

[Além disso | Além do mais], o professor Wilson é um [ grande | fantástico] administrador. [ Seus | Suas] [contratos | concessões] trouxeram uma [ grande | substancial] quantia em dinheiro para [ o | nosso] departamento. [ Esse dinheiro | Esses fundos] [permitiu | permitiram] que [criássemos | realizássemos] muitos programas [ especiais | importantes], [tais como | por exemplo] o programa Universidade 2000. Sem esses fundos, [ seríamos incapazes | não seríamos capazes] de dar continuidade a esse programa, que é tão [ importante | essencial] para nós. Afirmo ao senhor que ele é o profissional mais adequado para essa posição.

Infelizmente para Tom, assim que acaba de redigir e digitar essa carta, Ellen também digita a seguinte carta:

Caro Sr. Reitor,

Esta [carta | mensagem] tem como objetivo expressar minha [ honesta | franca] opinião a respeito do professor Tom Wilson, que [ é | está] [candidato | prestes] [a / para] assumir uma vaga permanente nesta universidade [ imediatamente | este ano]. Eu [conheço | trabalho com] Tom há seis anos. Ele é um [ incompetente | mau] pesquisador, não é bem visto em sua [ especialidade | área]. Sua pesquisa [raramente | esporadicamente] mostra [ bom-senso | conhecimento] dos [principais | mais importantes] problemas atuais.

Ele não é um [ professor | educador] [bastante | muito] [respeitado | admirado]. Seus alunos fazem [duras | pesadas] críticas de suas [ aulas | cursos]. Ele é o [professor | orientador] mais impopular [ da universidade | do departamento] devido a sua [ tendência | propensão] a [ridicularizar | embaraçar] os alunos que fazem perguntas em suas aulas.

[Além disso | Além do mais], Tom é um administrador [ terrível | fraco]. [Seus | Suas] [contratos | concessões] trouxeram apenas uma [ insignificante | pequena] quantia em dinheiro para [ o | nosso] departamento. A menos que mais [ verbas | fundos] sejam [alocadas | alocados], teremos de cancelar alguns programas essenciais, tais como o seu programa Universidade 2000. Infelizmente, sob essas [condições | circunstâncias], não posso recomendá-lo em sã consciência para essa posição.

Ellen passa a noite configurando seu computador para calcular os  $2^{32}$  sumários de mensagens de cada carta. Há chances de que um sumário da primeira carta corresponda a um sumário da segunda carta. Caso isso não aconteça, ela poderá incluir algumas outras opções e tentar de novo durante o fim de semana. Suponha que ela encontre uma correspondência. Vamos chamar a carta "boa" de A e a "ruim" de B.

Em seguida, através do correio eletrônico, Ellen envia a carta A a Marilyn para que seja aprovada. Ela mantém a carta B completamente secreta, sem mostrá-la a ninguém. É claro que Marilyn aprova a carta, calcula seu sumário de mensagens de 64 bits, assina o sumário e envia o sumário assinado ao reitor Smith utilizando o correio eletrônico. Por outro lado, Ellen envia a carta B ao reitor (não a carta A, como deveria fazer).

Depois de obter a carta e o sumário de mensagens assinado, o reitor executa o algoritmo de sumário de mensagens na carta B, observa que ela corresponde ao sumário que Marilyn enviou e despede Tom. O reitor não percebe que Ellen gerou duas cartas com o mesmo sumário de mensagens e enviou a ele uma mensagem diferente da que Marylin viu e aprovou. (Final opcional: Ellen conta a Dick o que fez. Dick não gosta do que ouve e termina o namoro com ela. Ellen fica furiosa e confessa tudo a Marilyn. Marilyn telefona para o reitor. Tom acaba ficando com o cargo.) Com o MD5, o ataque de aniversário se torna impraticável, pois mesmo com um bilhão de sumários por segundo, seriam necessários 500 anos para calcular  $2^{64}$  sumários de duas cartas com 64 variantes cada uma e, de qualquer forma, não poderíamos ter certeza de que haveria uma correspondência. É claro que, com 5000 computadores trabalhando em paralelo, 500 anos se transformam em cinco semanas. O SHA-1 é melhor (por ser mais longo).

