CAPÍTULO 8
CAMADA DE APRESENTAÇÃO

Antônio Carlos Neto (netiin.carlos.nic@hotmail.com)
Gustavo de Faria Silva (gustavofaria@ufu.br)
Lucas Rossi Rabelo (lucasrossi98@hotmail.com)
Marcelo Mendonça Borges (marcelomborges@outlook.com)
Matheus Pimenta Reis (matheuspr96@hotmail.com)

Sumário

- 8.1 - Questões de Design da Camada de Apresentação
- 8.2 - Notação de Sintaxe Abstrata 1 (ASN.1)
- 8.3 - Técnicas de Compressão de Dados
- 8.4 - Criptografia
- 8.5 - Exemplos da Camada de Apresentação

8 - Camada de Apresentação

- A camada de apresentação passou por mudanças ao longo do desenvolvimento do OSI
- Ficou por muito tempo em busca de uma função
- Foi usada para fazer as conversões necessárias para permitir que uma máquina ASCII para falar com máquinas EBCDIC
- Começo de sua estruturação

8 - Camada de Apresentação

- Finalmente se define sua função
- Lidar com a representação de dados, ou seja:
  1) Conversão
  2) Criptografia
  3) Compactação

8 - Camada de Apresentação

- As camadas inferiores lidam com o movimento dos dados
- E a camada de apresentação? Uma de suas responsabilidade é preservar o significado da informação transportada
- Um computador pode ter sua forma de representação de dados e como fazermos a troca de dados de forma que ambos se entendam
- Tarefa da camada de apresentação, codificar estruturas para codificar e decodificar dados.

8.1 - Questões de Design

- A camada de apresentação possui 4 Funções principais
  1) Que o usuário possa executar as primitivas da camada de sessão
  2) Especificar estruturas complexas de dados
  3) Gerenciar o conjunto de estruturas
  4) Converter dados de forma interna e externa

8.1 - Questões de Design

8.1.1 - Representação de Dados

- Como computadores que usam complemento de 2 se comunicam com os que usam complemento de 1
- Array de inteiros de 16bits de uma maquina pra outra como será representado o FFF0 (Hexa)?
- Em algum lugar preciso de uma conversão
- As camadas inferiores tiveram um enorme trabalho para garantir que as mensagens fossem transmitidas de forma correta

8.1 - Questões de Design

8.1.2 - Compressão de Dados

- Principais interesse pelas empresas em reduzir a quantidade de dados que trafega
- Quem está associado a este tema é a representação de dados
- Exemplo simples
- Como enviar um inteiro de 32 bits, simplesmente o codificamos em 4 bytes em alguma representação
- Amplamente usada e estudada, permite economia de espaço em memória e disco. Estudaremos alguns métodos aplicável na camada de apresentação.

8.1.3 - Segurança e Privacidade na Rede

- No começo era fácil controlar a segurança
- Como policiar milhões de bits dados que circulam na rede
- As organizações não garantia que seus dados não fossem copiados secretamente ou manipulados
- Aí surge um novo apelo, a Criptografia

8.1.3 - Segurança e Privacidade na Rede

- Exemplo: Envolve 2 processos o Hash(resumo) e a encriptação deste Hash
  1) Gera um resumo criptográfico da mensagem que se da o nome de Hash
  2) Este deve ser criptografado através de um sistema publico de chave
  3) O Autor da mensagem deve usar sua chave privada para assinar a mensagem e armazenar o Hash criptografado junto a mensagem original
  4) Para verificar a autenticidade deve ser gerado um novo resumo a partir da mensagem armazenada e descriptografá-la assim obtendo o Hash, se o Hash for igual ao original a mensagem é integra.

8.1.3 - Segurança e Privacidade na Rede

- Como solucionar estes 4 Serviços?
- Existe ao menos 4 serviços de segurança
  1) Que dados não possam ser lidos por pessoas não autorizadas
  2) Prevenção contra pessoas não autorizadas que desejem manipular os dados
  3) Verificar o remetente de cada mensagem
  4) Possibilitar que os usuários enviem documentos assinados eletronicamente

8.1.3 - Segurança e Privacidade na Rede

- A Criptografia
- A criptografia teve suas dificuldades de achar um lugar no modelo OSI, tanto que foi até omitida no padrão inicial
- Em teoria a criptografia pode ser feita em qualquer camada, mas na pratica três camadas são mais adequadas: Física, transporte e apresentação

8.1.3 - Segurança e Privacidade na Rede

- Criptografia na camada física
- Uma unidade de criptografia é inserida entre os computadores.
- Cada bit que sai é criptografado e cada bit que chega ao destinatário é descriptografado
- Chamado de “Link encryption”
- Sua principal vantagem é que criptografa tudo até o cabeçalho

8.1.3 - Segurança e Privacidade na Rede

- Análise de tráfego
- Para as aplicações comerciais a análise de tráfego não é um problema, uma das soluções é usar a criptografia de ponta a ponta.
- A camada de apresentação se encarrega que apenas os dados estruturados ou campos que exijam a criptografia seja sobrecarregada de fato

8.1.4 - Primitivas de Serviço na Camada de Apresentação

- Os usuários podem estabelecer sessões através do P-Connect.request fazendo com que a entidade de apresentação emita um S-connect.request
- Bem parecido com o visto na camada de sessão

8.1.4 - Primitivas de Serviço na Camada de Apresentação

- As 3 últimas linhas da figura (próximo slide) mostram as primitivas que foram originadas na camada de apresentação, sua função é que o usuário possa incluir qualquer estrutura de dados complexa
- Cabe a camada de apresentação identificar quais estruturas são necessárias para cada contexto
- Um usuário apresenta uma lista das bibliotecas necessárias, a outra parte pode aceitar ou rejeitar

8.1 - Questões de Design

8.1.4 - Primitivas de Serviço na Camada de Apresentação

OSI Presentation primitive	Request	Indication	Response	Confirm	Meaning
P–CONNECT	X	X	X	X	Establish a presentation connection
P–RELEASE	X	X	X	X	Graceful termination
P–U–ABORT	X	X			User initiated abrupt release
P–P–ABORT		X			Provider initiated abrupt release
P–DATA	X	X			Normal data transfer
P–EXPEDITED–DATA	X	X			Expedited data transfer
P–TYPED–DATA	X	X			Out–of–band data transfer
P–CAPABILITY–DATA	X	X	X	X	Control information data transfer
P–TOKEN–GIVE	X	X			Give a token to the peer
P–TOKEN–PLEASE	X	X			Request a token from the peer
P–CONTROL–GIVE	X	X			Give all the tokens to the peer
P–SYNC–MAJOR	X	X	X	X	Insert a major sync point
P–SYNC–MINOR	X	X	X	X	Insert a minor sync point
P–RESYNCHRONIZE	X	X	X	X	Go back to a previous sync point
P–ACTIVITY–START	X	X			Start an activity
P–ACTIVITY–END	X	X	X	X	End an activity
P–ACTIVITY–DISCARD	X	X	X	X	Abandon an activity
P–ACTIVITY–INTERRUPT	X	X	X	X	Suspend an activity
P–ACTIVITY–RESUME	X	X			Restart a suspended activity
P–U–EXCEPTION–REPORT	X	X			Report of a user exception
P–P–EXCEPTION–REPORT		X			Report of a provider exception
P–ALTER–CONTEXT	X	X	X	X	Change the context

8.2 - Notação de Sintaxe Abstrata 1 (ASN.1)

- Problema:
- “A chave para todo problema de representação, codificação, transmissão e decodificação de estruturas de dados é possuir uma maneira de descrever as estruturas de dados que seja flexível o suficiente para ser útil em uma grande variedade de aplicações e seja padrão o suficiente para que todos possam concordar sobre o que isso significa.”

8.2 - Notação de Sintaxe Abstrata 1 (ASN.1)

- Foi criado como parte do desenvolvimento do Modelo OSI.
- O sufixo “1” indica que foi a primeira a ser padronizada.
- O formato na qual é feita a codificação de estruturas de dados no ASN.1 para fluxo de bits para transmissão é chamado de sintaxe de transferência.

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.1 - Estruturas de Dados

- Em geral, cada aplicação possui algumas coleções de estruturas de dados que são relevantes para suas operações, e que devem ser transmitidas pela rede. Algumas dessas estruturas são usadas em uma grande variedade de aplicações.

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.1 - Estruturas de Dados

- A camada de aplicação possui várias aplicações diferentes, cada qual com uma variedade de complexas estruturas de dados que são transmitidas na forma de APDUs (Application Protocol Data Units). Os campos dessas APDUs costumam ter um tipo (Boolean, Integer, etc...) e em diversos casos, alguns campos podem ser omitidos ou possuir valores default. Devido a essa complexidade, mostrou-se necessária a utilização de um método mais formal para descrever estruturas de dados.

