INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA
PARAÍBA

Segurança de Redes

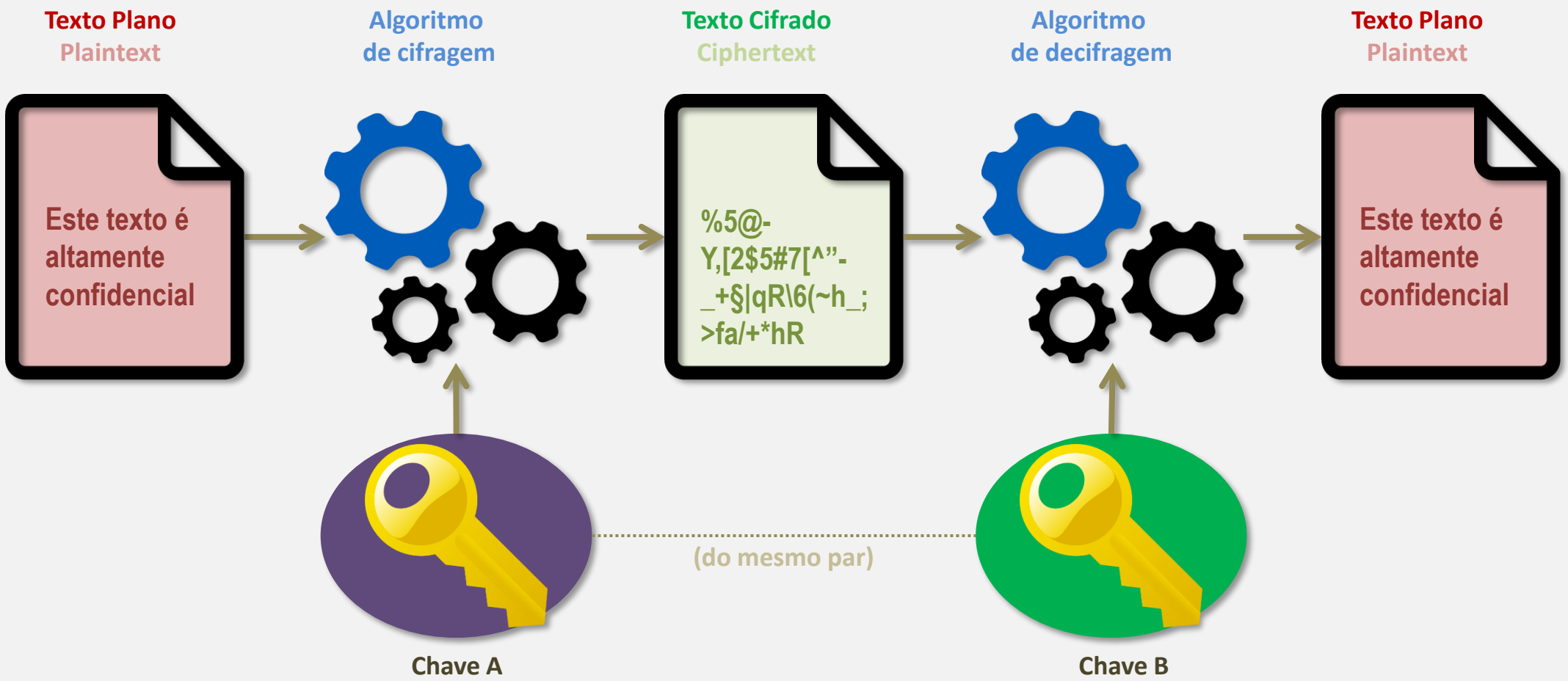
Criptografia Assimétrica

Dênio Mariz

Setembro 2020

Prof. Dênio Mariz, IFPB

Criptografia Assimétrica



Sistemas Assimétricos

- Criptografia de Chave Pública
- 1976 - Whitfield Diffie e Martin Hellman
- Consiste em usar um par de chaves:
  - chave pública e chave privada
  - Chave pública deve ser divulgada (é de conhecimento público)
  - Chave privada deve ser mantida em segredo pelo dono
- Problema de distribuição de chaves eliminado
- Notação usada: KP=chave pública, KR=chave privada

Sistemas Assimétricos

- Regra:
  - Tudo que for cifrado com a chave privada (KR) somente pode ser decifrado pela chave pública (KP) e vice-versa
  - Se Y = E(X, KR), então X = D(Y, KP)
  - Se Y = E(X, KP), então X = D(Y, KR)
  - É computacionalmente inviável deduzir **KR** a partir de **KP**
  - Ou seja, o conhecimento da chave pública não permite a descoberta da chave privada
- É computacionalmente viável calcular um par **{KR, KP}** que satisfaça os requisitos acima

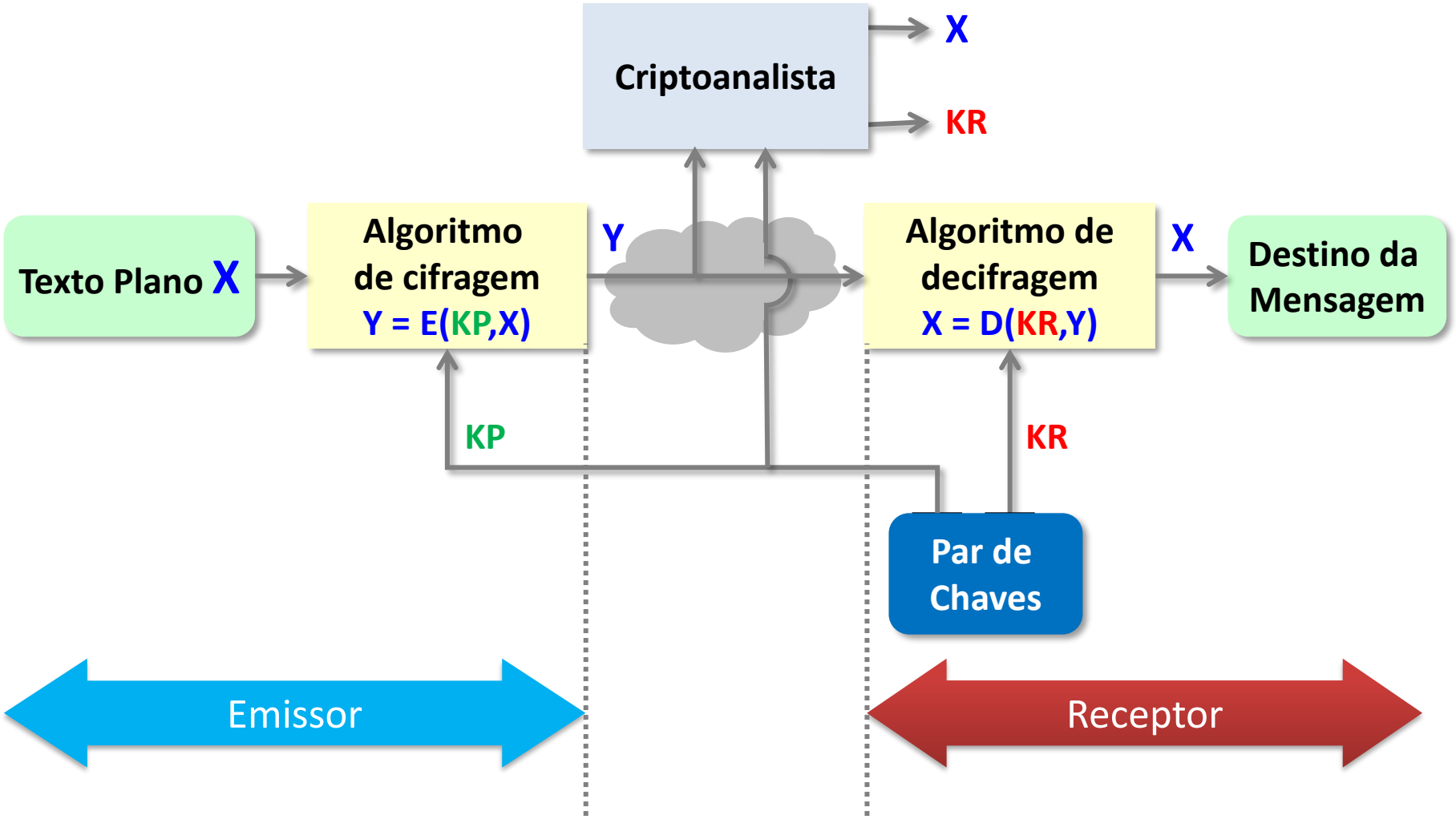
Sistemas Assimétricos

- Confidencialidade
  - Alice cifra: $Y = E(KP_{bob}, X)$
  - Bob Decifra: $X = D(KR_{bob}, Y)$