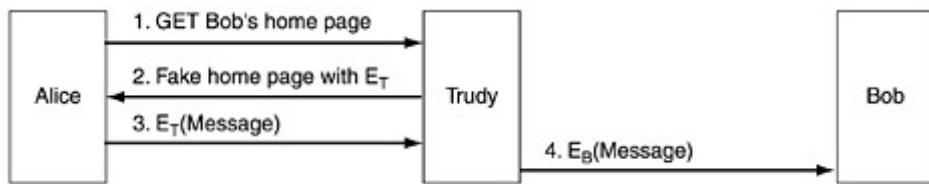
## 8.5 Gerenciamento de chaves públicas

A criptografia de chave pública torna possível a comunicação segura para pessoas que não compartilham uma chave comum, e também possibilita a assinatura de mensagens sem a presença de uma terceira parte confiável. Finalmente, os sumários de mensagens assinados permitem verificar com facilidade a integridade de mensagens recebidas.

Porém, existe um problema que ignoramos até aqui: se Alice e Bob não conhecem um ao outro, como ele irão obter as respectivas chaves públicas para iniciar o processo de comunicação? A

solução óbvia — colocar a chave pública no Web site — não funciona pela seguinte razão: suponha que Alice queira pesquisar a chave pública de Bob em seu Web site. Como ela fará isso? Bem, Alice começa digitando o URL de Bob. Seu navegador então pesquisa o endereço DNS da home page de Bob e envia a ele uma solicitação GET, como mostra a Figura 8.23. Infelizmente, Trudy intercepta a solicitação e responde com uma home page falsa, talvez uma cópia da home page de Bob, exceto pela substituição da chave pública de Bob pela chave pública de Trudy. Quando Alice codifica sua primeira mensagem com  $E_T$ , Trudy a decodificará, lerá e recodificará com a chave pública de Bob, enviando a mensagem a Bob, que não sabe que Trudy está lendo suas mensagens recebidas. Pior ainda, Trudy poderia modificar as mensagens antes de recodificá-las para Bob. É claro que há necessidade de algum mecanismo para garantir que as chaves públicas possam ser trocadas em segurança.

**Figura 8.23:** Um modo de Trudy subverter a criptografia de chave pública



### 8.5.1 Certificados

Como uma primeira tentativa de distribuição de chaves públicas com segurança, poderíamos imaginar um centro de distribuição de chaves disponível on-line 24 horas por dia a fim de fornecer chaves públicas por demanda. Um dos muitos problemas com essa solução é o fato de ela não ser escalável, e o centro de distribuição de chaves rapidamente se tornaria um gargalo. Além disso, se ele ficasse inativo, a segurança da Internet seria paralisada repentinamente.

Por essas razões, as pessoas desenvolveram uma solução diferente, que não exige que o centro de distribuição de chaves esteja on-line todo o tempo. De fato, ele não precisa estar on-line de modo algum. Em vez disso, ele certifica as chaves públicas pertencentes a pessoas, empresas e outras organizações. Uma organização que certifica chaves públicas é chamada **CA** (Certification Authority — autoridade de certificação).

Como um exemplo, suponha que Bob queira permitir que Alice e outras pessoas se comuniquem com ele em segurança. Ele pode ir até a CA com sua chave pública e seu passaporte ou com a carteira de motorista e solicitar a certificação. A CA emite então um certificado semelhante ao da Figura 8.24 e assina seu hash SHA-1 com a chave privada da CA. Em seguida, Bob paga a taxa da CA e obtém um disquete contendo o certificado e seu hash assinado.

**Figura 8.24:** Um certificado possível e seu hash assinado



A principal função de um certificado é vincular uma chave pública ao nome de um protagonista (indivíduo, empresa etc.). Os certificados em si não são secretos ou protegidos. Por exemplo, Bob poderia decidir colocar seu novo certificado em seu Web site, com um link na página principal informando: clique aqui para ver meu certificado de chave pública. O clique resultante retornaria o certificado e o bloco de assinatura (o hash SHA-1 assinado do certificado).

Agora vamos percorrer o cenário da Figura 8.23 novamente. Quando Trudy intercepta a solicitação de Alice para a home page de Bob, o que ela pode fazer? Trudy pode inserir seu próprio certificado e seu bloco de assinatura na página falsa; porém, quando Alice ler o certificado, verá imediatamente que não está se comunicando com Bob, porque o nome de Bob não está no certificado. Trudy pode modificar a home page de Bob durante a execução, substituindo a chave

pública de Bob pela sua própria chave. Porém, quando Alice executar o algoritmo SHA-1 no certificado, ela obterá um hash que não concorda com o que ela recebe ao aplicar a chave pública conhecida da CA ao bloco de assinatura. Como Trudy não tem a chave privada da CA, ela não tem meios de gerar um bloco de assinatura que contenha o hash da página da Web modificada com sua chave pública. Desse modo, Alice pode estar certa de que possui a chave pública de Bob e não a de Trudy ou de outra pessoa. Como prometemos, esse esquema não exige que a CA esteja on-line para verificação, eliminando assim um gargalo potencial.