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.1 - Estruturas de Dados

- A ideia do ASN.1 é definir todas os tipos de estruturas de dados necessários para cada aplicação e empacota-las em um módulo (biblioteca).

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.1 - Estruturas de Dados

- Quando um aplicativo quer transmitir uma estrutura de dados, ele pode passa-la para a camada de apresentação, junto com o nome ASN.1 da estrutura. Dessa forma, a camada de apresentação sabe quais são os tipos e tamanhos dos campos da estrutura, e portanto, sabe como codificá-los para transmissão.

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.1 - Estruturas de Dados

- Ao fim da conexão, a camada de apresentação receptora olha o identificador ASN.1 da estrutura de dados (codificado no primeiro byte ou bytes), e assim é possível saber quantos bits há no primeiro campo, quantos no segundo, seus tipos e assim por diante. Com essa informação, a camada de apresentação pode fazer as conversas necessárias do formato externo usado na conexão para o dispositivo interno usado pelo computador receptor.

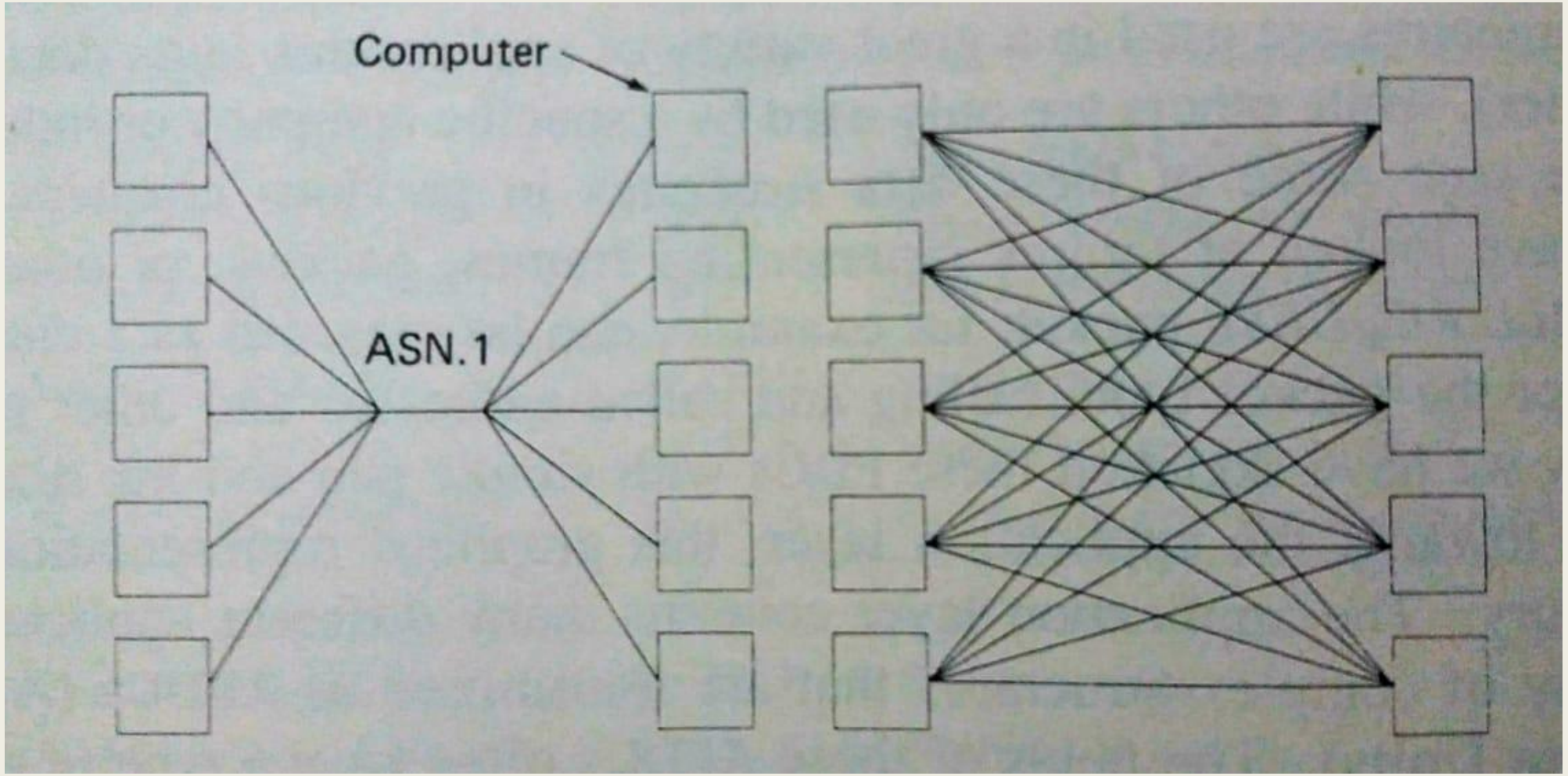
8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.1 - Estruturas de Dados

- Exemplo:
- Se o formato acordado para a transferência de inteiros é o complemento de dois e o receptor usa o complemento de um, a camada de apresentação pode converter todos números inteiros em um complemento antes de passar o APDU ao usuário.

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.1 - Estruturas de Dados



Uso e Desuso da ASN.1

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.2 - Sintaxe Abstrata

- **Os tipos de primitivas da ASN.1:**

Tipo da primitiva	Significado
INTEGER	Inteiro de tamanho arbitrário
BOOLEAN	TRUE ou FALSE
BIT STRING	Lista de 0 ou mais bits
OCTET STRING	Lista de 0 ou mais bytes
ANY	União de todos os tipos
NULL	Absolutamente nenhum tipo
OBJECT IDENTIFIER	Nome de objeto

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.2 - Sintaxe Abstrata

- Os tipos primitivos podem ser combinados para construir tipos mais complexos.

Tipo	Significado
SEQUENCE	Lista ordenada de diversos tipos
SEQUENCE OF	Lista ordenada de um único tipo
SET	Conjunto desordenado de diversos tipos
SET OF	Conjunto desordenado de um único tipo
CHOICE	Qualquer tipo individual tirado de uma determinada lista

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.2 - Sintaxe Abstrata

- É comum nos padrões internacionais definir tipos de dados complexos cujo os campos são opcionais, esses campos não precisam necessariamente serem transmitidos.
- ASN.1 utiliza o conceito de tagging; de forma que qualquer tipo de dados ou campo tenha uma tag que o identifique.
- São permitidas 4 tipos de tags: UNIVERSAL, APPLICATION, PRIVATE e específica ao contexto.
- Exemplo de tag: [APPLICATION 4]
- O uso da tag dispensa a necessidade da transmissão do tipo, assim é utilizada a expressão IMPLICIT para realizar essa supressão.

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.2 - Sintaxe Abstrata

- **Exemplos:**

```
Dinossauro ::= SEQUENCE{
        nome            OCTET STRING,
        tamanho         INTEGER,
        carnivoro       BOOLEAN,
        ossos           INTEGER,
        descoberta      INTEGER
}
```

Sintaxe Abstrata Sem Tagging

```
Dinossauro ::= [PRIVATE 6] SEQUENCE {
        nome[0]         IMPLICIT OCTET STRING
        tamanho[1]      IMPLICIT INTEGER,
        carnivoro[2]    IMPLICIT BOOLEAN,
        ossos[3]        IMPLICIT INTEGER,
        descoberta[4]   IMPLICIT INTEGER OPTIONAL
}
```

Sintaxe Abstrata Com Tagging

8.2 - Notação de Sintaxe Abstrata (ASN.1)