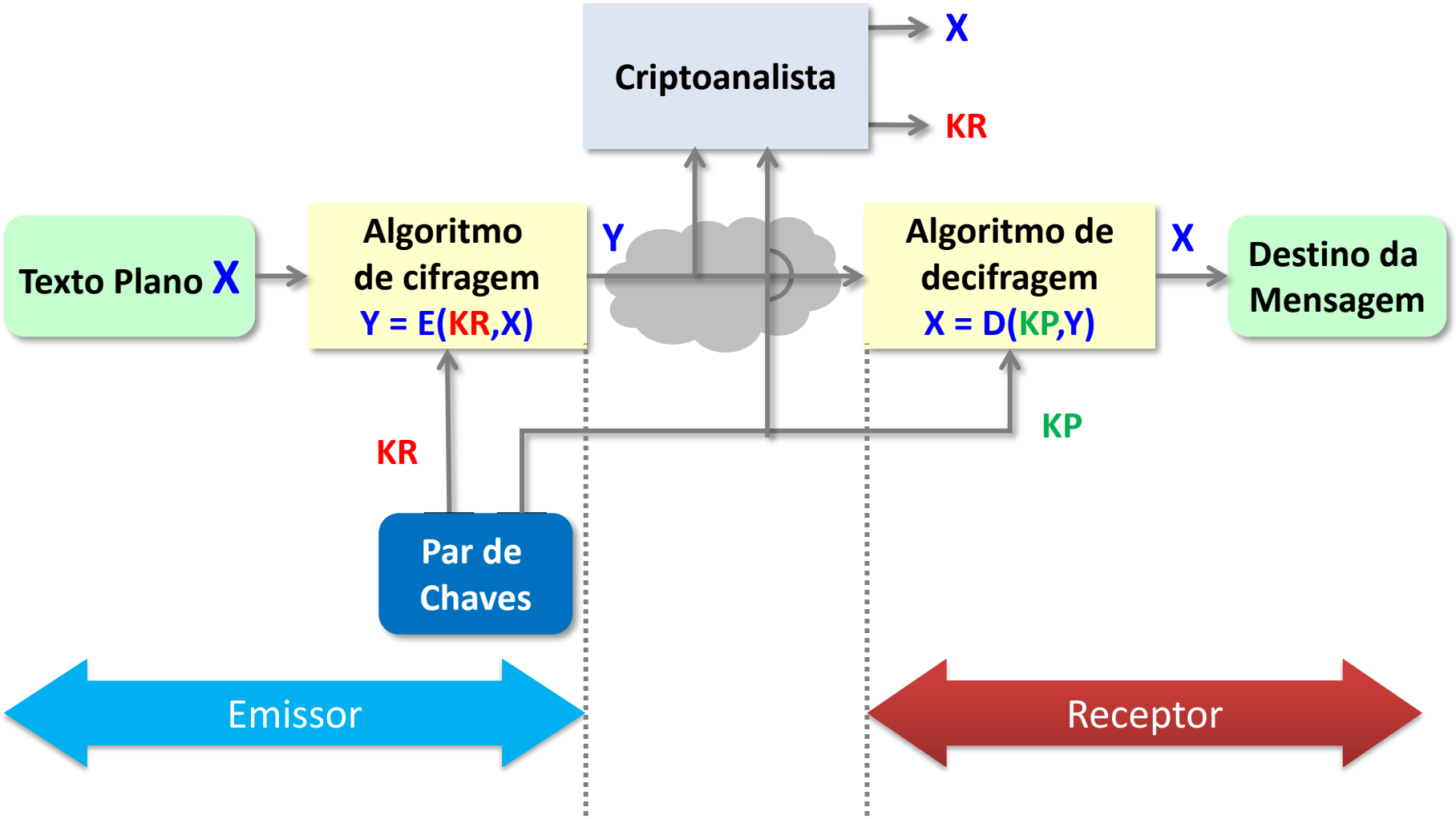
Alice=emissor
Bob=receptor

- Autenticação da origem
  - Alice cifra: $Y = E(KR_{alice}, X)$
  - Bob Decifra: $X = D(KP_{alice}, Y)$

Confidencialidade em Sistemas Assimétricos



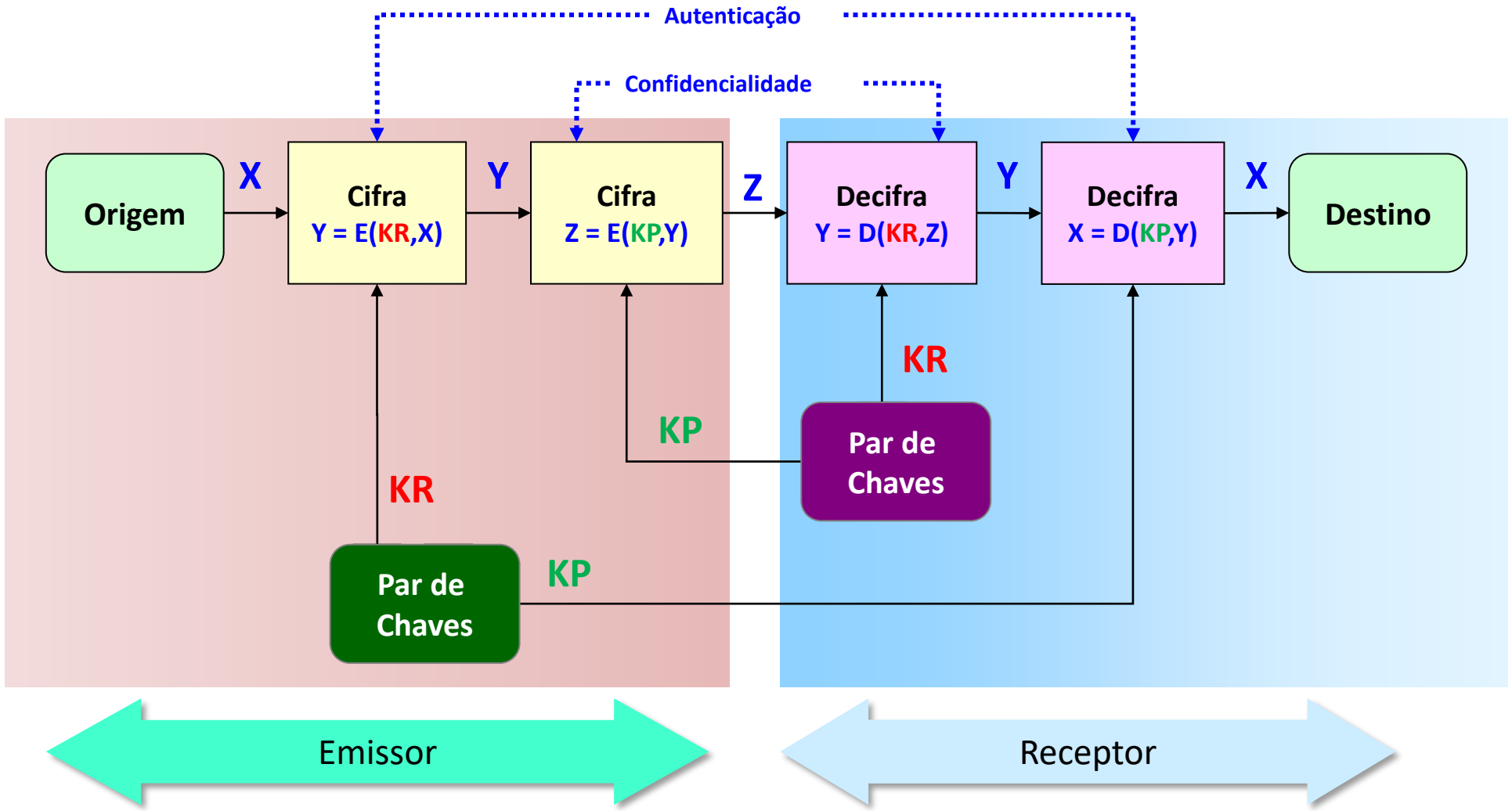
Autenticação em Sistemas Assimétricos



Sistemas Assimétricos

- Como ter ambos Autenticação e Confidencialidade?
  - Usando a chave privada de Alice (emissora) e a chave pública de Bob (receptor)
- Alice Cifra com sua chave privada (autenticação) e **depois** com a chave pública de Bob (confidencialidade)
  - Alice Cifra: $Z = E(KP_{bob}, E(KR_{alice}, X))$
  - Bob decifra: $X = D(KP_{alice}, D(KR_{bob}, Z))$

Autenticação+Confidencialidade em Sistemas Assimétricos



INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA
PARAÍBA

Criptografia Assimétrica
Algoritmo RSA

Dênio Mariz

Setembro, 2020

Prof. Dênio Mariz, IFPB

O Algoritmo RSA

- Algoritimo de chave pública (ou seja, assimétrico)
- Chaves pública e privada criadas a partir de números primos (p,q)
- Ron Rivest, Adi Shamir e Leonard Adelman

Fundamentos: Primos

- Um número p é primo só é divisível exatamente por dois números: 1 e ele mesmo
  - 1 não é primo por definição
  - Alguns primos: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, …
- Todo inteiro é formado por multiplicação de fatores primos
  - $N=p_1*p_2*p_3\ldots*p_n$
- Exemplos
  - 10 = 2*5 ($p_1=2, p_2=5$)
  - 1432 = 2*2*2*179 ($p_1=2, p_2=2, p_3=2, p_4=179$)

