na comunicação. Outro problema do S/MIME, como também do PGP, provém do fato que eles trabalham bem em redes fechadas, mas não têm tal desempenho quando rodando em redes abertas, tal como a Internet, por causa de suas características de interoperabilidade [4]. O S/MIME originariamente foi desenvolvido para o uso com chaves de cifragem de 40 bits, o que o torna

inseguro atualmente. Em implementações mais recentes, em sua versão 3, o S/MIME pode operar entre diversos tipos de cifragem simétrica, sendo uma delas o *Tripple* DES, o que elevaria seu tamanho de chave para 168 bits. Para algoritmos assimétricos de assinatura e cifragem o S/MIME pode operar o RSA.

8.1.3. Principais aspectos positivos do protocolo SET

- O projeto da Versão 1.0 do SET é baseado nos padrões da indústria, da Internet, e das organizações internacionais como definidos na ISO, IETF, PKCS, e padrões ANSI, portanto é um padrão aberto não proprietário.
- Por ser um protocolo aberto se apresenta com grande possibilidade de se tornar o padrão no processo de transações financeiras pela *Web*.
- O nível de adaptabilidade do SET é grande. As especificações da Versão 1.0 já foram publicadas prevendo quaisquer futuras versões. O SET permite diferentes funções de negócios através do uso de extensões do protocolo. A Versão 1.0 do SET inclui as mínimas funcionalidades requeridas para apoiar os Portadores de Cartão e Comerciantes na Internet. Embora o SET não possa antecipar todas as práticas dos comerciantes de cada mercado nacional de cada Adquirente, o projeto permite os tipos de aberturas e flexibilidades necessárias para tais adequações. Por exemplo, o projeto permite um modo de estender as mensagens de pagamento do SET. No Japão, os fornecedores têm opções para pagamento que os consumidores selecionam na hora da compra. Enquanto as mensagens normais do SET não têm lugar para essa informação adicional na Versão 1.0, os projetistas do protocolo permitiram uma extensão para o protocolo para solucionar tal problema.
- Suas especificações e a utilização de assinaturas duais eliminam a necessidade de envio do número do cartão ao *site* de compra e as informações de compra não são enviadas à entidade financeira que autoriza a transação.
- Embora seja um protocolo relativamente novo, sua comercialização é atendida por grandes empresas, havendo grande variedade de produtos de *software* e *hardware* para suporte ao SET disponíveis no mercado.
- O consórcio SETCo que apoia seu desenvolvimento é liderado por grandes instituições financeiras de cartões de múltipla finalidade como VISA e MasterCard. Portanto, a

liderança destas empresas e de outras empresas líderes em seus ramos de negócios é uma forte base econômica e mercadológica para alavancagem da utilização do protocolo SET.

8.1.4. Algumas dificuldades relativas ao protocolo SET

- Embora não se conheça quais são os planos do Consórcio SET referentes ao *DES*, utilizado na atual versão do protocolo, que tem sido alvo de pesquisas de chave por busca exaustiva bem sucedidas desde final dos anos 90, há que se considerar que o caminho natural é a utilização do seu substituto, o AES [13]. O padrão AES, que foi publicado oficialmente em 2001, e é recomendado pelo NIST para adoção pelos órgãos oficiais americanos e para utilização de órgãos não oficiais, usa chaves de comprimento de 128, 192 e 256 bits para cifrar e decifrar blocos de dados de 128 bits. Assim, acredita-se que, provavelmente, a versão 2.0 já venha incorporá-lo.
- Apesar da arquitetura proposta pelo SET ser muito boa nos aspectos relacionados ao atendimento dos requisitos necessários para o processamento seguro de transações eletrônicas em redes abertas como a *Internet*, uma série de fatores atualmente ainda inibe a sua aceitação como a solução definitiva para os problemas de segurança. Um deles, talvez o maior, é a sua alta complexidade.
- Existem alguns problemas que ainda precisam ser superados para que o comércio eletrônico cresça em qualidade e uso. Um desses problemas é a conquista da confiança dos compradores. Muito ainda precisa ser falado e provado para que essa confiança seja realmente conquistada. Dentro desse aspecto, temos a falta de assistência técnica que tranqüilize e oriente o usuário em caso de erros e falhas de comunicação.[10].
- Um problema que tem surgido com a aceitação do protocolo SET diz respeito à política de segurança de alguns países do mundo. Na França, por exemplo, existe a restrição que toda mensagem cifrada deve ser possível de ser decifrada pelas autoridades governamentais.
- Nos Estados Unidos a política do governo de controle de desenvolvimento e uso de cripto-sistemas é vista por muitos como um sério obstáculo ao comercio eletrônico global. No cerne desta questão está a distinção feita entre cifragem “fraca” e “forte”. De forma simples, em referência à política de exportação que afeta o SET, a tecnologia de cifragem fraca usa o que é conhecido como chaves de cifragem de 40 bits, enquanto a cifragem forte usa chaves de 56 bits ou mais. Entretanto, no aspecto de efetiva segurança de cripto-

sistemas, desde o final dos anos 90 uma chave de 56 bits não era suficientemente forte para resistir a uma busca exaustiva, com resultado bem sucedido, por algumas horas. Atualmente, chaves para serem consideradas seguras devem possuir comprimento de 128 bits ou mais.

- Os europeus estão particularmente preocupados sobre a política americana para o comércio eletrônico global, nominadamente sua continuada proibição na exportação de cifragem forte. Alguns empresários de empresas européias acreditam que comércio eletrônico irá somente tornar-se uma realidade se os usuários tiverem absoluta segurança quando eles enviarem seus dados.

8.1.5. O protocolo SET no mundo e no Brasil

Nos Estados Unidos, o NationsBank de Charlotte, Carolina do Norte conduziu a primeira transação SET do Banco em 1998. O que foi significativo neste evento foi o uso pelo NationsBank das especificações do SET 1.0 e *software* de diferentes fornecedores de tecnologia, demonstrando completa interoperabilidade. O NationsBank trabalhou estreitamente com a IBM, MasterCard, GlobeSet e GTE para criar um sistema para compra de itens do Emporium MasterCard, um site *Web* construído pela associação para ajudar os consumidores a fazerem pequenas compras iniciais e superar seus medos de compras eletrônicas. O NationsBank usou o CommercePOINT Wallet da IBM e o IBM Registry para os serviços de certificado do SET. A MasterCard, trabalhando em conjunto com a GlobeSet, forneceu o *software* do Servidor do Comerciante e o *software* do Portal de Pagamento. Este é só um exemplo da interoperabilidade de que o SET é dotado.

No mundo o protocolo tem muitas outras iniciativas no sentido de torná-lo um efetivo padrão de segurança. A Europa atualmente está forjando uma iniciativa de comércio eletrônico, liberando suas forças tecnológicas e diversidade cultural para levantar o comércio da Internet. Seu sucesso potencial está baseado, em parte, no fato que o mercado europeu representa o maior mercado unificado do mundo, sustentado pela aceitação de uma única moeda, o Euro. Os membros da iniciativa sentem que não somente o Euro irá sustentar o *e-commerce*, mas reciprocamente, o *e-commerce* irá sustentar o Euro.[4]

Para entender a importância do protocolo SET, em particular, e do comércio eletrônico em geral, pode-se simplesmente olhar para a Holanda, um dos mais desenvolvidos mercados de comércio eletrônico no mundo. De acordo com a International Data Corp. (IDC), a Holanda tem uma alta penetração de Computadores Pessoais (38 % das casas), alto uso de Internet (22 % dos usuários de PC tem acesso a Internet comparados com 16% nos EUA e 12% na Alemanha), e o alto uso de comércio eletrônico (33 % de usuários da Internet usam-na para comprar *on-line* versus 22% nos EUA)[4].