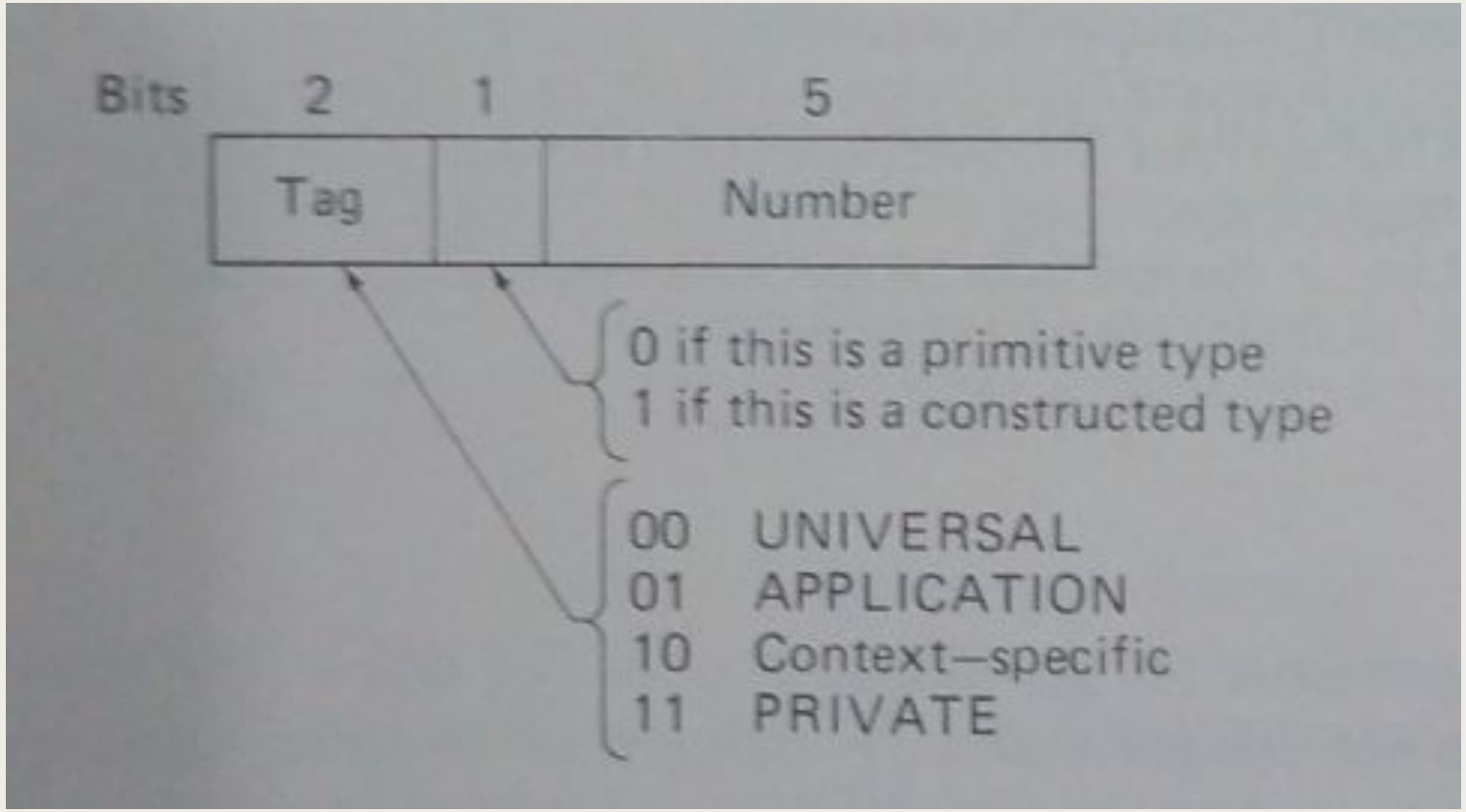
8.2.3 - Sintaxe de Transferência

- O princípio orientador da sintaxe de transferência da ASN.1 é que cada valor transmitido, tanto primitivo quanto construído, consiste em 4 campos:
  1) O identificador
  2) O tamanho do dado em bytes
  3) O Campo de dados
  4) O sinalizador de final de conteúdo.
- Os 3 primeiros sempre estão presentes. O último é opcional o primeiro campo identifica o item que segue

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.3 - Sintaxe de Transferência

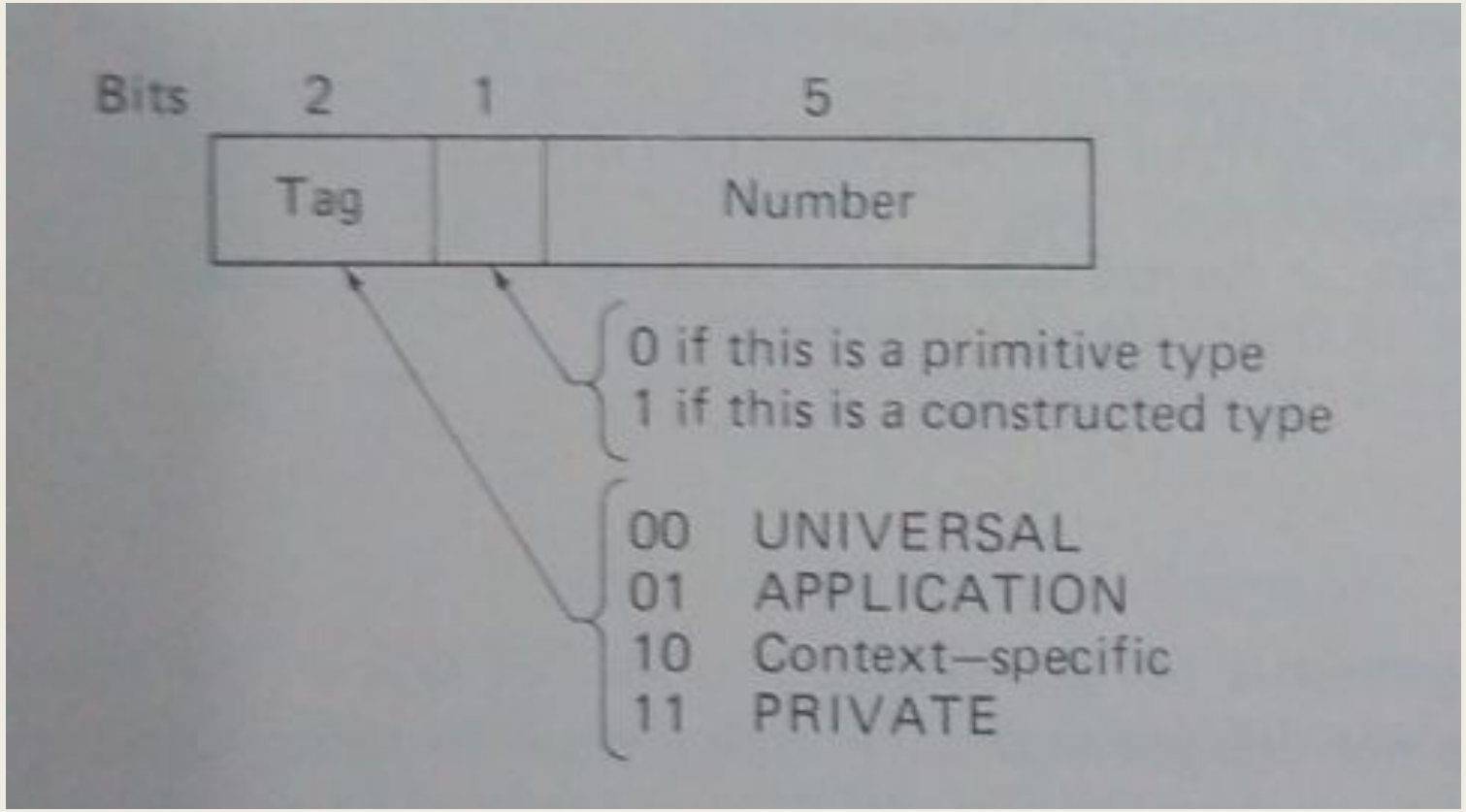
- O primeiro campo identifica o item que segue. Ele possui 3 campos



8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.3 - Sintaxe de Transferência

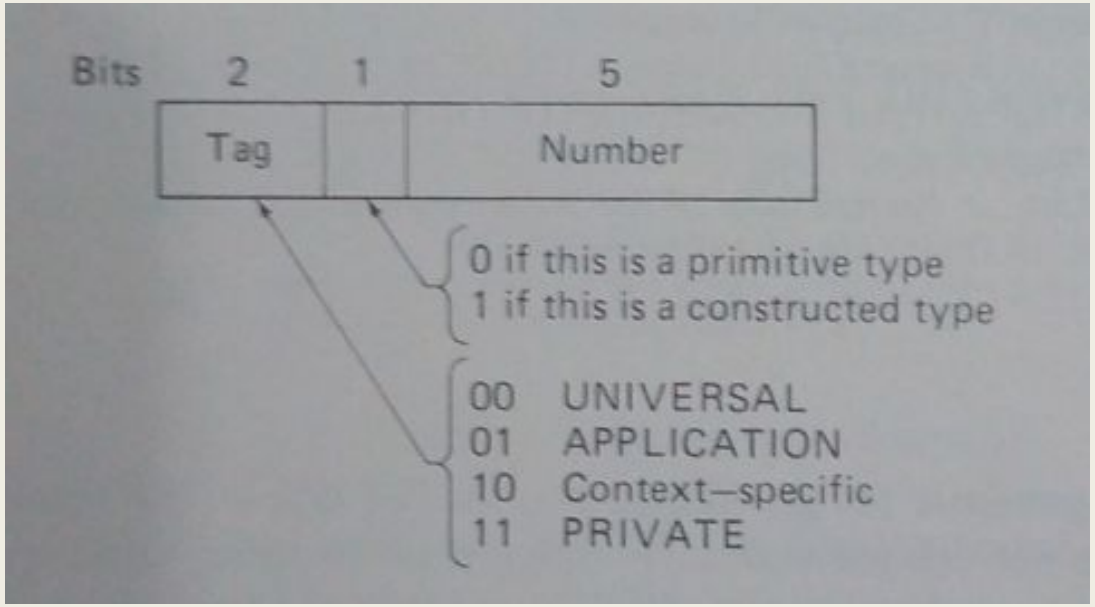
- O primeiro campo identifica o item que segue. Ele possui 3 campos