Embora a função padrão de um certificado seja vincular uma chave pública a um protagonista, um certificado também pode ser usado para vincular uma chave pública a um atributo. Por exemplo, um certificado poderia afirmar: esta chave pública pertence a alguém com mais de 18 anos. Ela pode ser usada para provar que o proprietário da chave privada não é menor de idade e, portanto, pode acessar material não apropriado para crianças e assim por diante, mas sem revelar a identidade do proprietário. Em geral, a pessoa que tivesse o certificado o enviaria ao Web site, ao protagonista ou ao processo que solicitasse informações sobre a idade. Esse site, protagonista ou processo, geraria então um número aleatório e o codificaria com a chave pública no certificado. Se o proprietário fosse capaz de decodificá-lo e enviá-lo de volta, essa seria prova de que o proprietário de fato tinha o atributo declarado no certificado. Como alternativa, o número aleatório poderia ser usado para gerar uma chave de sessão pela duração da conversação.

Outro exemplo de situação em que um certificado poderia conter um atributo é um sistema distribuído orientado a objetos. Em geral, cada objeto tem diversos métodos. O proprietário do objeto poderia fornecer a cada cliente um certificado com um mapa de bits dos métodos que o cliente tem permissão para invocar e vincular o mapa de bits a uma chave pública, usando um certificado assinado. Mais uma vez, se o proprietário do certificado puder provar a posse da chave privada correspondente, ele terá permissão para executar os métodos no mapa de bits. Ele não precisa conhecer a identidade do proprietário, uma característica útil em situações nas quais a privacidade é importante.

## 8.5.2 X.509

Se todo mundo quisesse que algo assinado foi enviado a CA com um tipo de certificado diferente, logo se tornaria um problema administrar todos os formatos diferentes. Para resolver esse problema, foi criado e aprovado pela ITU um padrão para certificados. O padrão é chamado **X.509** e seu uso está difundido na Internet. Ele passou por três versões desde a padronização inicial em 1988. Vamos descrever a V3.

O X.509 foi fortemente influenciado pelo mundo OSI, tomando emprestadas algumas de suas piores características (por exemplo, nomenclatura e codificação). Surpreendentemente a IETF aceitou o X.509, embora em quase todas as outras áreas — desde endereços de máquina até protocolos de transporte e formatos de correio eletrônico — ela tenha ignorado a OSI e tentado fazer tudo da maneira certa. A versão da IETF do X.509 é descrita na RFC 3280.

Em seu núcleo, o X.509 é um modo de descrever certificados. Os principais campos em um certificado estão listados na Figura 8.25. As descrições dadas na figura devem fornecer uma idéia geral do significado dos campos. Para obter informações adicionais, consulte o próprio padrão ou a RFC 2459.

Por exemplo, se Bob trabalhar no departamento de empréstimos do Money Bank, seu endereço X.500 poderá ser:

/C=US/0=MoneyBank/OU=Loan/CN=Bob/

onde C é o país, O é a organização, OU é a unidade organizacional e CN é o nome comum. As CAs e outras entidades são identificadas de modo semelhante. Um problema significativo com os nomes X.500 é que, se Alice estiver tentando entrar em contato com bob@moneybank.com e receber um certificado com um nome X.500, talvez não fique claro para ela a que Bob o certificado se refere. Felizmente, a partir da versão 3, os nomes DNS são permitidos, em lugar de nomes X.500; assim, esse problema eventualmente deve desaparecer.