Os dois maiores parceiros no mercado do *e-commerce* da Ásia, Japão e China, tem desenvolvido sistemas incompatíveis com o SET, embora o Japão esteja trabalhando para fazer suas especificações JSET (Japan SET) compatíveis com a Versão 1.0 do SET. O JSET está evoluindo por um par de razões: Primeiro, o comércio de cartão de crédito no Japão é regulado pelo Ministério de Comércio Internacional e Indústria em vez do Ministério das Finanças; Segundo, o Japão usa um algoritmo de segurança para o comércio de cartão de crédito diferente do de seu comércio interbancário. A China atualmente não tem padrão interbancário no todo, prova adicional da necessidade de um padrão internacional. Por exemplo, um cliente do Banco da China em Beijing não pode usar seu cartão eletrônico na filial do Banco da China em Shanghai, muito menos um cartão de outro banco. O projeto Golden Card em Shangai, uma *joint venture* de comércio eletrônico da China entre o Pu Dong Development Bank e a Bull da França, começou há mais de sete anos atrás e atualmente tem 350.000 cartões de transação CP8 em circulação. Este sistema usa o *software* de compensação interbancária proprietário da Pu Dong, que é incompatível com o SET. Isto tem feito a Visa International informar que está negociando com diversos bancos na China para introduzir um sistema SET/EMV (EMV - Europay, MasterCard e Visa) na China. A Visa informa também que seu plano em conjunto com a MasterCard é que este sistema seja internacionalmente universal, não proprietário.

O termo EMV é internacionalmente conhecido como a referência dos *SmartCard* para os cartões das companhias “Europay, Mastercard e Visa”. Um *SmartCard* se assemelha a um cartão de pagamento de plástico convencional, porém também contém um *chip* semicondutor com memória lógica e não volátil. Pode ser comparado a um dispositivo de depósito eletrônico seguro, o cartão armazena um *software* que detecta manuseio ou intrusões não autorizadas. Milhões destes cartões estão nas carteiras dos usuários na Ásia e Europa, e eles estão vagarosamente ganhando popularidade nos Estados Unidos. A referência EMV

contempla, entre outras, as seguintes características: acesso controlado aos *softwares* de pagamento do SET e a familiaridade e praticidade de um cartão de plástico em conjunto com os altos níveis de segurança do SET.

No Brasil, o resultado da pesquisa de 1998 do IBOPE em conjunto com o *site* Cadê? mostra que em torno de 15% dos usuários da Internet a utilizam para realizar compras eletronicamente. Esta quantidade é menos que a metade dos 33% de usuários da Holanda e aproximadamente três quartos dos 22% de americanos que se utilizam da Internet para fazer compras. Talvez o baixo percentual de compradores no Brasil se deva aos fatores elencados na mesma pesquisa como principais problemas na Internet, onde 25% dos usuários brasileiros informa que não consegue organizar as informações acessadas, 16% diz ficar perdido na Internet, 27 % acha demorado carregar páginas. A demora nas linhas telefônicas brasileiras e a falta de segurança na estruturação da Internet são os principais fatores que desmotivam consumidores da *Web* [10].

A despeito desta situação, há um crescimento substancial do uso do comércio eletrônico e consequentemente do SET no Brasil. Como exemplo deste crescimento, vamos abordar a primeira iniciativa nacional que foi o Bradesco Net – Internet Banking (implantado em 1996), que foi também o quinto site do mundo a disponibilizar aos seus clientes a possibilidade de realizar diversas operações bancárias pela Internet, com total segurança. O serviço foi uma das primeiras iniciativas mundiais de implementação do protocolo SET para pagamento em compras eletrônicas. O Bradesco Net foi inaugurado em março de 1998 no endereço eletrônico “www.bradesco.com.br/comércio”. O *shopping* foi construído no período de um ano, teve um custo de 700 mil dólares e nasceu com 13 lojas conveniadas. Segundo informações divulgadas no Seminário Compras Governamentais, em setembro de 1998 em Brasília, por representantes do Bradesco, as lojas conveniadas ao Bradesco Net já eram na ordem de 28 [11]. Em 2002 o mesmo serviço continua existindo, com o nome de ShopFácil, e já possui atualmente mais de 600 lojas conveniadas, além de outros serviços como acesso a bancos, páginas de ofertas, sistema de busca e comparação de preços, etc.

O ShopFácil é um serviço da Scopus, uma empresa do Banco Bradesco S/A. O *site* do Bradesco é um hospedeiro de lojas que queiram realizar vendas pela *Web* com total segurança nas compras efetuadas. O Bradesco não oferece produtos, ele fornece a infra-estrutura de um sistema de pagamentos pela Internet. As lojas interessadas assinam um convênio com o Bradesco Net e assim podem receber pagamentos eletrônicos efetuados com os cartões de

crédito, débito e/ou de poupança do banco. A principal característica desse *shopping* é a utilização das carteira eletrônicas, um conceito novo introduzido na especificação do protocolo SET. O sistema cobre os pontos mais críticos do comércio eletrônico: sigilo, autenticação (através dos certificados digitais), integridade, e não repúdio. Resumidamente, a compra virtual funciona da seguinte forma: ao fazer uma compra pela Internet numa loja conveniada, o cliente utiliza a carteira eletrônica, que ele mesmo instala no micro ou mantém em disquete. Ao concretizar a operação é emitido um recibo com o número da operação e a mercadoria é enviada pelo lojista. A *home-page* da loja conveniada pode estar hospedada num domínio próprio ou dentro do domínio do Bradesco Net. Caso a loja tenha um domínio próprio, na hora do pagamento, é feito o *link* para o *back-end* do Banco [10].

8.1.6. Perspectivas para o futuro do protocolo SET

A Criptografia baseada em Curvas Elípticas (*Elliptic Curve Criptography – ECC*), desenvolvida a partir de 1985, usa o sistema algébrico definido sobre os pontos de uma curva elíptica para criar os algoritmos de chaves públicas. Matemáticos têm estudado este tipo de sistema por muitos anos. Em 1985, Neal Koblitz e V. S. Miller propuseram o uso de curvas elípticas para cripto-sistemas de chave pública. Os patrocinadores do SET estão considerando o uso de ECC nas especificações do SET para a versão 2.0 por diversas razões. Alguns especialistas estão muito preocupados porque a atual implementação do SET requer grandes velocidades de processamento o que deve prejudicar sua aceitação. De forma simples, a ECC que têm sido testada com chaves com comprimento de 79, 89, 97, 109, 131, 163, 191, 239 e 359, produz forte segurança com chaves de comprimentos superiores a 108 bits, comparada às chaves de 512 bits e 1024 bits que são usadas atualmente pelo RSA no SET. Entretanto, o menor comprimento recomendado para operações comerciais é 132 bits. Os especialistas alegam que a principal vantagem da ECC é a utilização de chaves menores que rendem cifragem forte, mas com velocidades de processamento muito maior. A ECC parece muito benéfica onde o poder computacional é baixo (isto é, *SmartCards*, *PC Cards* e dispositivos *wireless*), onde o espaço de memória é restritivo, onde as necessidades de maiores velocidades de processamento podem ser atendidas e onde a largura de banda é restrita (isto é, comunicações sem fio). O potencial do uso da ECC na versão 2.0 do SET tem sido considerado seriamente pelos especialistas da indústria.