Prof. Dênio Mariz, IFPB

Fundamentos: Primos relativos

- Dois números p e q são relativamente primos quando não têm fatores comuns (maior divisor comum=1)
- Exemplo: 6 e 35 são primos relativos
  - Divisores de 6 são 2, 3
  - Divisores de 35 são 5, 7
  - 6 e 27 não são primos relativos porque ambos são divisíveis por 3

Fundamentos - Módulo

- Módulo
  - Se a mod n = r então a = $\lfloor a/n \rfloor$*n + r

- $\lfloor a/n \rfloor$ = divisão inteira de a por n

RSA – Criando um par de Chaves

➔ Criando Chaves pública e privada

- Selecione dois primos p e q
- Calcule calcule n = pq e z = (p-1)(q-1)
- Escolha um inteiro e < z que seja primo relativo de z
- Calcule um d < z tal que ed mod z = 1
- Chave Pública é KP={e,n}
- Chave Privada é KR={d,n}

Protegendo a Chave Privada

- As chaves Pública e Privada são derivadas de p, q, e, d
  - Não são escolhas memorizáveis como em sistemas simétricos
- Chaves Pública e Privada são armazenadas em **arquivos**
  - Existem formatos específicos
- A chave privada deve ser protegida por criptografia simétrica
  - Ex: Cifrar o arquivo da chave privada com AES 128 ou AES 256

RSA – Cifrando e decifrando

- Chave Pública é KP={e,n}
- Chave Privada é KR={d,n}

- Para cifrar:
  - Condição para Texto de Entrada: $X < n$
  - Texto Cifrado: $Y = X^e \bmod n$
- Para decifrar
  - Texto Entrada: Y
  - Texto Decifrado: $X = Y^d \bmod n$

Exemplo do RSA: Criando o par de chaves

1. Bob escolhe números primos $p = 5$ e $q = 11$
2. Bob calcula $n = pq = 55$ e $z = (p-1)(q-1) = 4*10 = 40$
3. Bob escolhe $e=3$ primo relativo de $z=40$
4. Bob escolhe um d tal que $ed \bmod z = 1$. Então, $d = 27$
   (veja que $3*27 \bmod 40 = 81 \bmod 40 = 1$)
5. A chave pública de Bob é o par de números $KP=(3, 55)$, e sua chave privada é o par de números $KR=(27, 55)$.

Exemplo do RSA: Envio de mensagem confidencial

6. Alice quer mandar uma mensagem para Bob
7. Bob envia sua chave pública para Alice (guarda a sua chave privada)
8. Alice tem uma mensagem $X=25$
9. Ela cifra $Y=X^e \bmod n$, ou $Y=25^3 \bmod 55 = 15625 \bmod 55 = 5$
10. Alice envia $Y=5$ para Bob.
11. Bob recebe Y e decifra $X=Y^d \bmod n$, ou $X=5^{27} \bmod 55 = 74505805969238281 25 \bmod 55 = 25$, obtendo a mensagem original $X=25$

Números Primos Usados na prática

- No exemplo, usamos números primos pequenos
- Mas RSA usa números primos de 1024 bits (309 dígitos).
- Exemplo:
  - p=11939328203597295838024350196951597430328245644827110011554786980321237763076600086965209349704443440631796281236701977864132319056169743441999921354610407925090313825973708244702472866333159179400043468028439630383152475195072835910738677271008877637121130650089600357560287359852871731538780772888059516765 1

Como “quebrar” o RSA

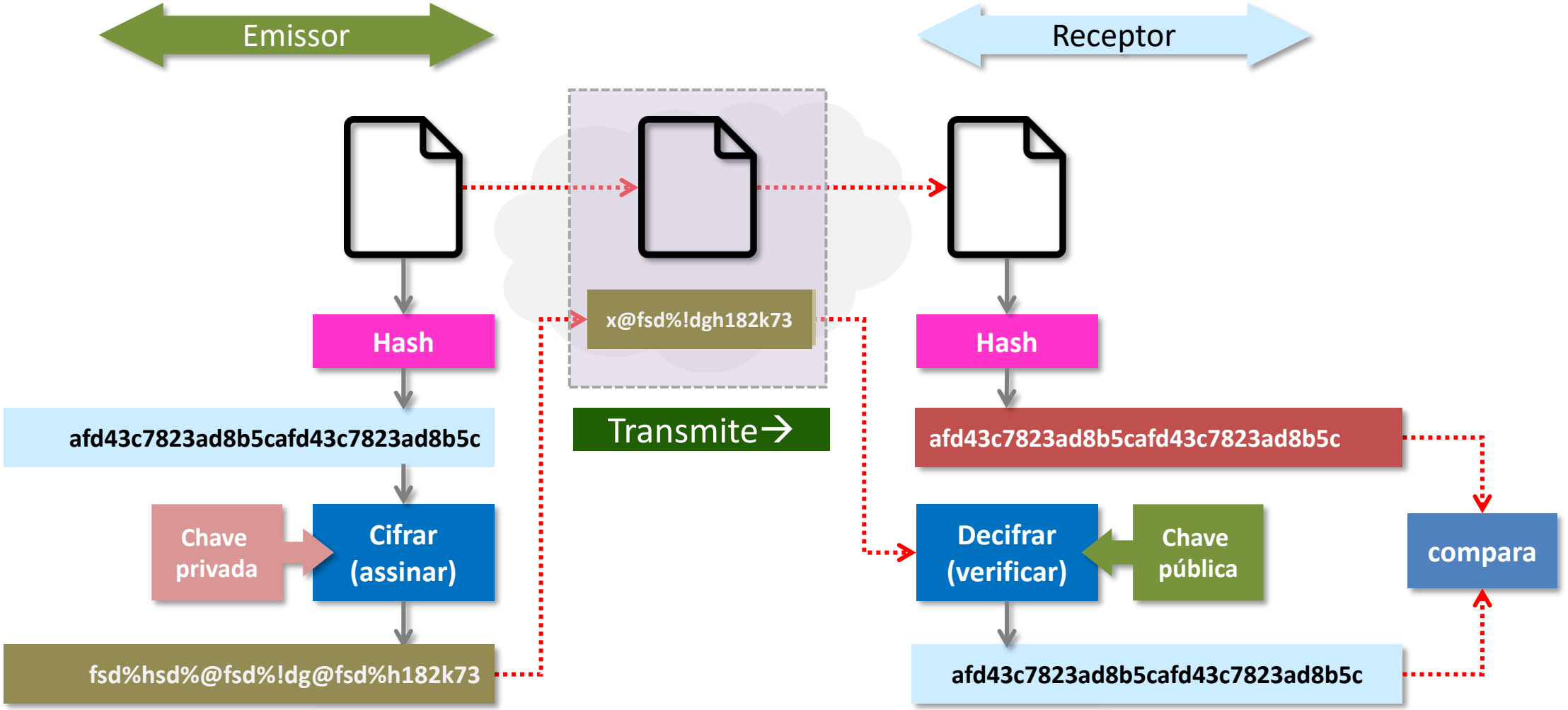
- Descobrir a chave privada a partir da chave pública
  - Fatorar n e obter seus dois fatores primos p e q
  - A partir de p, q e do expoente público e, obtém-se facilmente d
- Achar X na equação $Y=X^e \bmod n$
  - Uma vez que a mensagem original é X e que ambos n e o expoente e são públicos e Y pode ser capturado.
  - Essa tarefa é equivalente à fatoração de números
  - Estudos mostram que em casos especiais onde muitas mensagens relacionadas são cifradas com um mesmo expoente (escolhido muito pequeno), é possível recuperar a mensagem original

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA
PARAÍBA

Assinatura Digital

Prof. Dênio Mariz, IFPB

Assinatura Digital



HMAC – Hash-based Message Authentication Code

- Informação que confirma se a mensagem veio do remetente declarado (autenticidade) e não foi alterada (integridade)
- Requer:
  - Chave simétrica
  - Algoritmo para assinatura
  - Algoritmo de hash
- Diferente de assinatura digital
  - Assinatura digital requer chaves assimétricas e a certeza de que a chave pública é correta
  - MAC usa chave simétrica (requer apenas a garantia de que a chave é conhecida entre os parceiros)

Certificado Digital X.509

Prof. Dênio Mariz, IFPB

Gerenciamento de Chaves Públicas

- Chaves públicas não precisam ser secretas, mas precisam ser autênticas
  - Suponha que Alice e Bob não se conhecem, como Bob vai saber a chave pública de Alice?
  - Se alguem mandar pra Bob a chave pública de Alice, quem garante que ela é mesmo de Alice?
- Solução:
  - Bob deve confiar em alguém que garante que a chave pública é de Alice
  - Bob confia em uma autoridade certificadora, que emite um certificado contendo a chave pública de Alice