8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.3 - Sintaxe de Transferência

- O bit 2 identifica o tipo de tag
- os bits de marca são 00, 01, 10 e 11.
- O próximo se ele é ou não primitivo
- Os 5 bits restantes são usados na codificação da tag se estiver no intervalo de 0-30.(11111)



8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.3 - Sintaxe de Transferência

- A codificação do universal

Tag	Meaning
1	BOOLEAN
2	INTEGER
3	BIT STRING
4	OCTET STRING
5	NULL
6	OBJECT IDENTIFIER
7	OBJECT DESCRIPTOR
8	EXTERNAL
16	SEQUENCE and SEQUENCE OF
17	SET and SET OF
18	NumericString
19	PrintableString
20	TeletexString
21	VideotexString
22	IA5String
23	GeneralizedTime
24	UTCTime
25	GraphicString
27	GeneralString

8.2.3 - Sintaxe de Transferência

- Seguindo temos o campo identificador que informa quantos bytes os dados ocupam.
- Para comprimentos menores que 128 bytes são diretamente codificados em 1 byte
- Para tags maiores que 30:
- Usam os múltiplos bytes, com 7 bits de dados por byte e o bit de alta ordem

8.2 - Notação de Sintaxe Abstrata (ASN.1)

8.2.3 - Sintaxe de Transferência

- A codificação do campo de dados depende do tipo dos dados presentes:
- Os Inteiros são codificados em complemento de 2
- Os Booleans são codificados como 0 para false e True aceita qualquer outro valor
- String são codificados como elas mesmas, o problema é so indicar o comprimento
- Octeto de string são codificados com o padrão big endian, da esquerda pra direita
- O valor nulo e codificado apenas setando o campo length como 0

8.3 - Técnicas de Compressão de Dados

- Dados podem ser comprimidos, a fim de melhor utilização de banda.
- Dados podem ser de diferentes domínios (e.g.: Palavras na língua natural, números, sequências de bits)
- Existem basicamente três técnicas para compressão de dados(conjunto de dados finitos, frequência relativa, compressão baseada em contexto)

8.3 - Técnicas de Compressão de Dados

8.3.1 - Codificação Baseada em Símbolos

- Quando o conjunto de dados a ser encaminhado é padronizado, os dados podem ser mantidos em tabelas e transmitidos apenas os índices.
- E.g.: Invés de encaminhar os valores “verdadeiro” e “falso” em forma textual, podem ser transmitidos apenas os valores 0 e 1

8.3 - Técnicas de Compressão de Dados

8.3.2 - Codificação Baseada em Sequência

- Na linguagem natural, certas expressões aparecem mais do que outras, por exemplo, a letra “a” aparece mais que a letra “y”. Baseando-se nisso, é possível montar algoritmos que beneficiem as expressões mais recorrentes.
- Algoritmo de Huffman: Usado também em técnicas de compressão de arquivos como rar e zip.
    1) Ordene as expressões em ordem crescente de frequência.
    2) Pegue as duas primeiras e insira um arco entre elas, e reinsira na lista.
    3) Repita o processo até que a lista vire uma árvore.

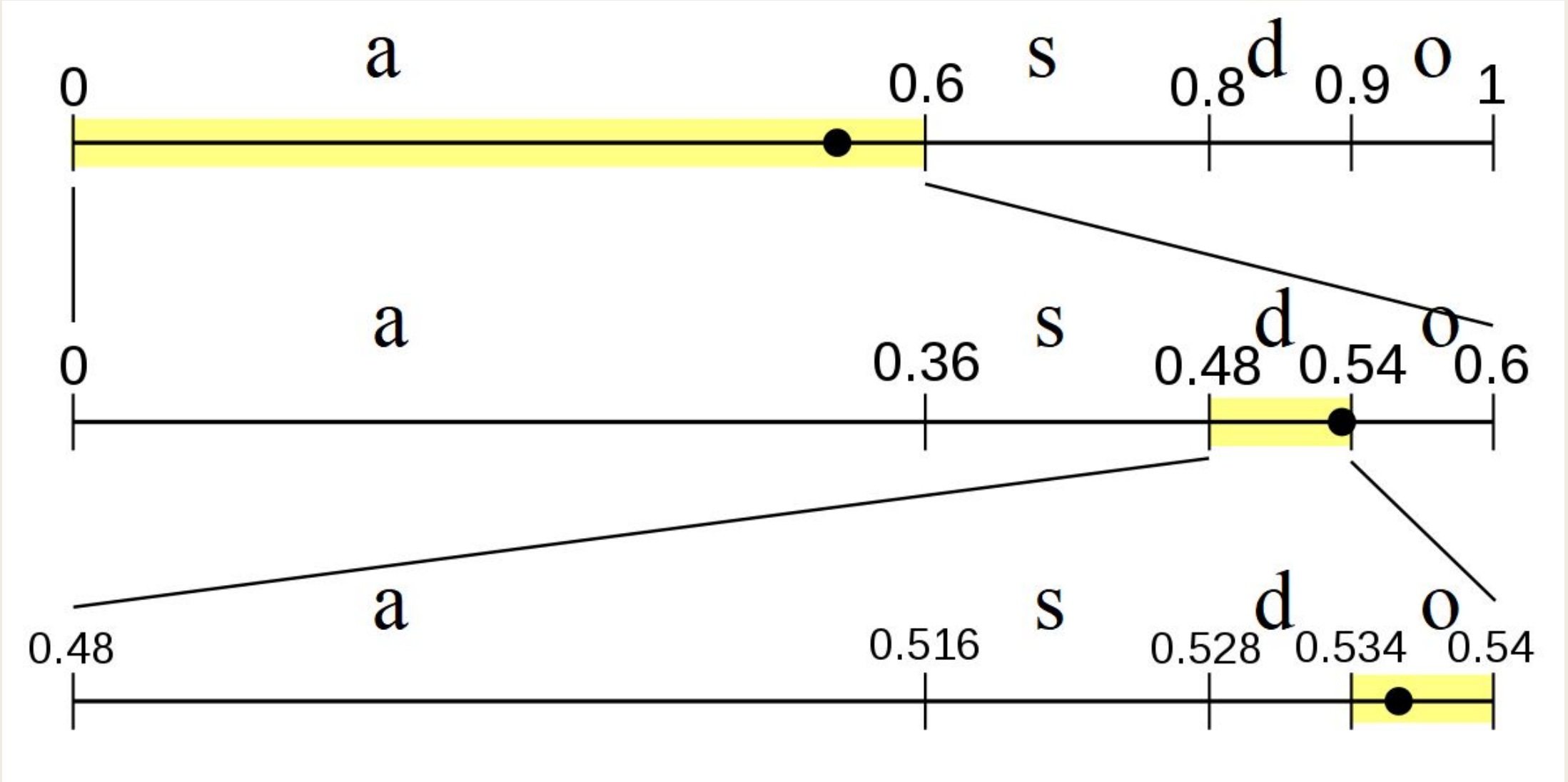
8.3 - Técnicas de Compressão de Dados

8.3.2 - Codificação Baseada em Sequência

- Codificação aritmética: Divida cada possível expressão como uma porção de 0 a 1. Posicione cada expressão da mensagem em sua respectiva porção. Divida a mesma em sub-porções e repita o processo até que a mensagem termine.
- Vantagem é que a mensagem inteira é codificada de uma só vez, mas observe que quanto mais expressões na mensagem, mais dígitos necessários para representar a mesma

8.3 - Técnicas de Compressão de Dados

8.3.2 - Codificação Baseada em Sequência



8.3.3 - Codificação Baseada em Contexto

- A codificação baseada em frequência já representa uma aproximação ao mundo real, mas uma melhoria ainda pode ser feita considerando o contexto em que cada expressão se encontra. Por exemplo, uma vogal provavelmente precede uma consoante.
- Código Baudot usa mais de uma tabela de símbolos para representar instruções e sequências especiais para transicionar entre as tabelas.
- Código “run length” transforma longas sequências de 0 na distância entre dois bits 1.