Atualmente 27 milhões de cartões de banco circulam na França. Alguns especialistas vêem os SmartCards como uma maneira de deslocar os consumidores dos métodos tradicionais de pagamento para o comércio eletrônico. O SET, juntamente com o padrão EMV, parece que será o padrão na França para as transações de pagamento, combinando o melhor da autenticação do consumidor, a familiaridade com o próprio cartão de plástico e a segurança incrementada do SET[4].

8.2. Conclusões

8.2.1. O que é necessário para que o SET seja o padrão universal

A variedade de produtos a venda na *Web* é enorme. Qualquer produto que não precise ser experimentado, cheirado ou apalpado e que não seja perecível, é um sério candidato à venda virtual. Alguns serviços, como por exemplo, educação à distância também entra nessa lista. Além disso, técnicas de marketing personalizado e direcionado ao consumidor alvo e novas formas de apresentação dos produtos, entre outros fatores, também estão na lista dos pontos de evolução dos *shoppings* virtuais. Tudo isso busca a simpatia e a comodidade dos compradores para essa nova forma de se realizar o comércio. Entretanto, uma coisa é inquestionável: a evolução do comércio eletrônica pela Web está diretamente ligada a evolução das técnicas que garantam a segurança na transmissão e armazenamento de dados sigilosos pela rede.

A utilização do protocolo SET torna o comércio eletrônico mais eficiente e seguro. Assim, o sucesso do comércio eletrônico global depende da aceitação não forçada do SET. Muitos fatores ainda irão determinar se a Internet se move para além da “economia de presentes”, como alguns céticos clamam, para se transformar no mercado eletrônico internacional. Porém ainda é necessário que alguns pontos sejam resolvidos para que o SET se consolide como um protocolo de segurança padrão universal, tais como:

- Há que se estabelecer acordos regulatórios internacionais com a definição clara de qual a interferência que podem ter os governos na questão do comércio eletrônico dentro e fora das fronteiras de seus países.

- Deve haver acordos internacionais de interoperabilidade para definição de padrões de rede e cifragem internos e externos aos países.
- Para que a Internet se torne o século 21 agora, negócios e tecnologias líderes devem acordar sobre os padrões de negócios de comércio eletrônico a serem estabelecidos por cada país.
- Assim sendo, para criar redes de comércio eletrônico, os comerciantes e bancos devem poder escolher ferramentas de cifragem e protocolos de um grande número de vendedores, mas devem levar em conta as políticas de cifragem governamentais. O SET é considerado como uma destas ferramentas com potencial para realizar tal interoperabilidade em função de sua habilidade nativa em atender padrões externos.
- A consolidação das tecnologias para o comércio eletrônico passa pela redução da inadimplência de usuários de cartões de crédito, com vistas a alavancar o investimento das operadoras para liberarem mais rapidamente serviços de processamento *on-line* dos cartões.
- Os desenvolvedores do SET devem se esforçar no sentido de simplificação de uso do protocolo com o objetivo de atender o clamor de muitos comerciantes que são unânimes em afirmar que é muito complicado para o usuário utilizar o protocolo (todo o processo de instalação e registro de carteiras eletrônicas, memorização de senhas, entender o funcionamento). O ideal é que o SET fosse mais simples de ser utilizado para que sua popularidade aumentasse junto ao usuário da Internet.[8]

Finalmente, com base no exame detalhado de toda a criptografia utilizada pelo Protocolo SET e por todas as informações de seu ambiente contextualizador, ou seja, proteção da informação, Internet e comércio eletrônico, aqui abordadas, podemos concluir que há mais vantagens reais e potenciais no uso do protocolo SET de que desvantagens. As desvantagens, em sua maioria, se situam no âmbito externo ao protocolo, ou seja, nas definições das políticas governamentais e dos próprios negócios. No aspecto interno, ou seja, quanto a sua complexidade, alegada como dificuldade intrínseca do protocolo, acredita-se que, em consequência da atualização tecnológica transcorrida entre sua primeira versão e a próxima a ser publicada, há grande possibilidade de redução, tornando-o mais amigável aos usuários finais e possivelmente reduzindo a demanda de esforços de seus desenvolvedores de produto.

Assim sendo, torna-se claro que o SET, melhor que qualquer outro padrão estabelecido, reúne a maioria das condições necessárias para executar o principal papel no apoio ao comércio

eletrônico global, que é prover segurança das informações em transações de comércio eletrônico sobre a Internet. Entretanto, com base nas razões já expostas, mormente quanto a impossibilidade de se predizer com exatidão como se dará o futuro do comércio eletrônico global, ainda não se pode precisar quando e de que forma o SET se estabelecerá como o protocolo padrão de segurança universal em suporte ao comércio eletrônico.

8.2.2. Sugestões para novas pesquisas sobre o Protocolo SET

Com objetivo de obter-se informações complementares para análise e a confirmação da tendência exposta ao longo deste trabalho, de que o SET efetivamente se tornará o padrão como protocolo de segurança, é importante a promoção de novos estudos, sempre comparativamente aos demais protocolos disponíveis no mercado, iniciando-se com a avaliação da sua real utilização, através da reunião de dados existentes e pesquisas junto aos *sites* de *e-commerce* no mundo e no Brasil, seguindo-se ainda pelas seguintes sugestões:

- Avaliação qualitativa sobre os aspectos do SET consider ados importantes pelos consumidores, comerciantes, instituições financeiras e companhias de cartão, tais como: dificuldades, facilidades, sugestões, consumo de recursos, custo-benefício da implementação, desempenho e outros.

- Avaliação do SET no *e-commerce* baseado em *SmartCards*.

- Avaliação das efetivas melhorias em planejamento pelo Consórcio SET para próxima versão e situação atual destas melhorias.

- Avaliação dos impactos na consolidação do SET como padrão com adoção da criptografia baseada em AES e ECC.

- Análise técnica comparativa entre o SET e demais protocolos de segurança.

- Avaliação comparativa do consumo de recursos de processamento pelo SET e demais protocolos.

Apêndice A

Exemplo da Aplicação do Algoritmo DES

Vamos cifrar o seguinte texto claro = 0123456789*ABCDEF* . Sua expansão binária é a seguinte p :

Posicionamento dos bits de p							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	54

Expansão binária de p							
0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

A aplicação de IP resulta em

Posição binária após permutação IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

$IP(p)$							
1	1	0	0	1	1	0	0
0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0
1	1	1	1	0	0	0	0
1	1	1	0	1	0	1	0

Da leitura do quadro de $IP(p)$ obtemos:

$L_0 = 11001100000000001100110011111111$

$R_0 = 11110000101010101111000011101010$

Usamos a chave DES:

$K = 133457799BBCDFF1$

cuja expansão binária é

Posicionamento de bits da chave K							
1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Expansão binária de K							
0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

Agora vamos computar as chaves de iteração K_i , onde $K_i = PC2(C_i, D_i)$. Primeiramente aplicaremos a função $PC1$ em K, onde após o mapeamento que reduz o comprimento de K de 64 bits para 56 , retiramos os valores de C_0 e D_0, conforme quadros abaixo.

PC1 – Mapeamento de K						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Computação de C_0 e D_0							
1	1	1	1	0	0	0	C_0
0	1	1	0	0	1	1	
0	0	1	0	1	0	1	
0	1	0	1	1	1	1	
0	1	0	1	0	1	0	D_0
1	0	1	1	0	0	1	
1	0	0	1	1	1	1	
0	0	0	1	1	1	1	

C_0 = 11110000110011001010101111

D_0 = 01010101011001100111100011111

Aplicando um deslocamento circular à esquerda $(v_1 = 1)$ em C_0 e D_0 obtem-se respectivamente C_1 e D_1.