**Figura 8.25:** Os campos básicos de um certificado X.509

Campo	Significado
Version	A versão do X.509

Serial number	Este número, somado ao nome da CA, identifica de forma exclusiva o certificado
Signature algorithm	O algoritmo usado para assinar o certificado
Issuer	Nome X.500 da CA
Validity period	A hora inicial e final do período de validade
Subject name	A entidade cuja chave está estando certificada
Public key	A chave pública do assunto e a ID do algoritmo que a utiliza
Issuer ID	Uma ID opcional que identifica de forma exclusiva o emissor do certificado
Subject ID	Uma ID opcional que identifica de forma exclusiva o assunto do certificado
Extensions	Muitas extensões foram definidas
Signature	A assinatura do certificado (assinado pela chave privada da CA)

Os certificados são codificados com o uso da **ASN.1 (Abstract Syntax Notation 1)** da OSI, que pode ser considerada uma estrutura em C, exceto por sua notação muito peculiar e extensa. Para obter mais informações sobre o X.509, consulte (Ford e Baum, 2000).

### 8.5.3 Infra-estruturas de chave pública

Fazer uma única CA emitir todos os certificados do mundo evidentemente não funcionaria. Ela entraria em colapso sob a carga e também seria um ponto central de falha. Uma solução possível poderia ser a existência de várias CAs, todas administradas pela mesma organização e todas usando a mesma chave privada para assinar certificados. Embora isso pudesse resolver o problema da carga e da falha, há um novo problema: o vazamento de chaves. Se houvesse dezenas de servidores espalhados pelo mundo, todos com a chave privada da CA, a chance de que a chave privada fosse roubada ou sofresse algum outro tipo de vazamento seria bastante aumentada. Tendo em vista que o comprometimento dessa chave arruinaria a infra-estrutura de segurança eletrônica do mundo, a existência de uma única CA central é muito arriscada.

Além disso, que organização operaria a CA? É difícil imaginar uma autoridade que fosse aceita em todo o mundo como uma entidade legítima e confiável. Em alguns países, as pessoas insistiriam em que essa entidade fosse um governo; em outros países, elas insistiriam que não fosse um governo.

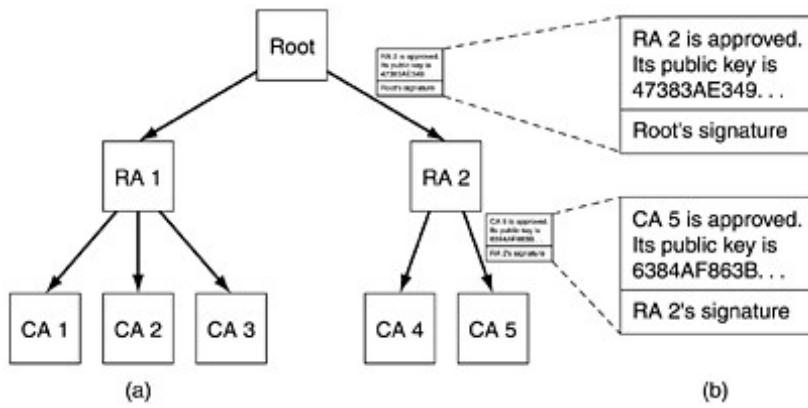
Por essas razões, foi desenvolvido um modo diferente de certificar chaves públicas, identificada pelo nome geral PKI (Public Key Infrastructure). Nesta seção, resumiremos como ela funciona em linhas gerais, embora existam muitas propostas relativas aos detalhes que provavelmente irão evoluir com o tempo.

Uma PKI tem vários componentes, incluindo usuários, CAs, certificados e diretórios. A função da PKI é fornecer um modo de estruturar esses componentes e definir padrões para os vários documentos e protocolos. Uma forma particularmente simples de PKI é uma hierarquia de CAs, como mostra a Figura 8.26. Nesse exemplo, mostramos três níveis mas, na prática, pode haver um número menor ou maior. A CA de nível superior, chamada raiz, certifica CAs do segundo nível, que denominamos RAs (Regional Authorities), porque podem cobrir alguma região geográfica, como um país ou um continente. Entretanto, esse termo não é padrão; de fato, nenhum termo é realmente padrão para os diferentes níveis da árvore. Por sua vez, as RAs certificam as CAs reais, que emitem os certificados X.509 para organizações e indivíduos. Quando a raiz autoriza uma nova RA, ela gera um certificado X.509 anunciando que aprovou a RA, inclui a chave pública da nova RA no certificado, assina o certificado e o entrega à RA. De modo semelhante, quando uma RA aprova uma nova CA, ela produz e assina um certificado declarando sua aprovação e contendo a chave pública da CA.