Certificado Digital

- Um documento digital assinado
  - Contém a chave pública (pessoa, site web, empresa)
  - Assinado por uma autoridade certificadora (AC)
  - Quem assinou atesta que a chave é autêntica
- Vantagens
  - Atesta autenticação e integridade da chave pública (assinatura)
  - Permite estabelecer confidencialidade
- Desvantagens
  - Requer confiança na AC, AC pode ser comprometida
  - Gerência de certificados
  - Aplicações como sites web devem (quase sempre) comprar um

Veja opção gratuita em Let's Encrypt

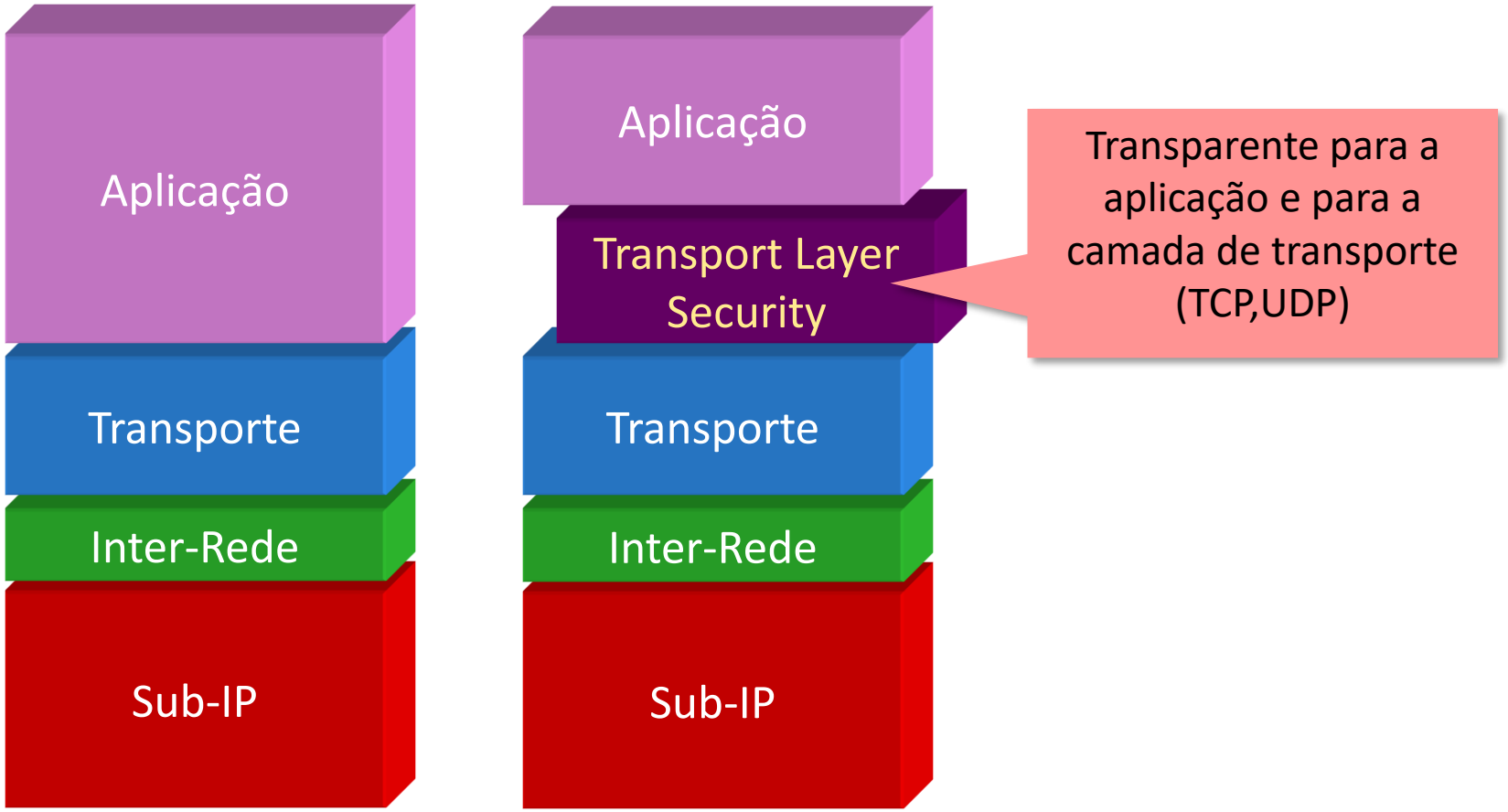
Certificado: padrão X.509

Version	Versão do X.509 (atual=v.3)
Serial Number	Número único por CA (usado na revogação)
Issuer	Nome do CA (padrão X.500)
Validity period	Data de Início e fim da validade do certificado
Subject	Entidade para a qual a chave está sendo certificada
Public key	**A chave pública da entidade + ID do algoritmo**
Issuer ID	ID do emissor do certificado (opcional)
Subject ID	ID do dono do certificado (opcional)
Extensions	Muitas extensões foram definidas
Signature Algorithm	Algoritmo usado para assinar o certificado
Signature	**Assinatura do certificado (com a chave privada do CA)**

Transport Layer Security

Prof. Dênio Mariz, IFPB

Modelo TCP/IP com Segurança



Modelo TCP/IP

Modelo TCP/IP com SSL/TLS

SSL/TLS – Visão Geral

- Protocolos para troca de chaves e envio de mensagens confidenciais
- Permitem autenticar o servidor para o cliente
- Permitem autenticar o cliente para o servidor [opcional]
- Estabelece uma chave de sessão simétrica (secreta) para troca de dados
- Permite verificação de integridade dos dados
- Permite compressão de dados [opcional]

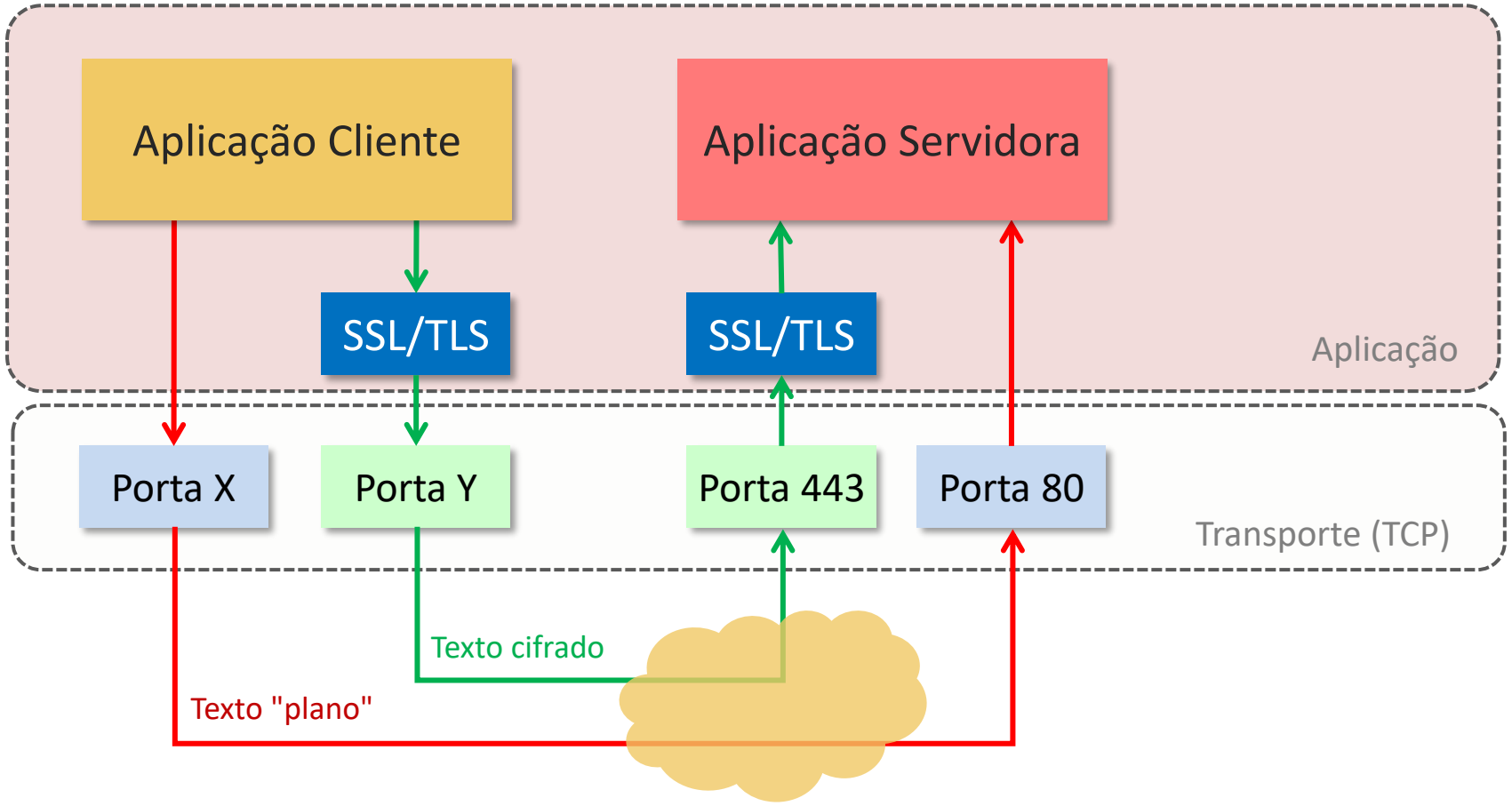
- Criptografia: TLS usa solução Híbrida
  - Chaves pública e privada para cifrar/decifrar uma chave simétrica Chave simétrica para cifrar/decifrar os dados