C_1 = 11100001100110010101011111

D_1 = 10101010110011001111100011110

Agora aplicamos a função $PC2$ em C_1 e D_1 para obtenção de K_1, sendo $K_i = PC2(C_i, D_i)$ conforme quadros abaixo.

Par (C_i, D_i)						
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49
50	51	52	53	54	55	56

$PC2$ – **Mapeamento de Par** (C_i, D_i) **para** K_i						
14	17	11	24	1	5	Mapeamento de um par (C, D) de 58 bits em cadeia de comprimento de 48 bits
3	28	15	6	21	10	
23	19	12	4	26	8	
16	7	27	20	13	2	
41	52	31	37	47	55	
30	40	51	45	33	48	
44	49	39	56	34	53	
46	42	50	36	29	32	

A operação anterior fornece o seguinte resultado:

Par (C_r, D_r)						
1	1	1	0	0	0	0
1	1	0	0	1	1	0
0	1	0	1	0	1	0
1	0	1	1	1	1	1
1	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	1	1	1	1	0
0	0	1	1	1	1	0

K_1						
0	0	0	1	1	0	Mapeamento de um par (C_1, D_1) para formação de K_1 de 48 bits de comprimento
1	1	0	0	0	0	
0	0	1	0	1	1	
1	0	1	1	1	1	
1	1	1	1	1	1	
0	0	0	1	1	1	
0	0	0	0	0	1	
1	1	0	0	1	0	

Então:

K_1 = 000110110000001011101111111111000111000001110010

Aplicamos a função Expansão em R_0, definida nos quadros de posicionamento de bits abaixo.

R			
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32

$E(R)$					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Sendo R_0 = 11110000101010101111000010101010,

R_0			
1	1	1	1
0	0	0	0
1	0	1	0
1	0	1	0
1	1	1	1
0	0	0	0
1	0	1	0
1	0	1	0

$E(R_0)$					
0	1	1	1	1	0
1	0	0	0	0	1
0	1	0	1	0	1
0	1	0	1	0	1
0	1	1	1	1	0
1	0	0	0	0	1
0	1	0	1	0	1
0	1	0	1	0	1

Dos quadros acima fazemos a leitura do resultado da Expansão:

$E(R_0)$ = 011110100001010101010101011110100001010101010101

Usando a chave K_1, obtida anteriormente calculamos $E(R_0) \oplus K_1$

$E(R_0)$ = 011110100001010101010101011110100001010101010101

K_1 = 000110110000001011101111111111000111000001110010

$E(R_0) \oplus K_1$ = 011000010001011110111010100001100110010100100111

Agora devemos fazer a operação dos S-boxes

$E(R_0) \oplus K_1 = B_1 B\ B_3 B_4 B_5 B_6 B_7 B_8$, então:

$B_1 = 011000$, $B_2 = 010001$, $B_3 = 011110$, $B_4 = 111010$, $B_5 = 100001$,

$B_6 = 100110$, $B_7 = 010100$, $B_8 = 100111$

Aplicando-se a operação S-box definida na Tabela 2.1 do capítulo 2, obtemos $C = C_1 C\ C_3 C_4 C_5 C_6 C_7 C_8$, onde :

C = 0101 1100 1000 0010 1011 0101 1001 0111

Posicionamento bits da cadeia C			
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32

Permutação Final P			
16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Posicionamento bits da cadeia C			
0	1	0	1
1	1	0	0
1	0	0	0
0	0	1	0
1	0	1	1
0	1	0	1
1	0	0	1
0	1	1	1

Permutação Final P			
0	0	1	0
0	0	1	1
0	1	0	0
1	0	1	0
1	0	1	0
1	0	0	1
1	0	1	1
1	0	1	1

$E_{K1}(R_0) =$ 0010 0011 0100 1010 1010 1001 $\rightarrow$ 1011 1011

Da expressão $(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus (f_{kr}(R_{i-1})), \quad 1 \le i \le r$, vem que :

$R_i = L_{i-1} \oplus (f_{ki}(R_{i-1}))$, então

$R_1 = L_0 \oplus f\ (R_0, K_1)$

$f_{k1}(R_0) =$ 0010 0011 0100 1010 1010 1001 1011 1011

$L_0 =$ 1100 1100 0000 0000 1100 1100 1111 1111

$R_1 =$ 1110 1111 0100 1010 0110 0101 0100 0100

Finalmente a palavra de 64 bits resultado da primeira iteração é , também de acordo com a fórmula 6.2.1., $(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus (f_{kr}(R_{i-1})), 1 \le i \le r$

$L_i = R_0$ = 11110000101010101111000011101010, assim:

$(L1, R1) = 11110000101010101111000011101010, 11101111010010100110010101000100$

As outras iterações são computadas de maneira análoga, exceto a última com a chave K_{16}, que está sujeita à uma inversão de metades, ficando as metades resultantes assim definidas:

$$R_{16} = L_{15} \oplus f_{k16}(R_{15}) \text{ e } L_{16} = R_{15}$$

Apêndice B

Acrônimos

ADB	*Actual Data Block* (OAEP)
ANSI	*American National Standards Institute*
API	*Application Programming Interface*
ASCII	*American Standard Code for Information Interchange*
ASN.1	*Abstract Syntax Notation One*
AVS	*Address Verification Service*
BC	*Block Contents* (OAEP)
BCA	*Brand Certificate Authority*
BCI	*Brand CRL Identifier*
BER	*Basic Encoding Rules*
BIN	*Bank Identification Number*
BT	*Block Type* (OAEP)
C	*Cardholder*
CA	*Certificate Authority*
CBC	*Cipher Block Chaining*
CCA	*Cardholder Certificate Authority*
CD-ROM	*Compact Disk Read Only Memory*
CDMF	*Commercial Data Masking Facility*
C*n*	Certificado para enésima geração de assinatura de chave raiz
CRL	*Certificate Revocation List*
DB	*Data Block* (OAEP)
DD	*Digested Data*
DEA	*Data Encryption Algorithm*
DEK	*Data Encryption Key (OAEP)*
DER	*Distinguished Encoding Rules*
DES	*Data Encryption Standard*

E	Operador de Cifragem Assimétrica
ECC	*Elliptic Curve Criptography*
EE	*End Entity: Cardholder (C), Merchant (M), or Payment Gateway (P)*
EH	Operador de Cifragem com Integridade
EK	Operador de Cifragem com uma chave fornecida
EMV	*Europay, MasterCard, Visa*
Enc	Encapsulação Simples com assinatura
EncB	Encapsulação Simples com assinatura e bagagem cifrada
EncBX	Encapsulação Extra com assinatura e bagagem cifrada
EncK	Encapsulação Simples com assinatura com chave fornecida
EncX	Encapsulação Extra com assinatura
E-Salt	Um recente, *salt* aleatório (OAEP)
EX	Operador de Cifragem Extra
EXH	Operador de Cifragem Extra com integridade
FIPS PUB	*Federal Information Processing Standards Publication*
GCA	*Geo-political Certificate Authority*
H	Operador de *hash* SHA-1
H*n*	*Hash* (SHA-1) de Certificado para enésima geração de Chave de Assinatura Raiz
H1	Operador de *Hash* #1 para o OAEP (retorna primeiros bytes)
H2	Operador de *Hash* #2 para o OAEP (retorna os bytes seguintes)
HDC	*Host Data Capture*
HMAC	*Keyed Hashing Message Authentication Code*
HTML	*Hyper-Text Markup Language*
HTTP	*Hyper-Text Transfer Protocol*
IEC	*International Electrotechnical Commission*
IETF	*Internet Engineering Task Force*
IIC	*Institution Identification Code*
IIN	*Institution Identification Number*
IP	*Internet Protocol*
ISO	*International Organization for Standardization*
ITU	*International Telecommunication Union*
L	Operador de ligação