8.4 - Criptografia

8.4.1 - Criptografia Tradicional

- Terminologia:
  - Criptografia: arte de planejamento de funções de tal modo que possam ser aplicados a textos para que esses sejam inteligíveis apenas a quem detém a chave.
  - Criptoanálise: arte de “quebrar” tais códigos.
  - Criptologia: a junção de criptografia e criptoanálise

8.4 - Criptografia

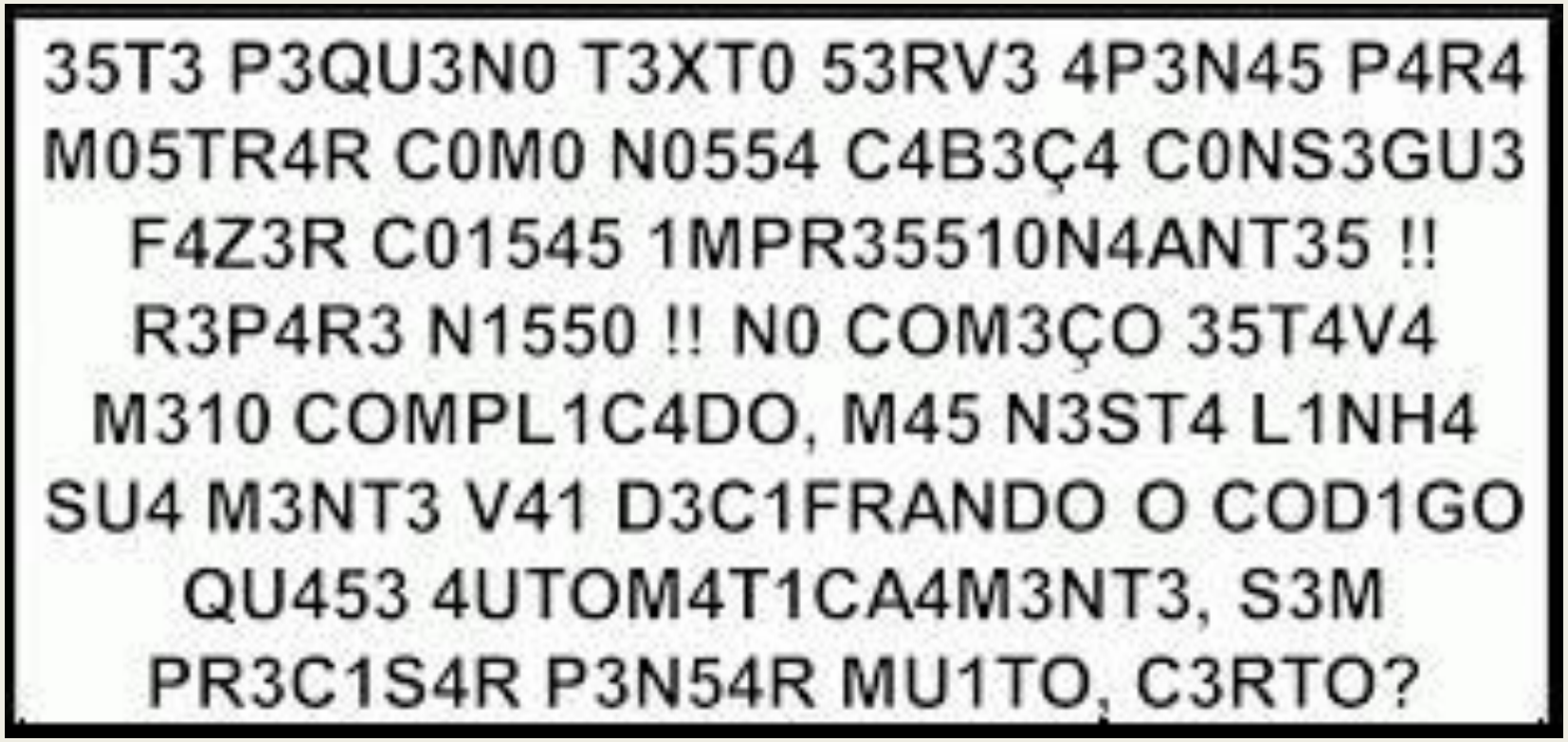
8.4.1 - Criptografia Tradicional

- Historicamente, a criptologia teve um avanço significativo nas guerras já que a comunicação devia ser sigilosa, os equipamentos deviam ser pequenos e a troca de chave devia ser simples e rápida.
- Uma regra básica da criptologia é que normalmente o criptoanalista conhece o método de codificação. Invés de ter que mudar toda a função criptográfica, podemos mudar apenas uma pequena string conhecida como chave a fim de selecionar uma das possíveis encriptações.
- Normalmente existem diversas possibilidades de quebrar uma criptografia, mas o criptoanalista corta algumas fazendo uso do conhecimento do contexto em questão.

8.4 - Criptografia

8.4.1 - Criptografia Tradicional

- Existem basicamente duas técnicas:
- Substituição: Consiste em substituir uma letra por outra



8.4 - Criptografia

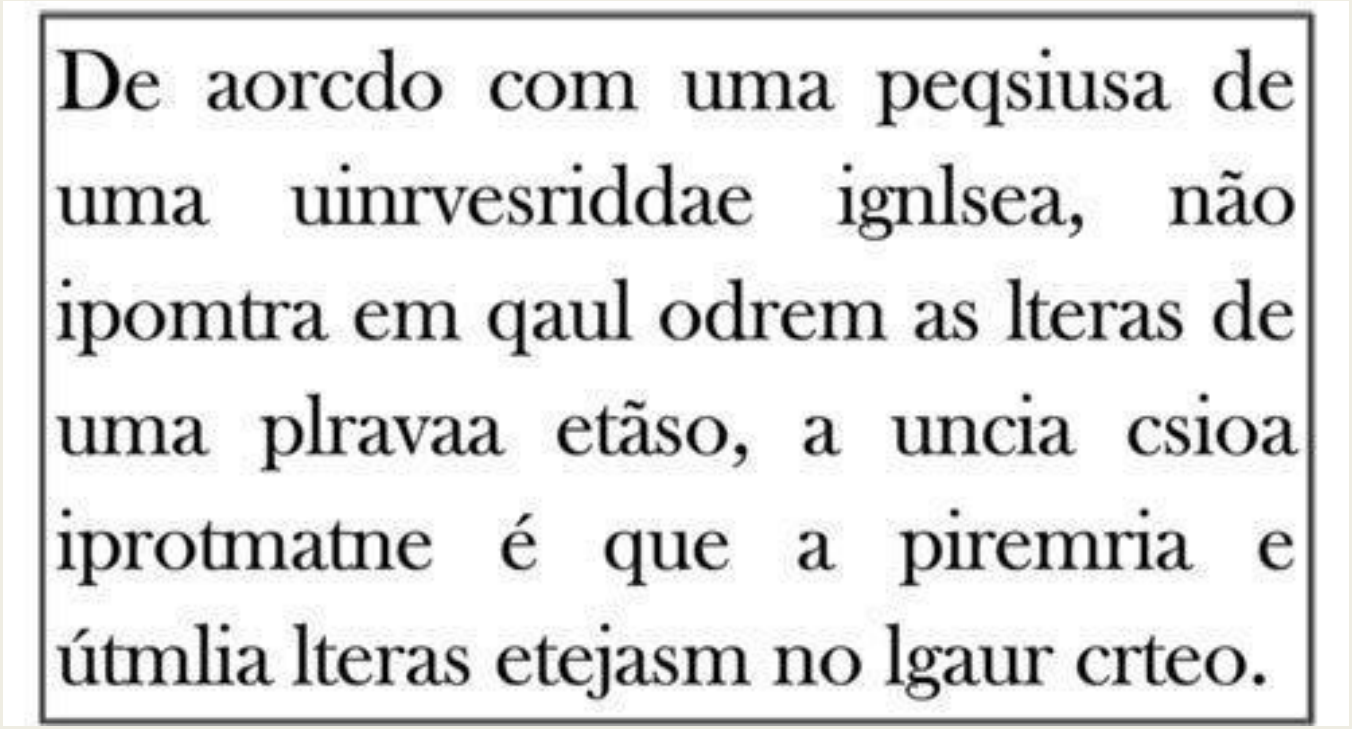
8.4.1 - Criptografia Tradicional

- Cifra de César: consiste em substituir cada letra X por uma letra X+3 posições módulo 26.
- De fato, o algoritmo de César pode ser generalizado para qualquer d posições. Logo pode ser construído 26 tabelas uma para cada d possível.
- Usando-se uma chave como base. Podemos construir a cifra de Vigenère

8.4 - Criptografia

8.4.1 - Criptografia Tradicional

- Existem basicamente duas técnicas:
- Transposição: permutação das letras na frase



8.4.1 - Criptografia Tradicional

- Seja uma chave k de tamanho |k|, então uma cifra de transposição arranja a mensagem em linhas de tamanho |k| e tem como saída as colunas reordenadas com base em cada um dos dígitos da chave k

8.4 - Criptografia

8.4.2 - O Padrão de Criptografia de Dados

- Criptografia Moderna utiliza transposição e substituição com algoritmos mais complexos;
- Utilização de hardware para transposição e substituição, P-box e S-box, respectivamente.