TLS – Transport Layer Security

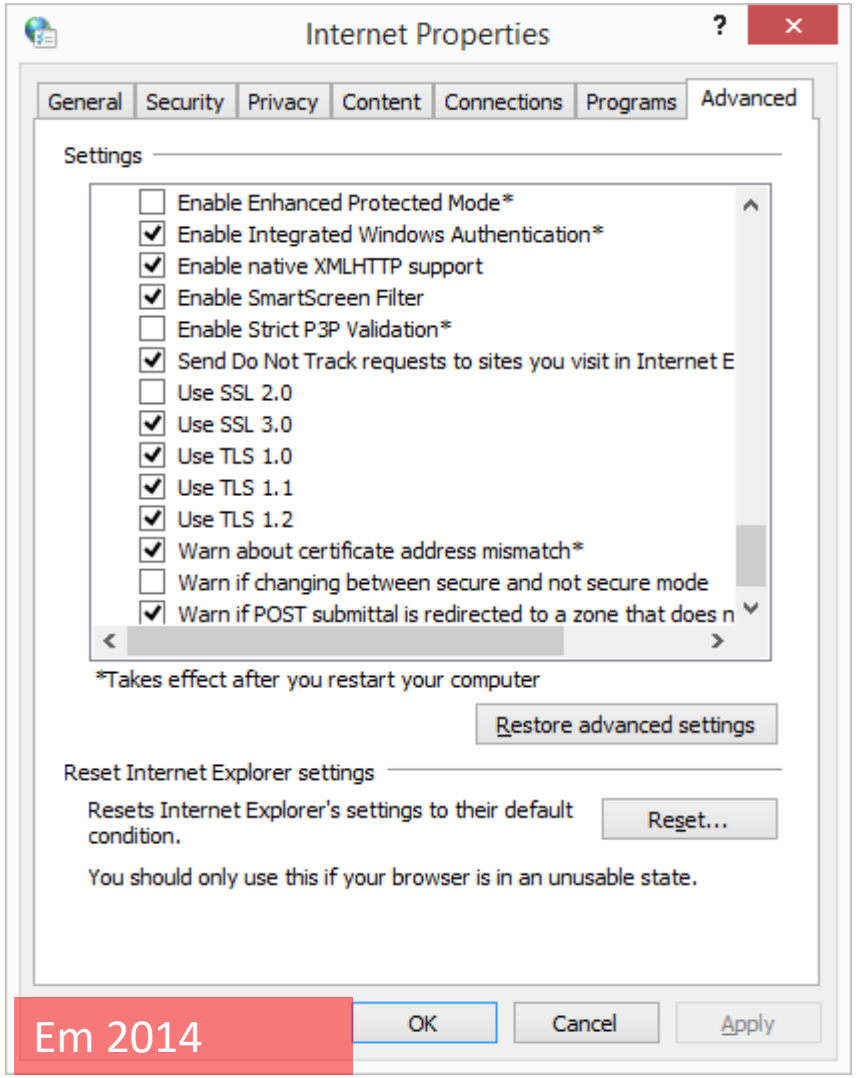
- SSL 3.0 – Publicado pela Netscape em 1996
- TLS 1.0 - RFC 2246 de Janeiro 1999
  - Baseado no SSL 3.0
  - Algumas vulnerabilidades já conhecidas desde 2002
- TLS 1.1 - RFC 4346 de Abril 2006
  - Proteção contra ataques de Cipher block chaining (CBC)
  - Melhorias no tratamento de de erros
  - Suporte para parâmetros registrados no IANA
- TLS 1.2 - RFC 5246 de Agosto 2008
  - Várias melhorias para descrição do hash usado pelo cliente e servidor
  - Adição de hash SHA-256
  - Expansão do suporte para authenticated encryption ciphers, Galois/Counter Mode (GCM)
  - Extensões para uso do AES
- Todas as versões do TLS foram refinadas na RFC 6176 de Março 2011
  - Remoção de compatibilidade para trás (ex: TLS não negocia mais o uso de SSL 2.0 nem 3.0)
- TLS 1.3 – Draft em andamento
  - Removerá suporte para MD5 e SHA-224 e algoritmos obsoletos, trará novos mecanismos