M	*Merchant* (Comerciante)
MAC	*Message Authentication Code*
MCA	*Merchant Certificate Authority*
MCC	*Merchant Category Code*
MIME	*Multipurpose Internet Message Extensions*
MOTO	*Mail Order/Telephone Order*
NIST	*National Institute of Standards and Technology*
NSA	*National Security Agency*
OAEP	*Optimal Asymmetric Encryption Padding*
OD	*Order Description: dados de venda trocados entre o Portador de cartão e o Comerciante*
OI	*Order Instruction (*ou *Information)*
OID	*Object Identifier*
OLE	*Object Linking and Embedding*
OSI	*Open Systems Interconnect*
P	*Payment Gateway (Portal de Pagamento)*
PAN	*Primary Account Number*
PCA	*Payment Gateway Certificate Authority*
PDB	*Padded Data Block (OAEP)*
PGWY	*Payment Gateway*
PI	*Payment Instruction (*or *Information)*
PIN	*Personal Identification Number*
PKCS	*Public Key Cryptography Standards*
PK-E	*Public Key for Encryption*
PK-S	*Public Key for Signature*
POS	*Point of Sale*
RA	*Registration Authority*
R*n*	*Chave de Assinatura Raiz # n*
RCA	*Root Certificate Authority*
RRPID	*Request/Response Pair Identifier*
RSA	*Rivest Shamir Adleman* (RSA é um cripto-sistema de chave pública)
RSADSI	*RSA Data Security Incorporated*
S	Operador de Mensagem assinada

SET	*Secure Electronic Transaction*
SHA-1	*Secure Hash Algorithm - Revision 1*
SMTP	*Simple Mail Transfer Protocol*
SO	Operador de Somente Assinatura
TBE	*To Be Enveloped*
TBL	*To Be Linked*
TBS	*To Be Signed*
TCP	*Transmission Control Protocol*
TCAP	*Transaction Capabilities Procedures*
TDC	*Terminal Data Capture*
URL	*Universal Resource Locator*
V	*Verification String (OAEP)*
W3C	*World Wide Web Consortium*
WWW	*World Wide Web*
XOR	Operação “ou-exclusivo” de bit

Apêndice C

Glossário

Adquirente	A instituição financeira (ou seu agente) que adquire do aceitador de cartão os dados financeiros relativo à transação e inicia aqueles dados em um sistema de intercâmbio.
Ameaça	Uma condição que pode causar que a informação ou os recursos de processamento da informação sejam intencionalmente ou acidentalmente perdidos, modificados, expostos, tornados inacessíveis, ou em último caso afetados em detrimento da instituição.
Arranjo	Um agrupamento lógico de campos ou estruturas de dados que podem ser repetidos múltiplas vezes em uma mensagem.
Arranjo Ordenado	Um agrupamento lógico de campos ou dados estruturados que podem ser repetidos múltiplas vezes em uma mensagem e para a qual o ordenamento relativo de cada ocorrência é significativo.
Assinatura Digital	Informação digital cifrada com a chave privada de uma entidade que é anexada a uma mensagem para assegurar ao recipiente a autenticidade e integridade da mensagem. A assinatura digital prova que a mensagem foi assinada pela entidade que tem acesso à chave privada.
Assinatura Dual	Uma assinatura digital que cobre duas ou mais estruturas de dados pela inclusão de resumos seguros para cada estrutura de dados em único bloco cifrado .
Ataque de dicionário	Um ataque criptográfico onde o atacante constrói um dicionário pela cifragem de dados conhecidos com todas possíveis chaves de forma que a chave de alguma determinada mensagem possa ser obtida observando-se os dados cifrados no dicionário.
Ataque de precomputação	Um ataque onde o adversário precomputa uma tabela usada para quebrar a cifragem ou senhas.
Ataque de Repetição	Um ataque no qual uma mensagem válida é repetida por um adversário que atua como o originador. Então, a mensagem repetida parece válida, com exceção de dados de refrescamento (*nonces*)
Ataque de Substituição	Um ataque no qual o atacante substitui uma mensagem por outra.

Autenticação	O processo que busca validar a identidade ou provar a integridade da informação. A Autenticação em sistemas de chave pública usa assinaturas digitais.
Autoridade de Certificação	Uma entidade confiada por um ou mais usuários para criar e assinar certificados. É uma espécie de cartório eletrônico.
Autoridade de Registro	Uma organização de terceira parte independente que processa aplicações de cartão de pagamento para múltiplas marcas de cartão e aplicações diretas para as instituições financeiras apropriadas.
Autorização	O processo pelo qual uma pessoa apropriadamente designada ou pessoas com concessão, executam alguma ação em nome de uma organização. Este processo avalia o risco da transação, confirma que uma determinada transação não aumenta a dívida do possuidor da conta acima do limite de crédito da conta, e reserva a quantia especificada de crédito. (Quando um comerciante obtém autorização, o pagamento da quantia autorizada está garantido - contanto, obviamente, que o comerciante tenha seguido as regras associadas com o processo de autorização.)
Autorização de Reversão	Uma transação enviada quando uma autorização prévia precisa ser cancelada (uma reversão completa) ou diminuida (uma reversão parcial). Uma reversão parcial contém um campo adicional, a quantia substituta que será menos que a quantia autorizada. Uma reversão completa será usada quando a transação não pode ser completada, como quando o portador de cartão cancela o pedido ou o comerciante descobre que os bens já não estão mais disponíveis (descontinuados). Uma reversão parcial será usada quando a autorização era para o pedido inteiro e alguns dos bem não podem ser remetidos (resultando em uma remessa parcial).
Bagagem	Um termo que denota uma tupla codificada obscurecida, que é incluída em uma mensagem do SET porém anexada aos dados externos do encapsulamento de dados do PKCS. Isto evita a super cifragem da tupla previamente cifrada, mas garante o encadeamento com a porção da mensagem do PKCS .
Browser	*Software* que é executado no sistema de processamento do portador de cartão que provê uma interface para as redes públicas de dados .
Cadeia de Certificado	Um arranjo ordenado de certificados digitais, incluindo o certificado Raiz que é usado para validar um certificado específico.
Cadeia de Confiança	Um sinônimo para cadeia de certificado.
Captura	Uma transação enviada depois que o comerciante tenha remetido os bens. Esta transação ativará o movimento de capitais do Emissor para o Adquirente e então para a conta do comerciante.