8.4 - Criptografia

8.4.2 - O Padrão de Criptografia de Dados

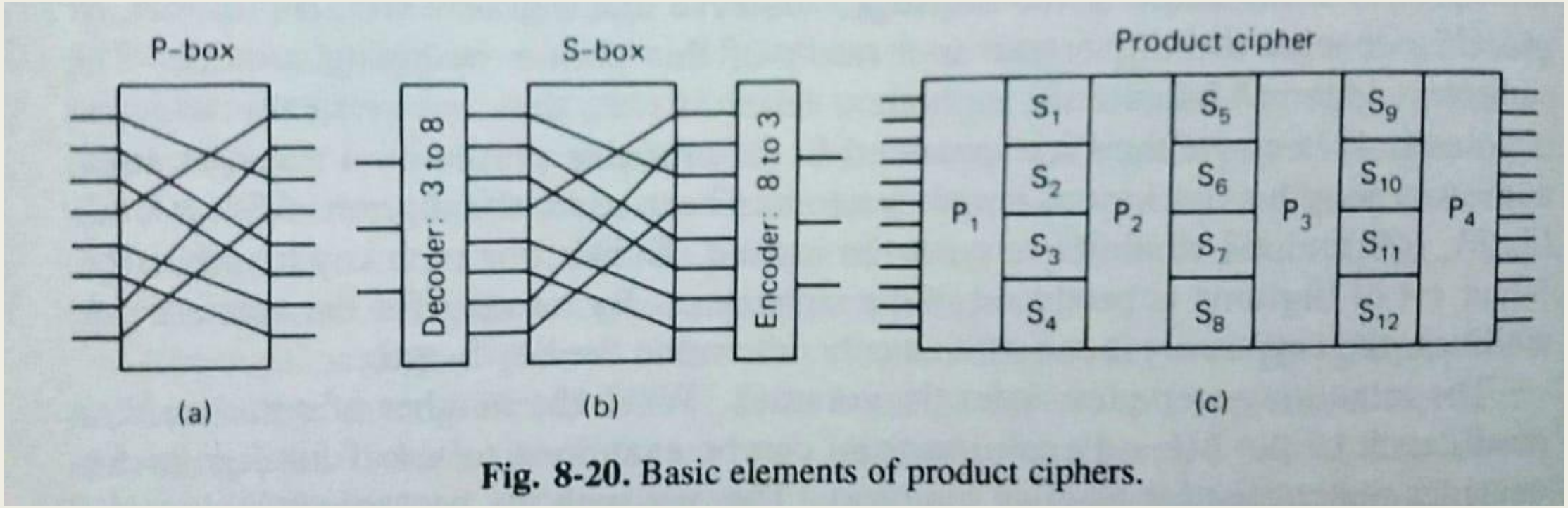


Fig. 8-20. Basic elements of product ciphers.

8.4 - Criptografia

8.4.2 - O Padrão de Criptografia de Dados

- Em janeiro de 1977, o governo dos EUA adotou como padrão de informações não classificadas o produto desenvolvido pela IBM, conhecido como Data Encryption Standard(DES);

- DES possui 64 bits de entrada, 56 bits que compõe a chave, 19 estágios distintos.

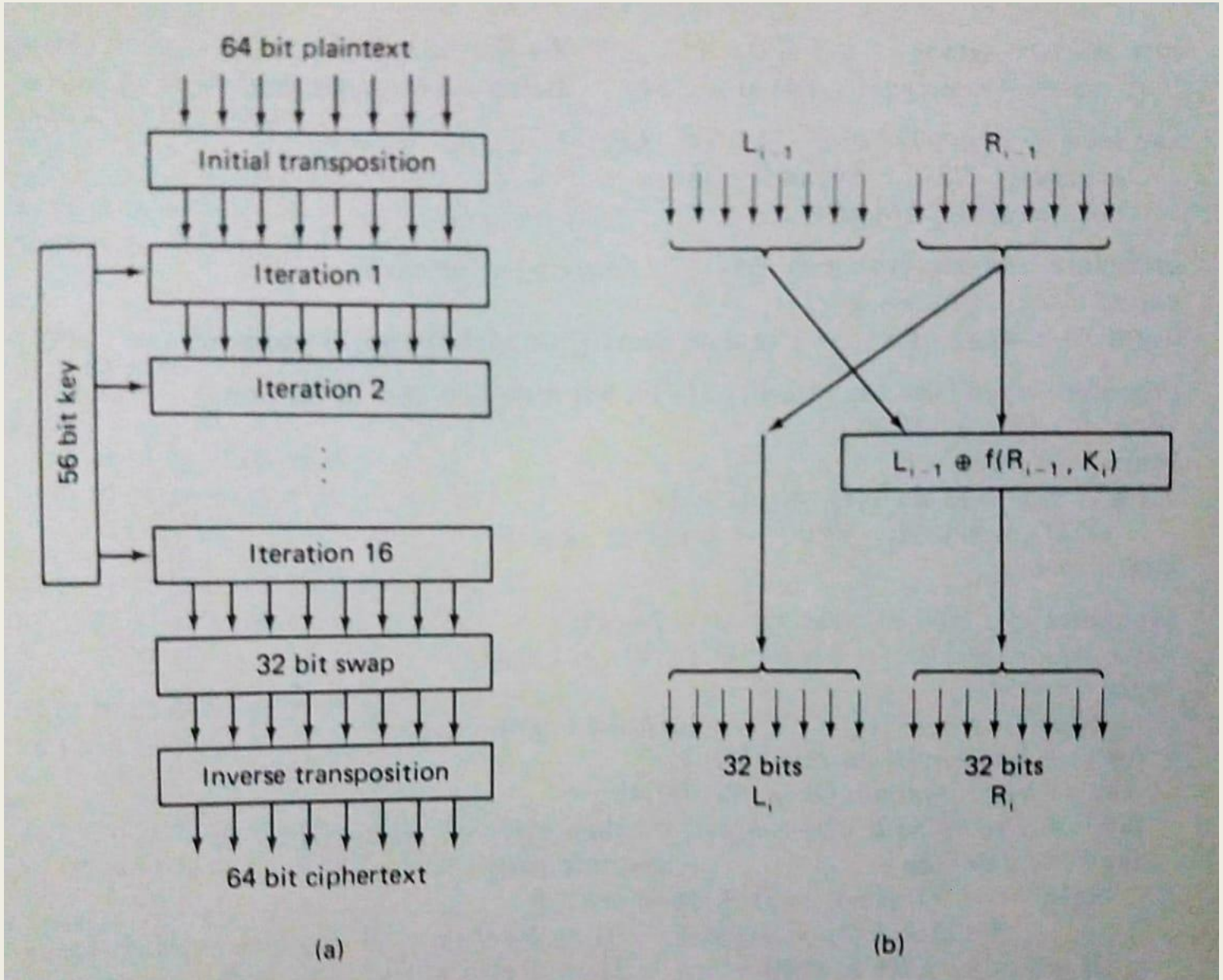


Fig. 8-21. The data encryption standard. (a) General outline. (b) Detail of one iteration.

8.4.2 - O Padrão de Criptografia de Dados

- A função é composta de 4 estágios:
  1) Com uma regra de transposição e duplicação, expande 32 bits(R_{i-1}) em 48 bits(E);
  2) Faz um OU EXCLUSIVO entre E e K_{i-1};
  3) Divide o resultado em 8 grupos de 4 bits cada, colocando em S-Box diferentes;
  4) Coloca o resultado em uma P-Box.

8.4 - Criptografia

8.4.2 - O Padrão de Criptografia de Dados

- Para obtermos resultados diferentes, antes de cada iteração, utiliza de transposição e rotação na chave(K_{i-1});
- Null Cipher, método para dificultar a quebra da criptografia, utiliza de ruídos, ou seja, mensagens aleatórias. Porém sobrecarrega a largura de banda;

8.4 - Criptografia

8.4.2 - O Padrão de Criptografia de Dados

- Stream Cipher, método mais complicado de operar, pois utiliza registradores para captar entrada e saída;

- A utilização de terminais é vantajosa, já que, não é preciso coletar 8 caracteres depois de emitir o texto criptografado, economizando tempo.