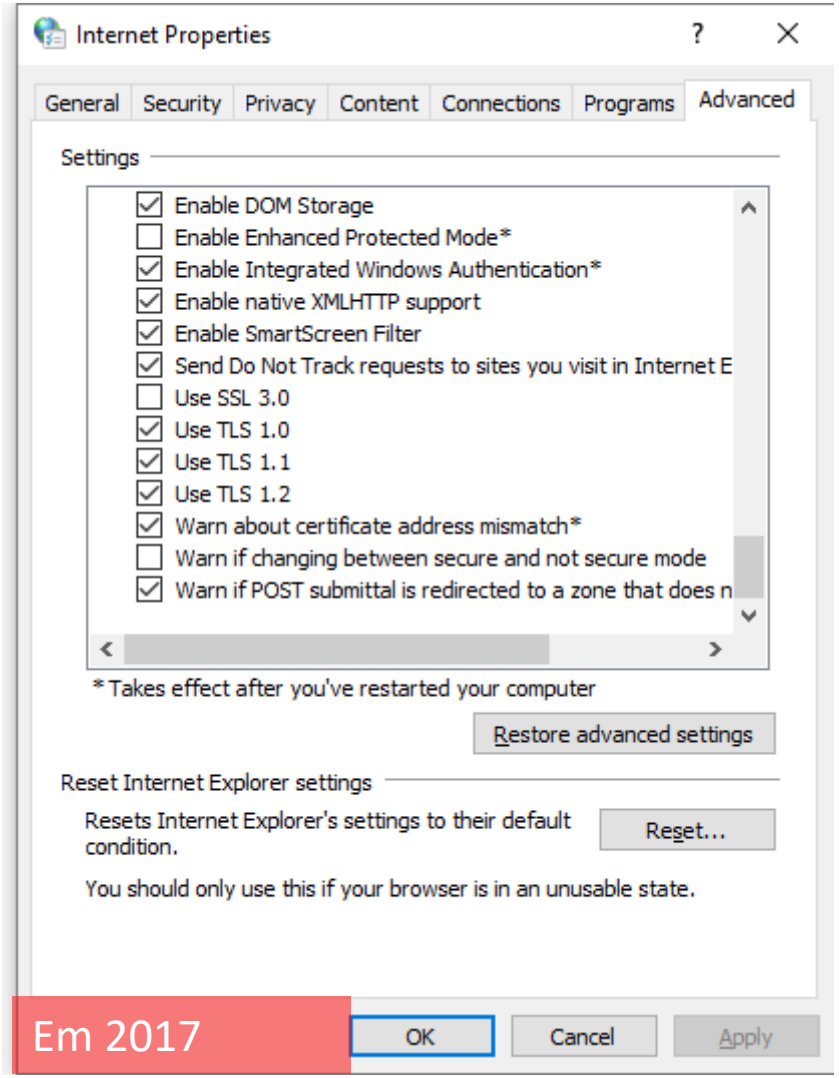
TLS/SSL no Mundo Real

SSL/TLS na aplicação



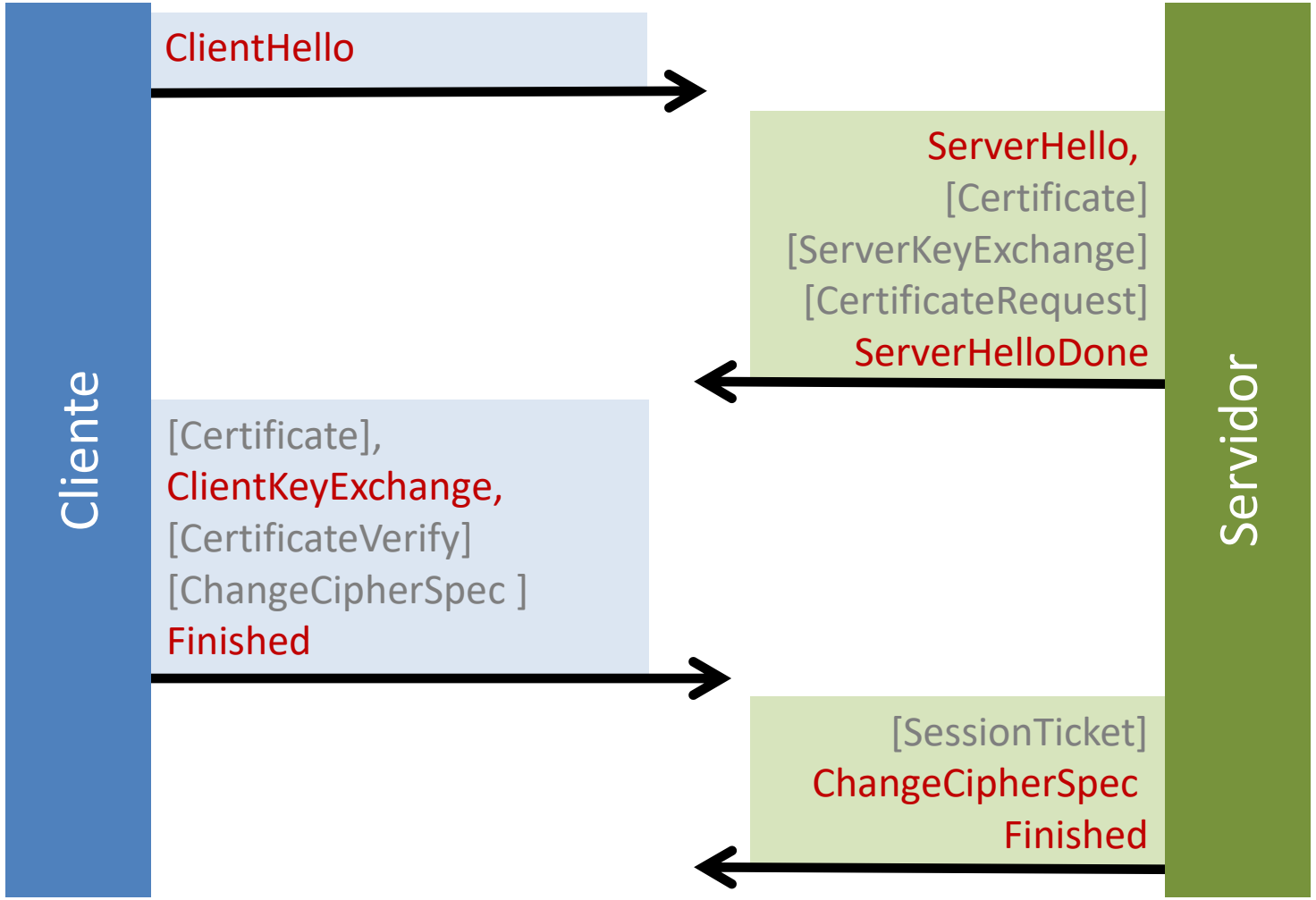
Opções da Internet (windows 8,10)





Control Panel → Internet Options

TLS Handshake Protocol



Mensagens do TLS handshake 1/4

- ClientHello
  - Version: Versão do protocol TLS do cliente
  - Client Random: Número pseudorandomico de 28 bytes para cálculo do “Master secret” (a ser usado na criação da chave se sessão)
  - Session ID: Identificador único da sessão para o cliente
  - Cipher Suite: A suite de cifras suportada pelo cliente, em ordem de preferência
  - Compression Method: Método de compressão selecionado (opcional, normalmente null)

Mensagens do TLS handshake 2/4

- ServerHello
  - Server Version: A maior versão TLS do servidor que seja suportada pelo cliente
  - Server Random: Número pseudorandômico de 28 bytes para cálculo do “Master secret”
  - Session ID: Identificador único da sessão para o servidor
  - Cipher Suite: Apenas uma das suites de cifras selecionada (normalmente a mais forte). Se nenhuma for suportada, um alerta é Gerado e o handshake falha.
  - Compression Method: Método de compressão selecionado (opcional, normalmente null)
- Server Certificate
  - Envia a cadeia de certificados do servidor para o cliente (X.509v3)
  - Não enviada apenas quando a cifra não exige um certificado (ex: DH)

Mensagens do TLS handshake 3/4

- ServerKeyExchange
  - Informações adicionais para estabelecimento da chave (apenas para alguns algoritmos específicos – e.g. Diffie Hellman)
- CertificateRequest
  - Servidor solicita o certificado do cliente (opcional)
  - Inclui nomes das autoridades raiz confiáveis do servidor
- ServerHelloDone
  - Avisa ao cliente do fim do ServerHello
- Certificate (client)
  - Envia cadeia de certificados do cliente para o servidor (se solicitado por CertificateRequest)
- ClientKeyExchange
  - A chave simétrica secreta é estabelecida pelo cliente (fase final do DH ou envio via RSA cifrada com a pública do servidor)

Mensagens do TLS handshake 4/4

- CertificateVerify
  - Envia uma assinatura digital para todas as mensagens trocadas até então para verificação por parte do servidor
  - Serve para provar ao servidor que o cliente realmente é o dono da chave pública
  - Apenas quando o certificado do cliente é enviado para o servidor e tem capacidade de assinatura
- Session Ticket
  - Servidor envia um "Ticket de Sessão" que é a chave da sessão cifrada com uma chave que somente o servidor possui (*Session Ticket Encryption Key - STEK*). Isso permitirá a um cliente "retomar" uma sessão reduzindo a sobrecarga do handshake.
- ChangeCipherSpec (client, Server)
  - Notifica o parceiro que as mensagens subseqüentes serão protegidos com algoritmos e chaves recém-negociados.
  - Mensagens seguintes serão cifradas com o algoritmo simétrico negociado
- Finished (client, server)
  - (Também chamada de Encrypted Handshake Message)
  - É um hash de toda a conversa anterior.
  - É a primeira mensagem cifrada pelos algoritmos negociados. Parceiro deve verificar a corretude.
  - Após enviar a sua e conferir a do parceiro, dados podem ser enviados protegidos pelo mesmo mecanismo.

TLS – Cipher Suite

- Um identificador de 3 bytes que define um conjunto de algoritmos (cipher suite) necessários para proteger uma conexão TLS

- Formato: PROTO_KX_AU_WITH_ENC_MAC
  - PROTO – protocolo (TLS, SSL, SSL2)
  - KX – algoritmo para troca de chaves {RSA, DH, DHE, ECDHE}
  - AU – algoritmo de autenticação (opcional)
  - ENC – algoritmo de criptografia simétrica
  - MAC – Autenticação da mensagem (integridade)

Exemplo de cifra TLS

➔TLS_RSA_WITH_AES_128_CBC_SHA

- TLS = o protocolo TLS
- RSA = RSA para troca de chave
- RSA = RSA para autenticação
- AES_128_CBC = AES 128 no modo "cipher block chaining"
- SHA= algoritmo de hash

→ TLS_RSA_WITH_AES_128_CBC_SHA é obrigatória no TLS 1.2

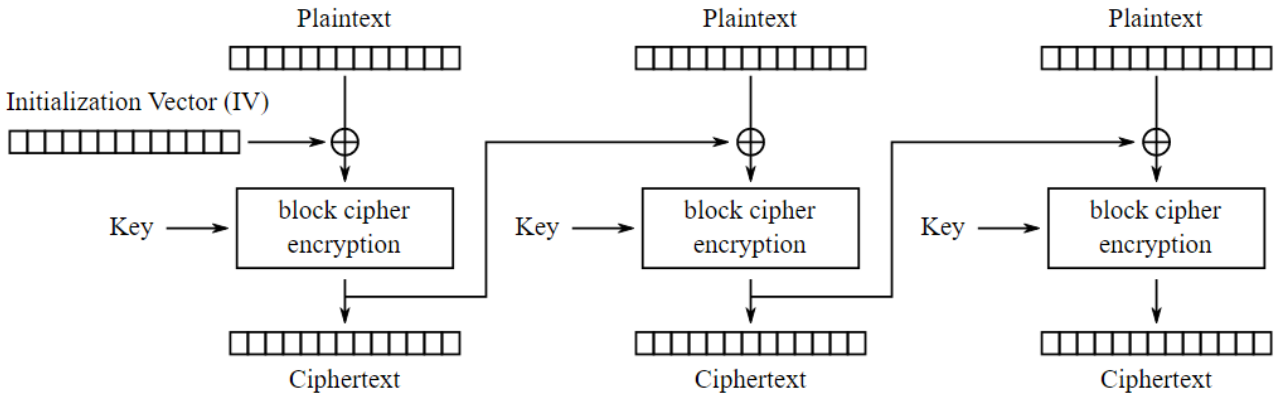
Exemplo de conjuntos de cifras TLS

Cipher ID	Name	Key Exchange	Authentication	Encryption	Bits	MAC
0x000000	TLS_NULL_WITH_NULL_NULL	NULL	NULL	NULL	0	NULL
0x000001	TLS_RSA_WITH_NULL_MD5	RSA	RSA	NULL	0	MD5
...	...	...	...	...	...	...
0x00002E	TLS_RSA_PSK_WITH_NULL_SHA	RSA	PSK	NULL	0	SHA
0x00002F	TLS_RSA_WITH_AES_128_CBC_SHA	RSA	RSA	AES_128_CBC	128	SHA
0x000030	TLS_DH_DSS_WITH_AES_128_CBC_SHA	DH	DSS	AES_128_CBC	128	SHA
0x000031	TLS_DH_RSA_WITH_AES_128_CBC_SHA	DH	RSA	AES_128_CBC	128	SHA
0x000032	TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DHE	DSS	AES_128_CBC	128	SHA
0x000033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE	RSA	AES_128_CBC	128	SHA
0x000034	TLS_DH_Anon_WITH_AES_128_CBC_SHA	DH	Anon	AES_128_CBC	128	SHA
0x000035	TLS_RSA_WITH_AES_256_CBC_SHA	RSA	RSA	AES_256_CBC	256	SHA
0x000036	TLS_DH_DSS_WITH_AES_256_CBC_SHA	DH	DSS	AES_256_CBC	256	SHA
0x000037	TLS_DH_RSA_WITH_AES_256_CBC_SHA	DH	RSA	AES_256_CBC	256	SHA
0x000038	TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DHE	DSS	AES_256_CBC	256	SHA
...	...	...	...	...	...	...