Cartão de Pagamento	Um termo usado pelo SET para se referir coletivamente a cartões de crédito, cartões de débito, e cartões de banco emitidos por uma instituição financeira e que refletem uma relação entre o portador de cartão e a instituição financeira.
Certificação	O processo de averiguação se um conjunto de exigências ou critérios foi cumprido e atestamento deste fato a outros, normalmente por algum instrumento escrito. Um sistema que tenha sido inspecionado e avaliado como completamente obediente ao protocolo SET pelas partes e processos devidamente autorizados, deverá ser dito ter sido certificado complacente.
Certificado	Um tipo especial de estrutura de dados assinado digitalmente que contém informação sobre uma chave pública e do proprietário da chave pública. O SET define os seguintes tipos de certificados: assinatura, chave de cifragem , assinatura de certificado, e assinatura de CRL.
Certificado de Chave Pública	Chave pública e dados de identificação assinados por uma autoridade de certificação para prover autenticação e integridade da chave.
Certificado Raiz	O certificado no topo da hierarquia de certificados.
Chave Criptográfica	Um valor que é usado para controlar um processo criptográfico, como cifragem ou autenticação. O conhecimento de uma chave apropriada permite a correta decifragem ou validação de uma mensagem.
Chave de sessão	Uma chave de cripto-sistemas simétricos que é usada ao longo da duração de uma mensagem ou durante uma sessão de comunicação.
Chave Privada	Uma chave criptográfica usada com um algoritmo criptográfico de chave pública, exclusivamente associada a uma entidade e não tornada pública. Esta chave é usada criar assinaturas digitais ou decifrar mensagens ou arquivos.
Chave Pública	Uma chave criptográfica usada com um algoritmo criptográfico de chave pública e disponibilizada publicamente. É usada para verificar assinaturas que foram criadas com a chave privada casada. Chaves públicas também são usadas para cifrar mernsagens ou arquivos que podem ser decifrados somente usando a chave privada casada.
Cifra de Feistel	Uma classe especial de cifra de blocos iterados onde o texto cifrado é calculado à partir de um texto claro pela aplicação repetida de uma mesma transformação.
Cifragem	O processo de conversão da informação para torná-la numa forma ininteligível para todos exceto aos possuidores de uma chave criptográfica específica. O uso de cifragem protege a informação entre o processo de cifragem e o processo de decifragem (o inverso da cifragem), contra a revelação sem autorização.

Código de Autenticação de Mensagem (*Message Authentication Code* – MAC)	O código, anexado a uma mensagem pelo remetente que é o resultado do processamento da mensagem por um processo criptográfico. Se o receptor pode gerar o mesmo código, obtem-se confiança de que a mensagem não foi modificada e que foi originada pelo possuidor da chave criptográfica apropriada .
Comerciante	Um vendedor de bens, serviços, e/ou informações que aceita pagamento por itens eletronicamente. O comerciante também pode prover serviços de venda e/ou entrega de itens de venda eletronicamente.
Comércio Eletrônico	A troca de bens e serviços por pagamento entre o portador de cartão e o comerciante quando algumas ou todas as transações são executados por comunicação eletrônica.
Confidencialidade	A proteção da informação sensível e pessoal contra ataques não intencionais e intencionais e contra revelação.
Crédito	Uma transação enviada quando o comerciante precisa retornar valor ao portador de cartão (pelo Adquirente ou Emissor) em seguida a uma mensagem de captura válida, quando, por exemplo, foram devolvidos bens ou estavam defeituosos.
Criptoanálise	A arte e ciência de “quebrar” a cifragem ou alguma forma de criptografia.
Criptografia de Chave Pública	Um campo da criptografia inventado em 1976 por Whitfield Diffie e Martin Hellman. A criptografia de chave pública depende de um par casado de chaves inversas. A informação cifrada com uma chave só pode ser decifrada com a outra. Esta chave pública proporciona a um usuário a facilidade para cifrar dados .
Criptoperíodo	O espaço de tempo durante o qual uma chave específica é autorizada para uso ou quais chaves para um determinado sistema devem permanecer efetivas.
Cripto-sistema	Um algoritmo de cifragem-decifragem (cifra), junto com todos possíveis textos claros , textos cifrados e chaves.
Criptografia	O processo usado para cifragem ou autenticação da informação. A disciplina que encarna princípios, meios, e métodos para a transformação de dados de forma a esconder seu conteúdo de informação, previne sua modificação não detectada e uso sem autorização, ou uma combinação delas.
Destruição da Informação	Qualquer condição que faz a informação não utilizável, a despeito da causa.
Emissor	A instituição financeira ou seu agente que emite o número de conta primário (*Primary Account Number* – PAN) único para o Portador de cartão da marca de cartão de pagamento.

Envelope Digital	Uma técnica criptográfica para cifrar dados e enviar a chave de cifragem junto com os dados. Geralmente, um algoritmo simétrico é usado para cifrar os dados e um algoritmo assimétrico é usado para cifrar a chave de cifragem.
Envoltória de Messagem	Um conjunto comum de elementos de dados que são pre-afixados a cada mensagem do SET para identificar a versão particular do protocolo, revisão, data/hora, identificadores de transação, e identificador de par de pedido/resposta (RRPID) para este ciclo.
Espalhamento	Um uso impróprio de uma facilidade de comunicações via rede na qual a mesma mensagem é enviada simultaneamente a muitos recipientes.
Fora de banda	Informação trocada usando meios de comunicação que são independentes da especificação do protocolo do SET.
Hash	Uma função que mapeia valores de um grande (possivelmente muito grande) domínio em uma faixa menor. Pode ser usado para reduzir uma mensagem potencialmente longa em um " valor de *hash*" ou "resumo de mensagem" que é suficientemente compacto para ser introduzido em um algoritmo de assinatura digital. Um "bom hash" é tal que os resultados da aplicação num (grande) conjunto de valores num domínio serão uniformemente (e aleatoriamente) distribuidos sobre a faixa.
***Host* de Captura de Dados**	Esta é uma opção de processamento sob a qual o computador hospedeiro (*host*) do Adquirente armazena as transações de "captura" de pagamento dos comerciantes. Dependendo da operação do sistema, e do acordo entre o Comerciante e o Adquirente, e do tipo de transações envolvidas, o Comerciante pode enviar um pedido de autorização combinado com o pedido de captura em uma única mensagem conhecida como um "pedido de venda". Outras opções admitem as transações de autorização e captura separadas, como também vários modos de processamento postal para balanceamento de lote de captura.
Idempotência	A propriedade por meio da qual pode-se repetir uma operação e o resultado é o mesmo. Em termos de protocolo, o envio de uma mensagem idempotente repetidamente não deveria resultar em nenhuma mudança do resultado
Impressão Digital	O resumo calculado sobre o item para gerar ou verificar a assinatura.
Indisponibilidade de Serviço	A inabilidade para ter acesso a informação ou aos recursos de processamento da informação por qualquer razão, como desastre, falha de energia, ou ações maliciosas.
Instituição Financeira	Um estabelecimento responsável por facilitar transações iniciadas pelo cliente ou transmissão de capitais para a extensão de crédito ou custódia, empréstimo, troca, ou fornecimento de dinheiro.

Integridade	A qualidade da informação ou um processo que está livre de erro, se induzido acidentalmente ou intencionalmente.
Interativo	Uma classe genérica de mecanismo de transporte de rede que é dependente de uma sessão lógica que é mantida durante a troca de mensagem (por exemplo, sessões de *World Wide Web*).
Internet	A maior coleção de redes do mundo, interconectadas de tal modo que pode funcionar como uma única rede virtual.
Interoperabilidade	A habilidade de trocar mensagens e chaves, manualmente e em um ambiente automatizado, com qualquer outra parte que implementa o padrão, desde que ambas as implementações usem opções compatíveis do padrão e instalações de comunicações compatíveis.
Lista de Revogação de Certificados (*Certificate Revocation List* - CRL)	Uma lista de números de série de certificados emitidos por uma autoridade de certificado que indica os certificados que são inválidos antes do vencimento normal devido a comprometimento, desafiliação, ou alguma outra circunstância incomum .
Logaritmo discreto	Dados dois elementos d , g num grupo tal que existe um inteiro r satisfazendo $g^r = d$, r é chamado de logaritmo discreto de d na base g .
Modificação da Informação	A mudança sem autorização ou acidental da informação, se detectada ou não detectada.
Não interativo	Uma classe genérica de mecanismos de rede de transporte que não é dependente de uma sessão lógica que é mantida durante a troca de mensagem (por exemplo, sessões de correio eletrônicas).
Não repúdio	A prova da integridade e origem dos dados - ambos em relação a forjabilidade - que pode ser verificada por qualquer parte. O SET não provê não repúdio, é sua intenção que o não repúdio seja feito via regras e políticas próprias da implementação de cada marca de cartão.
Nonce	Um valor aleatoriamente gerado usado para vencer ataques de repetição.
Número de Conta Primário (*Primary Account Number* –PAN)	O número assinado que identifica o emissor do cartão e o portador de cartão . Este número de conta é composto de um número de identificação do emissor, uma identificação de número de conta individual, e um dígito verificador que o acompanha, como definido pela ISO 7812-1985.
Pagamentos Recorrentes	Um tipo de transação de pagamento iniciado pelo Portador de cartão que permite ao Comerciante processar autorizações múltiplas. Há dois tipos de pagamentos recorrentes: pagamentos múltiplos para uma quantia fixa (por exemplo, quatro pagamentos de x valor) ou repetidos faturamentos (por exemplo, uma conta mensal de um provedor de serviço Internet).