Nossa PKI funciona de modo semelhante. Suponha que Alice precise da chave pública de Bob, a fim de se comunicar com ele; então, ela procura e encontra um certificado contendo a chave, assinado pela CA 5. Porém, Alice nunca ouviu falar da CA 5. Tudo que ela sabe é que a CA 5 pode ser a filha de 10 anos de Bob. Ela poderia ir até a CA 5 e dizer: prove sua legitimidade. A CA 5 responde com o certificado que recebeu da RA 2, que contém a chave pública da CA 5. Agora, munida da chave pública da CA 5, Alice pode confirmar que o certificado de Bob foi de fato assinado pela CA 5 e, portanto, é válido.

A menos que a RA 2 seja o filho de 12 de Bob. Nesse caso, a próxima etapa é pedir a RA 2 que prove sua legitimidade. A resposta à consulta de Alice é um certificado assinado pela raiz e contendo a chave pública da RA 2. Agora, Alice tem certeza de que possui a chave pública de Bob.

**Figura 8.26:** (a) Uma PKI hierárquico. (b) Uma cadeia de certificados



No entanto, como Alice encontra a chave pública da raiz? Por mágica. Supõem-se que todo mundo conhece a chave pública da raiz. Por exemplo, seu navegador pode ter sido comercializado com a chave pública da raiz embutida.

Bob é o tipo de sujeito amigável e não querer dar muito trabalho a Alice. Ele sabe que Alice vai ter de verificar a CA 5 e a RA 2; assim, para evitar dificuldades, ele reúne os dois certificados necessários e os fornece a ela juntamente com o seu próprio certificado. Agora, ela pode usar seu conhecimento da chave pública da raiz para confirmar o certificado de nível superior e a chave pública que ele contém para verificar o segundo certificado. Desse modo, Alice não precisa entrar em contato com ninguém para fazer a verificação. Como os certificados são todos assinados, ela pode detectar com facilidade quaisquer tentativas de falsificar seu conteúdo. Uma cadeia de certificados como essa que volta à raiz, às vezes é chamada **cadeia de confiança** ou caminho de certificação. A técnica é amplamente utilizada na prática.

É claro que ainda temos o problema de saber quem vai administrar a raiz. A solução é não ter uma única raiz, mas sim várias raízes, cada uma com suas próprias RAs e CAs. De fato, os navegadores modernos são pré-carregados com as chaves públicas de mais de 100 raízes, às vezes referidas como âncoras de confiança. Desse modo, pode-se evitar ter uma única autoridade confiável no mundo inteiro.

Entretanto, agora existe a questão de como o fornecedor do navegador decide quais das supostas âncoras de confiança são de fato confiáveis e quais são desprezíveis. Tudo se reduz à confiança do usuário no fornecedor do navegador para fazer escolhas sensatas e não aprovar simplesmente todas as âncoras de confiança dispostas a pagar por sua inclusão na lista. A maioria dos navegadores permite que os usuários inspecionem as chaves da raiz (em geral, sob a forma de certificados assinados pela raiz) e eliminem qualquer uma que parecer obscura.

## Diretórios

Outra questão importante para qualquer PKI é onde estão armazenados os certificados (e suas cadeias de retorno até alguma âncora de confiança conhecida). Uma possibilidade é fazer cada usuário armazenar seus próprios certificados. Embora isso seja seguro (isto é, não existe nenhum meio dos usuários adulterarem certificados assinados sem detecção), também é inconveniente. Uma alternativa proposta é usar o DNS como um diretório de certificados. Antes de entrar em contato com Bob, é provável que Alice tenha de pesquisar seu endereço IP usando o DNS; então, por que não fazer o DNS retornar toda a cadeia de certificados de Bob juntamente com seu endereço IP?

Algumas pessoas acham que essa é a melhor alternativa, mas outras talvez prefiram servidores de diretórios dedicados cuja única tarefa seja administrar certificados X.509. Tais diretórios poderiam fornecer serviços de pesquisa usando propriedades dos nomes X.500. Por exemplo, na teoria tal serviço de diretório poderia transferir uma consulta como: "Forneça uma lista de todas as pessoas chamadas Alice que trabalham em departamentos de vendas de algum lugar nos Estados Unidos ou no Canadá". O LDAP poderia conter tais informações.

## Revogação

O mundo real também está repleto de certificados, como de passaportes e carteiras de motoristas. Às vezes, esses certificados podem ser revogados, bem como as carteiras de motoristas que são flagrados dirigindo bêbedos ou cometendo outras infrações de trânsito. O mesmo problema ocorre

no mundo digital: a autoridade que concede um certificado pode decidir revogá-lo porque a pessoa ou organização que possui o certificado cometeu algum abuso. Ele também pode ser revogado se a chave privada foi exposta ou, pior ainda, se a chave privada da CA foi comprometida. Desse modo, uma PKI precisa lidar com a questão da revogação.