https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4

Cipher Block Chaining

- Antes de cifrar o texto plano, um XOR é feito entre o texto plano e o texto cifrado anterior
  - Cada bloco de texto cifrado depende de todos os blocos de texto plano processados até esse ponto.
  - Um vetor de inicialização deve ser usado no primeiro bloco.



Cipher Block Chaining (CBC) mode encryption

Fonte: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Avaliação da Segurança do SSL/TLS

Cipher			Protocol version						Status
Type	Algorithm	Nominal strength (bits)	SSL 2.0	SSL 3.0 [n 1][n 2][n 3][n 4]	TLS 1.0 [n 1][n 3]	TLS 1.1 [n 1]	TLS 1.2 [n 1]	TLS 1.3 (Draft)	
Block cipher with mode of operation	AES GCM[32][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	AES CCM[33][n 5]		N/A	N/A	N/A	N/A	Secure	Secure	
	AES CBC[n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	Camellia GCM[34][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	
	Camellia CBC[35][n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	ARIA GCM[36][n 5]	256, 128	N/A	N/A	N/A	N/A	Secure	Secure	
	ARIA CBC[36][n 6]		N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	SEED CBC[37][n 6]	128	N/A	N/A	Depends on mitigations	Secure	Secure	N/A	
	3DES EDE CBC[n 6][n 7]	112[n 8]	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	
	GOST 28147-89 CNT[31][n 7]	256	N/A	N/A	Insecure	Insecure	Insecure		Defined in RFC 4357
	IDEA CBC[n 6][n 7][n 9]	128	Insecure	Insecure	Insecure	Insecure	N/A	N/A	Removed from TLS 1.2
	DES CBC[n 6][n 7][n 9]	56	Insecure	Insecure	Insecure	Insecure	N/A	N/A	
		40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	Forbidden in TLS 1.1 and later
	RC2 CBC[n 6][n 7]	40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
Stream cipher	ChaCha20-Poly1305[42][n 5]	256	N/A	N/A	N/A	N/A	Secure	Secure	Defined for TLS 1.2 in RFCs
	RC4[n 11]	128	Insecure	Insecure	Insecure	Insecure	Insecure	N/A	Prohibited in all versions of TLS by RFC 7465
		40[n 10]	Insecure	Insecure	Insecure	N/A	N/A	N/A	
None	Null[n 12]	-	N/A	Insecure	Insecure	Insecure	Insecure	Insecure	Defined for TLS 1.2 in RFCs

Fonte: http://en.wikipedia.org/wiki/Transport_Layer_Security (Mar-2017)

Testando TLS

- Testando TLS
  - www.ssllabs.com/ssltest
- TLS
  - https://technet.microsoft.com/en-us/library/cc785811.aspx
  - https://sites.google.com/site/tlsssloverview/handshake-process
  - https://hpbn.co/transport-layer-security-tls/
  - http://www.pierobon.org/ssl/ch2/detail.htm
- Resumo de ataques TLS
  - https://tools.ietf.org/html/rfc7457

Referências

- MAC
  - https://en.wikipedia.org/wiki/Message_authentication_code
- Cipher Suites
  - http://www.thesprawl.org/research/tls-and-ssl-cipher-suites/
- TLS
  - http://blog.catchpoint.com/2017/05/12/dissecting-tls-using-wireshark/
  - https://tools.ietf.org/html/rfc5246