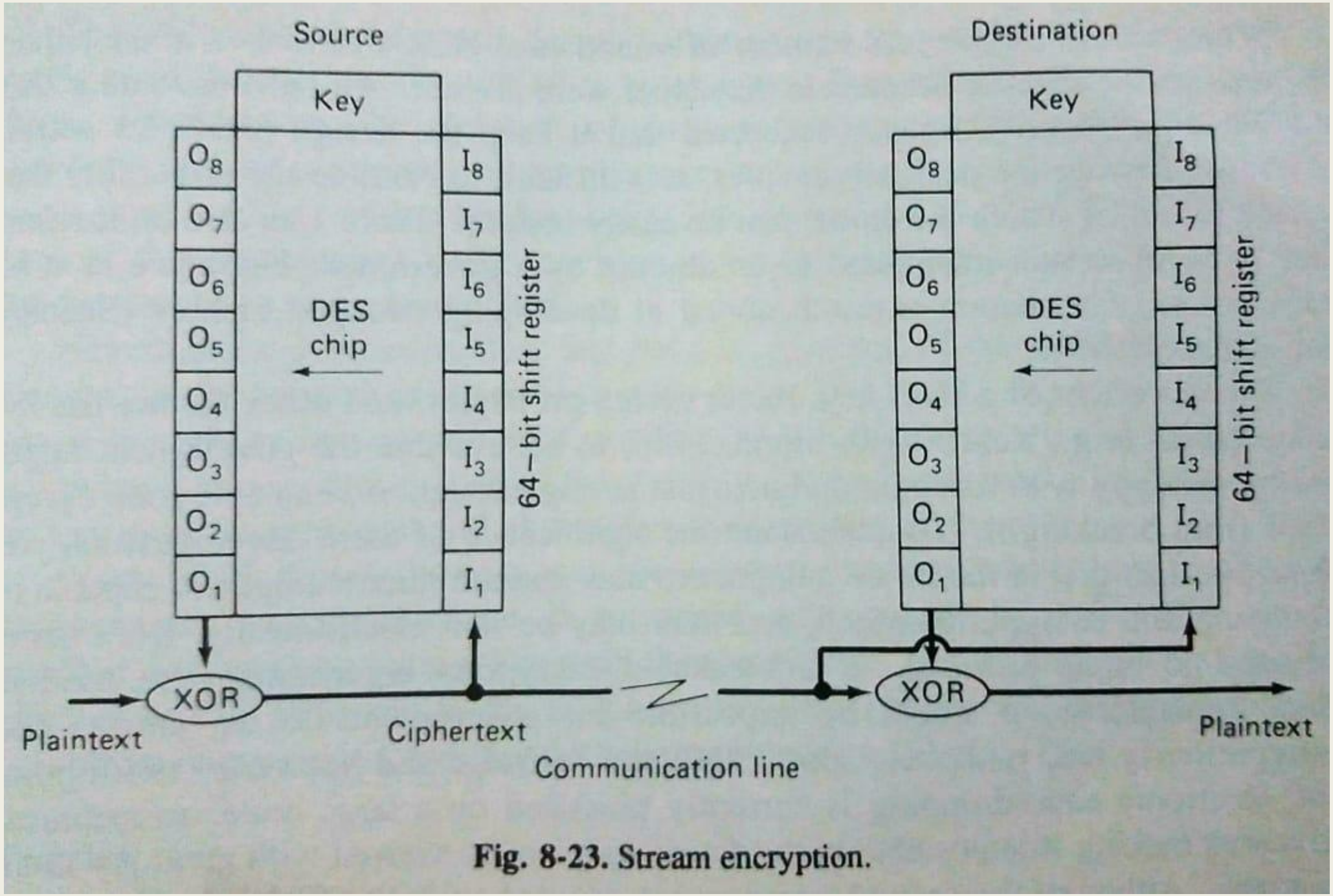


Fig. 8-23. Stream encryption.

8.4 - Criptografia

8.4.2 - O Padrão de Criptografia de Dados

- DES foi criticado desde que foi lançado;
- A chave era considerada muito curta, sendo que a desenvolvida pela IBM continha 128 bits;
- O design era mantido em segredo, o que dificultava a quebra, menos para o próprio Governo.

8.4 - Criptografia

8.4.3 - O Problema da Distribuição de Chave

- Receptor de uma mensagem precisa usar a mesma chave para decriptá-la que o transmissor usou para encriptá-la
- Tradicionalmente pares de chaves idênticas eram geradas e enviadas aos seus destinos por correio pessoal
- Método insatisfatório em casos reais (Exemplo: Banco)

8.4 - Criptografia

8.4.3 - O Problema da Distribuição de Chave

- Uma solução para o problema é usar uma hierarquia de chave
- Organizações escolhem chave mestra e distribuem para cada escritório (por correio pessoal)
- Escritórios são agrupados em regiões, e o escritório central de cada região escolhe uma chave regional
- As chaves regionais são encriptadas utilizando a chave mestra e distribuídas pela rede

8.4 - Criptografia

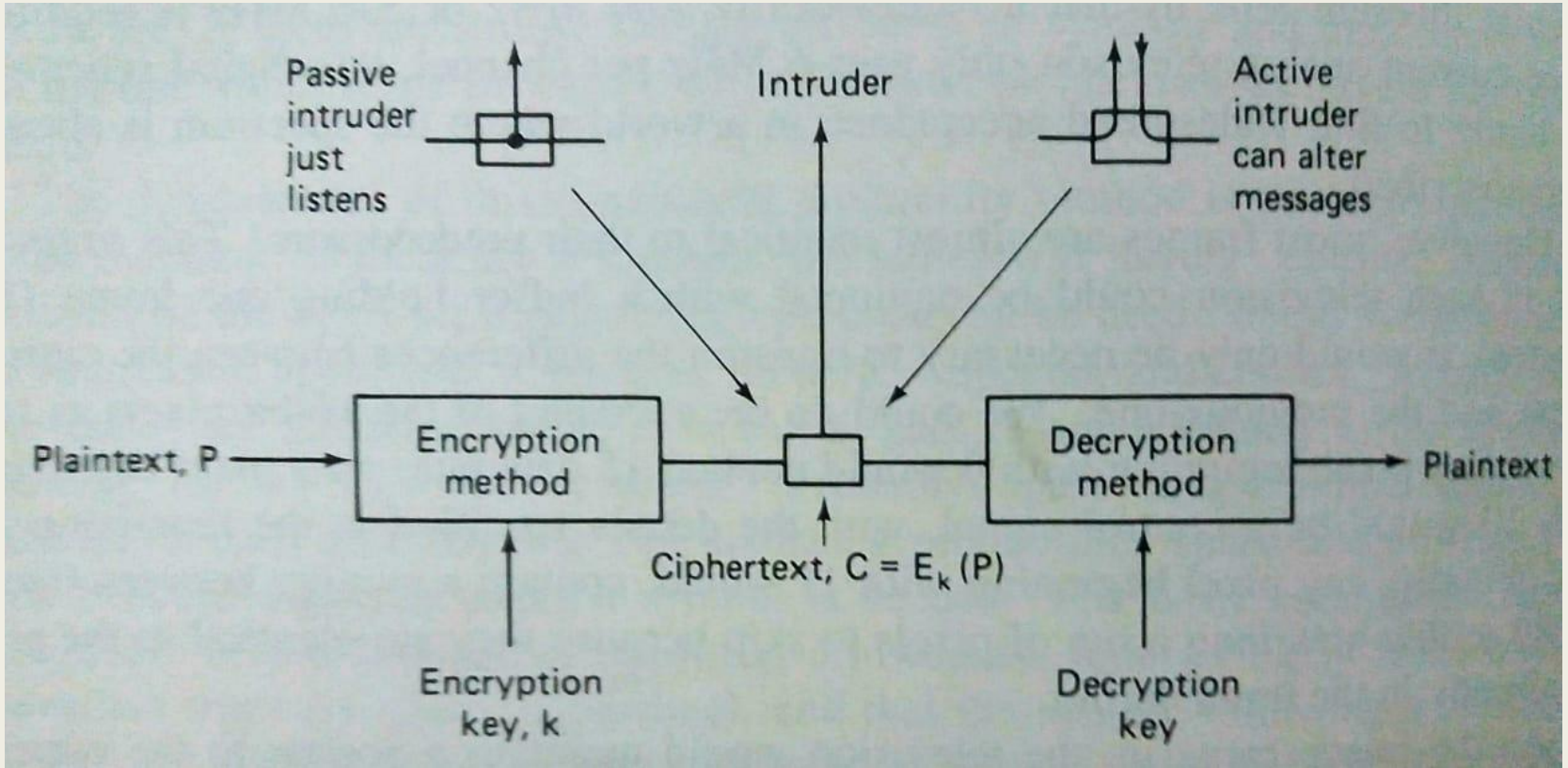
8.4.3 - O Problema da Distribuição de Chave