Par de Pedido-Resposta	Um par de mensagens fluindo em direções opostas entre as mesmas partes e compartilhando o mesmo RRPID.
Parcelamento de Pagamentos	Um tipo de transação de pagamento negociado entre o Comerciante e o Portador de cartão que permite ao Comerciante processar autorizações múltiplas. O Portador de cartão especifica um número máximo de autorizações permitidas para serem realizadas em parcelamento do pagamento.
Payload	A informação enviada com uma mensagem sobre os dados do negócio.
Pedido de Bens ou Serviços	O preço, moeda corrente, método de pagamento, número de pagamentos, e outras condições da transação (também chamado de "Descrição do Pedido" no SET).
Pedido de Investigação	O pedido feito pelo portador de cartão ao comerciante para determinar o estado de um pedido de compra.
Pedido por telefone / Pedido pelo Correio (*Mail Order/Telephone Order* – MOTO)	O tipo de transação de cartão de pagamento onde a informação do pedido e pagamento é transmitida pelo correio ao Comerciante, ou através de telefone, em contraste com uma transação "de cartão presente" de transação face-a-face nas quais o cliente faz uma compra na loja do Comerciante. Este tipo de transação também é chamado de transação MOTO (*Mail Order/Telephone Order*).
PKI – *Public-Key Infrastructure*	PKIs são criadas para resolver problemas de gerenciamento de chave.
Política	Um elemento de dados definido no certificado X.509 que designa a política da marca, ou seja as definições de como o certificado será usado.
Portador de Cartão	O possuidor de uma conta de cartão de pagamento válida e usuário de *software* que apóia comércio eletrônico.
Portal de Pagamento	Um sistema operado por um Adquirente com a finalidade de prover serviços de comércio eletrônico aos Comerciantes em suporte ao Adquirente, e que se interfaceia com o Comerciante para possibilitar autorização e captura de transações.
Problema do logaritmo discreto	É o problema de achar r tal que $g^r = d$, onde d e g são elementos num dado grupo. Para alguns grupos, o problema do logaritmo discreto é um problema muito difícil e é usado em criptografia de chave pública.
Pseudo-aleatório	Um valor que é estatisticamente aleatório gerado por um processo algorítmico.
Rede	Uma coleção de sistemas de comunicação e informação que podem ser compartilhados entre vários usuários.

Remessa Parcial	Acontece quando o comerciante está impossibilitado de prover ou entregar um ou mais dos bens ou serviços pedidos para o portador de cartão, mais provávelmente devido a estoque insuficiente. O comerciante indica uma intenção de executar um pedido de autorização subseqüente ao Adquirente para os bens e serviços cancelados.
Renovação de Certificado	O processo pelo qual um certificado novo é criado para uma chave pública existente .
Resumo de Mensagem	O resultado de comprimento fixo quando uma mensagem de comprimento variável é introduzida em uma função *hash* unidirecional. Os resumos de mensagem ajudam verificar que uma mensagem não foi alterada porque a alteração da mensagem alteraria o resumo.
Revelação da Informação	Qualquer condição que resulta em observação sem autorização ou potencial observação da informação.
Reversão de Captura	Uma transação enviada quando a informação em uma mensagem de captura prévia estava incorreta ou nunca deveria ter sido enviada (tal como quando os bens não foram remetidos de fato). Se a reversão de captura é o resultado de informação incorreta, será seguida por uma nova mensagem de captura com a informação correta.
Reversão de Crédito	Uma transação enviada quando a informação em uma transação de crédito prévia estava incorreta ou nunca deveria ter sido enviada.
Revogação de Certificado	O processo de revogação de um certificado válido fornecido pela entidade que emitiu o certificado.
Risco	A possibilidade de perda por causa de uma ou mais ameaças à informação (não confundir com risco financeiro ou empresarial).
Salt	Uma cadeia de bits aleatórios (ou pseudo-aleatórios) concatenados com uma chave ou senha para conter ataques de precomputação.
Semente (*Seed*)	Tipicamente uma sequência de bits aleatórios usada para gerar, usualmente, uma outra sequência maior de bits pseudo-aleatória.
Sequência	Um agrupamento abstrato de zero ou mais elementos de dados. Também chamado uma “tupla”.
Servidor	Um computador que age como provedor de algum serviço a outros computadores, como no processamento de comunicações, se conecta com arquivos armazenados ou facilidades de impressão.
Smart Card	Um cartão, de dimensões similarares a de um cartão de crédito, que contém um *chip* de computador e é usado para armazenar ou processar informação.

Terminal de Captura de Dados	Uma opção de processamento na qual transações autorizadas são armazenadas num sistema baseado no Comerciante e submetidas de uma única vez ao Adquirente como uma transação de captura controlada e é especificado pelo Comerciante. Sob esta opção, o Comerciante controla os conteúdos do "lote de transações", como também o tempo de submissão da transação de captura. Não é exigido do Adquirente manter arquivos de captura em nome do Comerciante.
TCAP	*Transaction Capabilities Procedures*: É um protocolo para o desenvolvimento de serviços "inteligentes" sobre redes de telecomunicações. Um exemplo de utilização desse protocolo é o roteamento associado a chamadas telefônicas *tool-free*
Thumbs	Uma instância de uma ou mais impressões digitais.
Token* de *Hardware	Um dispositivo portátil (por exemplo, cartão inteligente, e cartões PCMCIA) especificamente projetado para armazenar informação criptográfica e possivelmente executar funções criptográficas de uma maneira segura.
Transação	Uma sucessão de uma ou mais mensagens relacionadas.
Transação de Vendas	Uma transação de autorização de pagamento que permite a um comerciante autorizar uma transação e pedir pagamento em uma única mensagem para o Adquirente .
Tupla	Um agrupamento abstrato de zero ou mais elementos de dados. Também chamado uma sequência.

Apêndice D

Padrões Externos Utilizados pelo Set

O projeto do SET é baseado nos padrões estabelecidos pela indústria , Internet e organizações internacionais como definido pelos padrões ISO, IEFT, PKCS e ANSI. Este apêndice identifica os padrões, algoritmos e certificados utilizados pelas especificações do SET.