Um primeiro passo nessa direção é fazer cada CA emitir periodicamente uma CRL (Certificate Revocation List — lista de revogação de certificados) fornecendo os números de série de todos os certificados que ela revogou. Tendo em vista que os certificados contêm prazos de validade, a CRL só precisa conter os números de série de certificados ainda não vencidos. Uma vez que seu prazo de validade tenha passado, um certificado se torna automaticamente inválido, e assim não há necessidade de distinção entre os que alcançaram o prazo limite e os que foram de fato revogados. Em ambos os casos, eles não podem mais ser utilizados.

Infelizmente, a introdução de CRLs significa que um usuário prestes a usar um certificado deve agora adquirir a CRL para ver se o certificado foi revogado. Se foi, ele não deve ser usado. Porém, mesmo que o certificado não esteja na lista, ele poderia ter sido revogado logo após a lista ter sido publicada. Desse modo, a única forma de realmente ter certeza é consultar a CA. Além disso, no próximo uso do certificado, a CA tem de ser consultada de novo, pois o certificado poderia ter sido revogado alguns segundos antes.

Outra complicação é o fato de um certificado revogado poder ser reabilitado, por exemplo, se tiver sido revogado por não pagamento de alguma taxa que foi paga posteriormente. Ter de lidar com a revogação (e talvez com a reabilitação) elimina uma das melhores propriedades dos certificados, ou seja, a possibilidade de usá-los sem ter de entrar em contato com uma CA.

Onde as CRLs devem ser armazenadas? Um boa alternativa seria armazená-las no mesmo local em que estão os próprios certificados. Uma estratégia é a CA publicar ativamente CRLs periódicas e fazer os diretórios processá-las apenas removendo os certificados revogados. Se os diretórios não forem usados para armazenar os certificados, as CRLs poderão ser guardadas no cache em diversos locais convenientes na rede. Tendo em vista que uma CRL também é um documento assinado, se ela for adulterada, essa ação poderá ser facilmente detectada.

Se os certificados tiverem uma longa duração, as CRLs também serão longas. Por exemplo, se os cartões de crédito forem válidos por 5 anos, o número de revogações pendentes será muito maior do que seria se fossem emitidos novos cartões a cada 3 meses. Um modo padrão de lidar com CRLs longas é emitir uma lista mestre com pouca freqüência, mas em intervalos de atualizações freqüentes para a lista. Isso reduz a largura de banda necessária para distribuir as CRLs.

## **8.6 Segurança da comunicação**

Agora, concluímos nosso estudo das principais ferramentas. A maior parte das técnicas e protocolos importantes foi abordada. O restante do capítulo estuda a aplicação dessas técnicas na prática para proporcionar segurança às redes, além de alguns conceitos sobre os aspectos sociais da segurança, no final do capítulo.

Nas quatro seções a seguir, examinaremos a segurança da comunicação, isto é, como levar os bits secretamente e sem alteração da origem até o destino, e como manter bits indesejáveis do lado de fora. Essas não são de modo algum as únicas questões de segurança em redes, mas certamente estão entre as mais importantes, o que nos dá um bom ponto de partida.

### **8.6.1 IPsec**

A IETF sabia há muitos anos que havia carência de segurança na Internet. Não era fácil aumentá-la, porque havia uma disputa para definir onde colocá-la. A maioria dos especialistas em segurança acredita que, para serem realmente seguras, a criptografia e as verificações de integridade devem ser realizadas de fim a fim (isto é, na camada de aplicação). Desse modo, o processo de origem criptografa e/ou protege a integridade dos dados e os envia ao processo de destino, onde eles são descriptografados e/ou verificados. Qualquer adulteração realizada entre esses dois processos, inclusive dentro de qualquer sistema operacional, poderá então ser detectada. A dificuldade com essa abordagem é que ela exige a troca de todas as aplicações, a fim de torná-las cientes da segurança. Nessa visão, a segunda melhor abordagem é inserir a criptografia na camada de transporte ou em uma nova camada entre a camada de aplicação e a camada de transporte, tornando-a ainda fim a fim, mas sem exigir que as aplicações sejam alteradas.