- Quando dois escritórios querem comunicar, utilizam chave de sessão encriptada pela chave regional
- Continua sendo um método inefetivo, pois precisa de transporte físico das chaves por fora da rede e não é possível realizar a comunicação com outras organizações

8.4 - Criptografia

8.4.3 - O Problema da Distribuição de Chave

- Puzzles: Um criptograma que tem a intenção de ser quebrado
- Método de Merkle



8.4 - Criptografia

8.4.3 - O Problema da Distribuição de Chave

- Supondo que A inicia uma conversa com B
- A mensagem é enviada criptografada seguida de dezenas de milhares de puzzles
- No ponto de vista do intruso é complicado definir qual o melhor puzzle para começar a trabalhar, então ele vai precisar escolher em ordem aleatória
- Isso aumenta bastante o tempo necessário para encontrar a chave

8.4 - Criptografia

8.4.3 - O Problema da Distribuição de Chave

- Proteção de Chave:
- É importante esconder a chave de si mesma
- Sistema de Shamir: Utilização de polinômios
- Exemplo:
  - p(x) = (a3 * (x^3)) + (a2 * (x^2)) + (a1 * x) + a0
- Cada funcionário recebe um ponto (x, p(x))
- 4 pontos formam o polinômio
- a0 é a chave
- a1, a2 e a3 são valores aleatórios

8.4.4 - Criptografia de Chave Pública

- Método de Merkle também é ineficiente
- Método de Diffie e Hellman: Preocupa-se com os algoritmos de encriptação e decriptação do que com manter as chaves escondidas
- Considerando D como algoritmo de decriptação e E como algoritmo de encriptação, e P sendo a mensagem
- Requerimentos:
  1) D(E(P)) = P
  2) É extremamente difícil deduzir D de E
  3) E não pode ser quebrado por ataque de escolha de plaintext
- Obedecendo os requerimentos E e a chave podem ser

8.4.4 - Criptografia de Chave Pública

- **Algoritmo de MIT:**
  1) **Escolher dois números primos muito grandes, p e q, cada um maior do que 10^100**
  2) **Computar n = (p*q) e z = (p-1)*(q-1)**
  3) **Escolher um número relativamente primo à z, chamando-o de d**
  4) **Achar e tal que e*d = 1 mod z**
- **Para encriptar uma mensagem P faz C = (P^e) mod n**
- **Para decriptar C faz P = (C^d) mod n**
- **Complicado fatorar números muito grandes**

8.5 - Exemplos da Camada de Apresentação

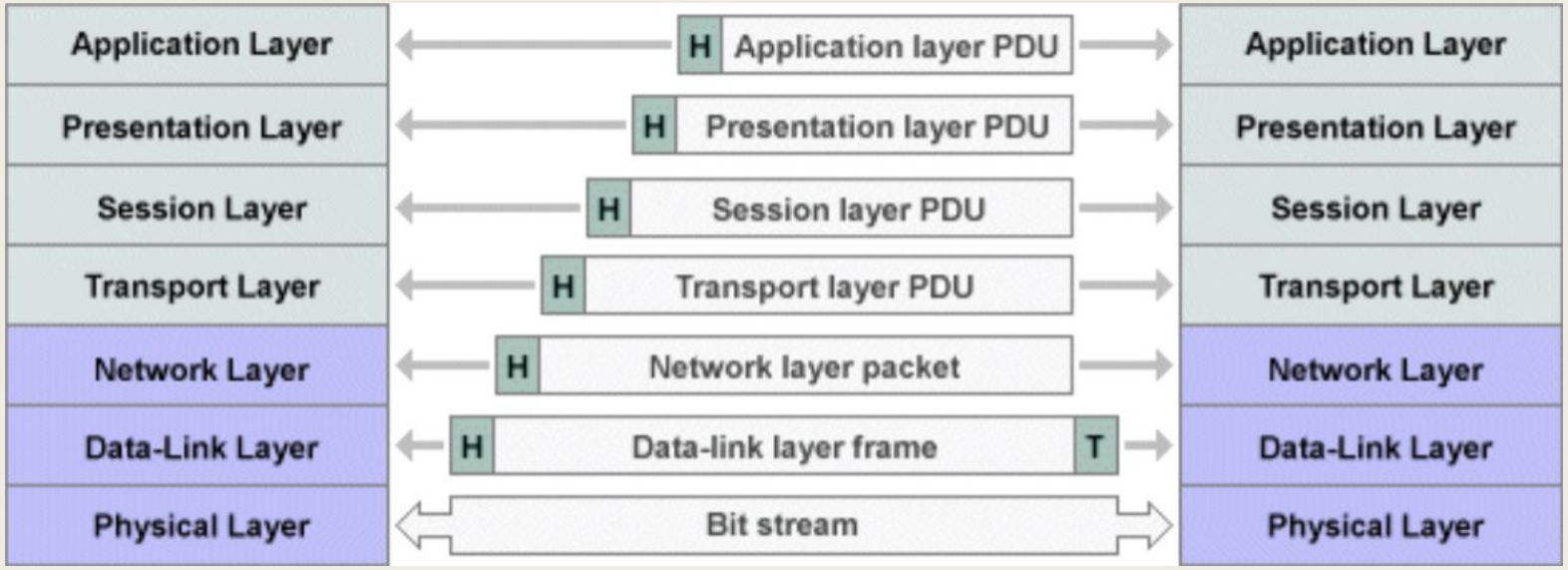
8.5.1 - Camada de Apresentação em Redes Públicas

- Redes Públicas que implementa a camada de sessão também implementa camada de apresentação;
- O único serviço real da camada de apresentação é negociação e gerenciamento do contexto de apresentação.

8.5 - Exemplos da Camada de Apresentação

8.5.1 - Camada de Apresentação em Redes Públicas

- O padrão utilizado é ASN.1, utilizado para representar e transmitir estruturas de dados ou APDUs.



8.5 - Exemplos da Camada de Apresentação

8.5.1 - Camada de Apresentação em Redes Públicas

- A camada de apresentação basicamente disponibiliza os serviços da camada de sessão para a camada de aplicação;
- O protocolo da camada de apresentação é simples, pelo fato dele invocar o serviço na camada de sessão(envia PDUs).

8.5 - Exemplos da Camada de Apresentação

8.5.1 - Camada de Apresentação em Redes Públicas

- As PDUs utilizado pela camada de apresentação se dividem em:
  - *Estabelecimento de conexão;*
  - *Liberação anormal;*
  - *Transferência de dados;*
  - *Gerenciamento de contexto.*

PDDU	Name	Request	Indication	Response	Confirm
CP	Connect Presentation	X	X		
CPA	Connect Presentation Accept			X	X
CPR	Connect Presentation Reject			X	X
ARU	Abnormal Release, User initiated	X	X		
ARP	Abnormal Release, Provider initiated		X		
TD	Transfer Data	X	X		
TE	Transfer Expedited	X	X		
TTD	Transfer Typed Data	X	X		
TC	Transfer Capability	X	X		
TCC	Transfer Capability Confirm			X	X
AC	Alter Context	X	X		
ACA	Alter Context Acknowledge			X	X
RS	Resynchronize	X	X		
RSA	Resynchronize Acknowledge			X	X

Fig. 8-28. The presentation PDUs and their associated primitives.

8.5 - Exemplos da Camada de Apresentação

8.5.2 - Camada de apresentação em ARPANET

- ARPANET não possui camada de apresentação;
- Não existe uma maneira geral de passar estruturas de dados arbitrárias entre máquinas incompatíveis;
- A aplicação deve definir seus próprios padrões;

8.5 - Exemplos da Camada de Apresentação

8.5.3 - Camada de apresentação em MAP e TOP

- MAP e TOP suportam as funções básicas da camada de apresentação OSI, estabelecendo conexões e gerenciando múltiplos contextos;
- As primitivas associadas a comutadores de contexto nos limites de atividade não são suportadas porque o conceito inteiro de atividades não é suportado na camada de sessão;
- O Uso da ASN.1 é obrigatório;

8.5 - Exemplos da Camada de Apresentação

8.5.4 - Camada de apresentação em USENET.

- USENET não possui camada de apresentação;
- Cada aplicativo possui um conhecimento interno dos formatos externos necessários e apenas faz todas as conversões internamente antes de oferecer qualquer mensagem de transmissão.

Bibliografia

- Andrew S. Tanenbaum - “Computer Networks”, 2nd edition Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 1988, ISBN: 0-13-162959-X, Capítulo 2