ASN.1	Abstract Syntax Notation A ASN.1 é a notação usada pelo SET para a especificação de mensagens. A versão de 1995 da especificação ASN.1 é descrita nos documentos ISO/IEC 8824-1, 8824-2, 8824-3, e 8824-4.
DER	*Distinguished Encoding Rules* Implementa a codificação em uma forma não ambígua dos dados do protocolo de mensagens de pagamento e de certificados (como especificado no padrão X.509). A versão de 1995 da especificaçãso DER está descrita na ISO/IEC 8825-1.
DES	*Data Encryption Standard* Padrãopara cifragem de dados (como especificado na FIPS PUB 46-2). A chave DES é distribuída em uma forma cifrada dentro de um envelope digital usando criptografia de chave pública.
HMAC	Mecanismo de *hash* com chave para função de compartilhamento e ocultação de segredo.
HTTP	*Hyper-Text Transport Protocol* Este Protocolo de Transporte da *World Wide Web* suporta os *browsers* e servidores existentes (como especificado na RFC 1945).
ISO 3166:1993	Códigos para representação de nomes de países.
ISO 4217:1995	Códigos para a representação de moedas e fundos.
ISO 7812:1985	Identificação de Cartões – Sistema de Numeração e procedimento de registro que inclui a definição para computação de dígito verificador.

ISO 8583:1993	Mensagens de Transações Financeiras Geradas por Cartões – Especificações de Troca de Mensagens.
ISO 9594-8:1997	ITU-T Recomendação X.509 (1997), Tecnologia da Informação – Interconexão de Sistemas Abertos – O Diretório : Authentication.Framework O formato de certificado utilizado pelo SET.
ISO 9834-7	Provê uma autoridade de registro internacional para identificadores de objetos
MIME	*Multipurpose Internet Message Extensions* Usado para cifragem de envelopes para mensagens de pagamento, dota *browsers* da capacidade de operar mensagens de pagamento e dá suporte ao comércio eletrônico baseado em *e-mails*.
PKCS	*Public Key Cryptography Standards* Define a sintaxe de mensagem criptográfica (PKCS #7) e sintaxe de mensagem de pedido de certificado (PKCS #10).
RFC 1766	Etiqueta padrão de idioma.
SHA-1	*Secure Hashing Algorithm* Desenvolvido conjuntamente pelo NIST e NSA (como especificado na FIPS 180-1). O SET usa o SHA-1 para todas as assinaturas digitais.
TCP/IP	Família de Protocolos para tratamento da informação transportada sobre a Internet.
X.509	ITU-T Recomendação X.509 (1997)|ISSO/IEC 9594-8:1997 Padrão para cifragem de Certificados de Chaves Públicas O formato de certificado utilizado pela Especificação SET é definido na ISO padrão X.509 versão 3; ANSI X9.57 (também ISO/IEC 9594-8:1993).

Bibliografia

[1] MasterCard e Visa (1997), **SET Secure Electronic Transactions Specification, Book 1: Business Description Guide, Version 1.0,** url: www.setco.com.

[2] MasterCard e Visa (1997), **SET Secure Electronic Transactions Specification, Book 2: Programmers Guide, Version 1.0,** url: www.setco.com,.

[3] MasterCard e Visa (1997), **SET Secure Electronic Transactions Specification, Book 3: Formal Protocol Defition, Version 1.0,** url: www.setco.com.

[4] Merkow, M. S. , Breithaupt, J. , Wheeler K. L.(1998) , **Building Set Aplications for Secure Transactions**, USA: John Wiley & Sons, Inc.

[5] Burnett, S. , Paine, S., (2002), **Criptografia e Segurança : o guia oficial RSA**, Rio de Janeiro: Campus Ltda.

[6] Terada, R., (2000) , **Segurança de Dados Criptografia em Redes de Computador**, São Paulo: Edgard Blucher Ltda.

[7] Buchmann, J. A . (2002) , **Introdução à Criptografia** , São Paulo : Berkeley Brasil.

[8] Jannuzzi, G. F. (1999), **Soluções de Comércio Eletrônico**, IME – Instituto Militar de Engenharia , Rio de Janeiro, Brasil, url: www.glauter@leblon.ime.eb.br.

[9] Veloso, C. J. M. (2002), **Criptologia Uma ciência fundamental para tratamento de informações sigilosas**, MÓDULO e-security, Minas Gerais, Brasil, url: www.modulo.com.br.

[10] Torres, M. F. C. F. (2000), **Comércio Eletrônico – Estudo de Caso**, Unicamp, São Paulo, Brasil.

[11] Departamento de Produtos de Informática do Bradesco (1998), **A Internet no Bradesco – Seminário – Compras Governamentais**, Brasília, Brasil, e-mail: 4000.odecio@bradesco.com.br.

[12] NIST (1993), **Federal Information Processing Standandards Publications 46-2: DATA ENCRYPTION STANDARDS (DES)**, EUA: National Institute of Standards and Technology.

[13] NIST (2001), **Federal Information Processing Standandards Publications 197: ADVANCED ENCRYPTION STANDARDS (AES)**, EUA: National Institute of Standards and Technology.

[14] Daemen, J., Rijmen, V.(1999), **AES Proposal: Rijndael, AES Algorithm Submission.**

[15] Feistel, H., (1974), **Block cipher cryptografic system**, U. S. Patent # 3,798,359.

[16] NIST (1995), **Federal Information Processing Standandards Publications 81: DES MODES OF OPERATION**, EUA: National Institute of Standards and Technology.

[17] Diffie, W., Hellman, M. E. (1976), **New Directions in cryptography**, IEEE Transactions on Information Theory, vol 22, # 6, pp. 644-654.

[18] RSA Laboratories (1998), **PKCS # 1 v 2.0: RSA Cryptography Standard**, EUA: RSA Data Security.

[19] Rivest, R., Shamir, A., Adleman, L., (1978), **A method for obtaining digital signatures on public key criptosystems** , Communications of the ACM, Feb , vol. 21(2) , 120-126.

[20] Pomerance, C. (1985), **The Quadratic Sieve factoring algorithm** , Eurocrypt 84 , Lec. Notes in Comp. Sci. 209 , 169-182.

[21] Lenstra, K., Lenstra, H. W., Manasse, M. S., Pollard, J. M. (1993), **The number field sieve** , vol. 1554, Lecture Notes im Mathematics , 11- 12, Springr-Verlag.

[22] Shamir, A., (1999), **Factoring large numbers with the TWINKLE device**, Proceedings of EUROCRYPT'99, Lecture Notes in Computer Science , Springer-Verlag.

[23] NIST (1995), **Federal Information Processing Standandards Publications 180-1: SECURE HASH STANDARD**, EUA: National Institute of Standards and Technology.

[24] NIST (1994), **Federal Information Processing Standandards Publications 186: DIGITAL SIGNATURE STANDARD (DSS)**, EUA: National Institute of Standards and Technology.

[25] Information Technology Laboratory (2002), **Federal Information Processing Standandards Publications 198: The Keyed-Hash Message Authentication Code (HMAC)**,Gaitesburg: : National Institute of Standards and Technology.

[26] ITU Rec. X.509 (1993), **ISO/IEC 9594-8: 1995, including Draft Amendment 1:Certificate Extensions (Version 3 certificate)**

[27] RSA Laboratories (1993), **PKCS # 7 : RSA Cryptography Message Syntax Standard**, EUA: RSA Data Security.

[28] Moraes, A.C.K., Silva, J.F., Costa, W.C. (2002), **Internet - Invasão e as Perdas Financeiras,** url: www.gocsi.com.

[29] IBM T.J. Watson Research Center (1971), IBM Research Report RC 3326, **The design of Lucifer : A cryptographic device for data communications**, Yorktown Heigths, N. Y. , 10598, U.S.A.

[30] Johnson, D., Matyas, S. Le, A., Wilkins, J. (1993), **Design of the Commercial Data Masking Facility Data Privacy Algorithm**, Proceedings of the First ACM Conference on Communications and Computer Security, ACM Press, U.S.A.

[31] Burton, D. M. (1998), **Elementary Number Theory,** 4ª Ed., McGraw-Hill.