INSTITUTO FEDERAL DE
EDUCAÇÃO CIÊNCIA E
TECNOLOGIA
PARAÍBA

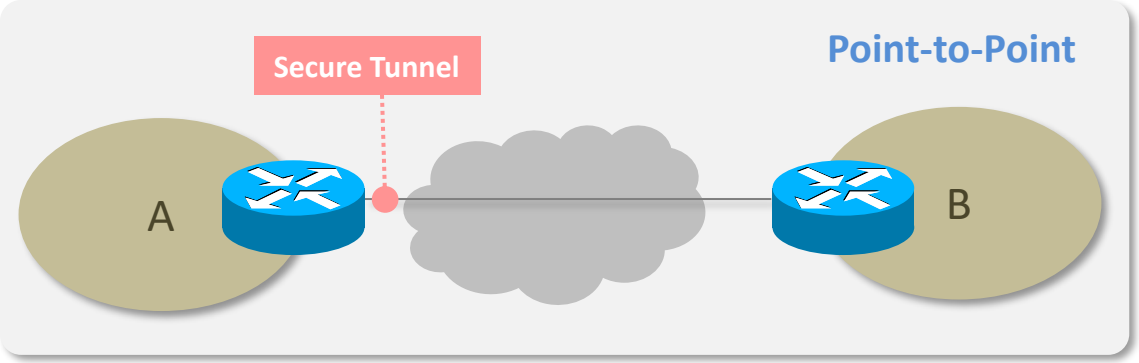
Virtual Private Networks & IPSec

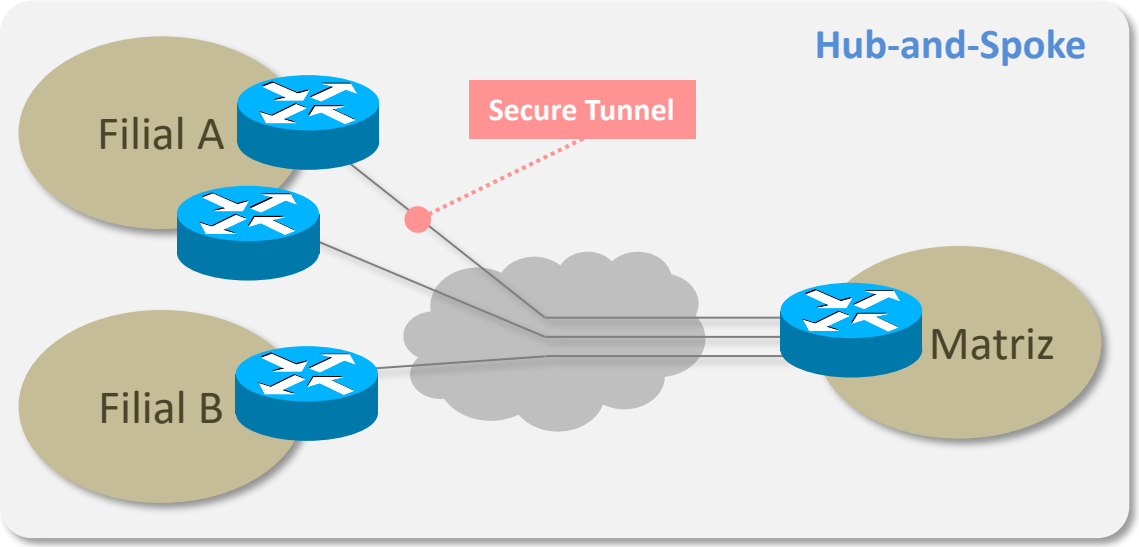
Prof. Dênio Mariz, IFPB

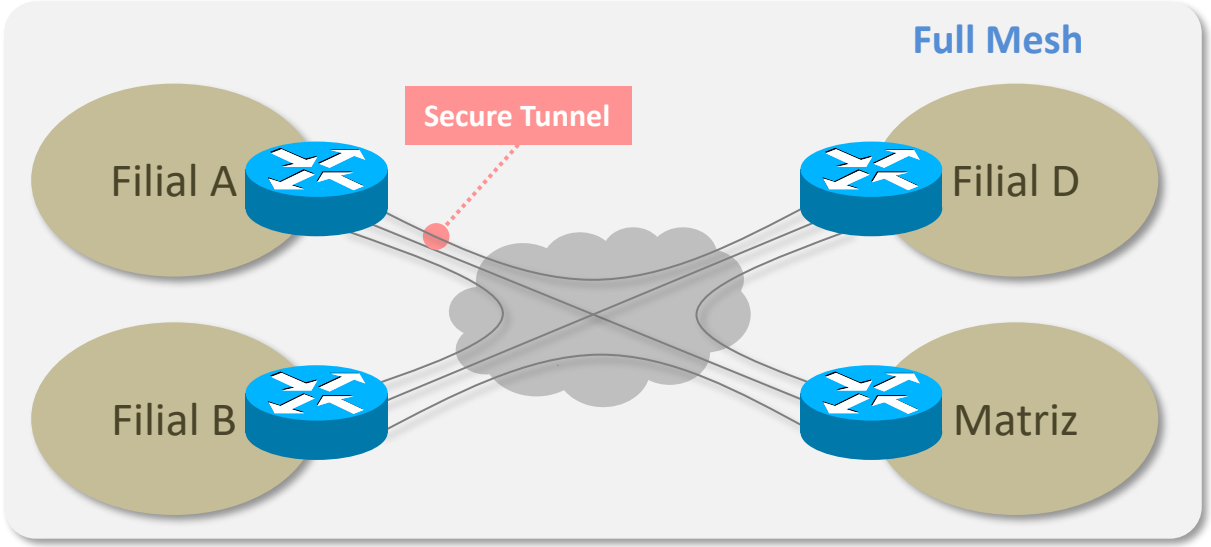
Virtual Private Network

- Definição
  - Uma rede que permite tráfego confidencial de dados sobre uma rede insegura (e.g. Internet) através do uso de protocolos de tunelamento e procedimentos de segurança
- Por que virtual?
  - É uma rede lógica segura
  - Cria a ideia de um Túnel seguro ponto-a-ponto

Topologias VPN

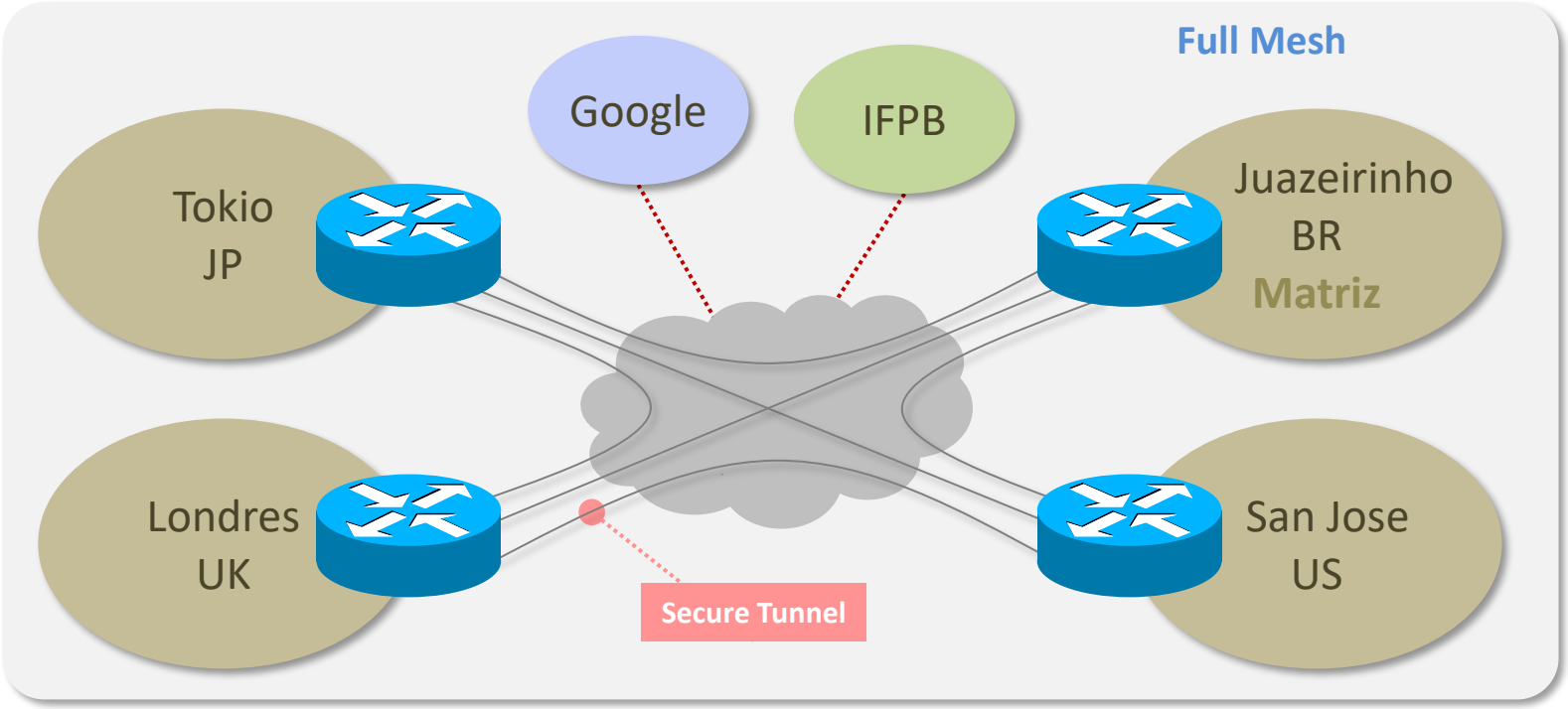






Exemplo de VPN

- Empresa com matriz e filiais
- VPN envia tráfego seguro apenas entre escritórios (túneis)
- Tráfego normal entre quaisquer outras redes
- Túneis são tratados apenas na borda da rede (gateways VPN)
- Transparente para hosts internos na rede local



IPSec – Características de Segurança

- Confidencialidade
  - Emissor cifra pacotes antes de enviar ao receptor (Túnel Seguro Virtual)
- Proteção de Integridade
  - Garante ao receptor identificação de modificações no pacote
- Autenticação nas pontas do túnel
  - Perite verificar a origem de cada pacote
- Replay Protection
  - Previne atacante de salvar pacotes criptografados e resubmetê-los depois sem ser detectado

Estabelecendo um “Caminho Seguro”

- Um caminho seguro (túnel) entre dois pontos eles precisa:
  - Consenso em um conjunto de protocolos
  - Decisão sobre qual algoritmo de criptografia usar
  - Troca de chaves públicas para criptografar mensagens
- Para usar o túnel:
  - Hosts internos enviam pacotes normalmente
  - Roteadores de borda decidem o que fazer com o pacote de acordo com uma política pré-configurada
  - Nenhum outro roteador da Internet é alterado
  - Nenhum roteamento especial é necessário

IPSec – Conceitos Básicos

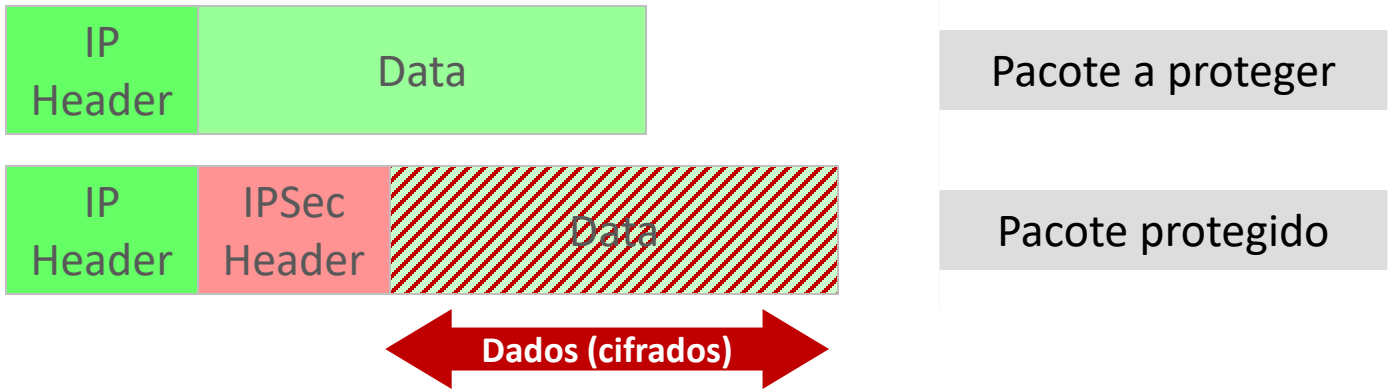
- SA - Secure Association
  - Identifica uma conexão lógica entre dois pontos IPSec (um para cada sentido do trânsito)
  - Armazena informações necessárias para proteger dados na VPN (algoritmos, chaves, protocolos)
  - Cada pacote IPSec carrega um SPI - Secure Parameter Index, que indica qual SA usar
- SAD – Secure Associations Database
  - Armazena as várias SAs em uso pelo roteador na borda (VPN Gateway)
- SPD - Secure Policy Database
  - Um conjunto de regras e políticas
  - Usado para tomar decisões quanto ao que fazer com pacotes IP específicos (descartar, encaminhar plano, criptografar)

IPSec – Protocolos, Cabeçalhos

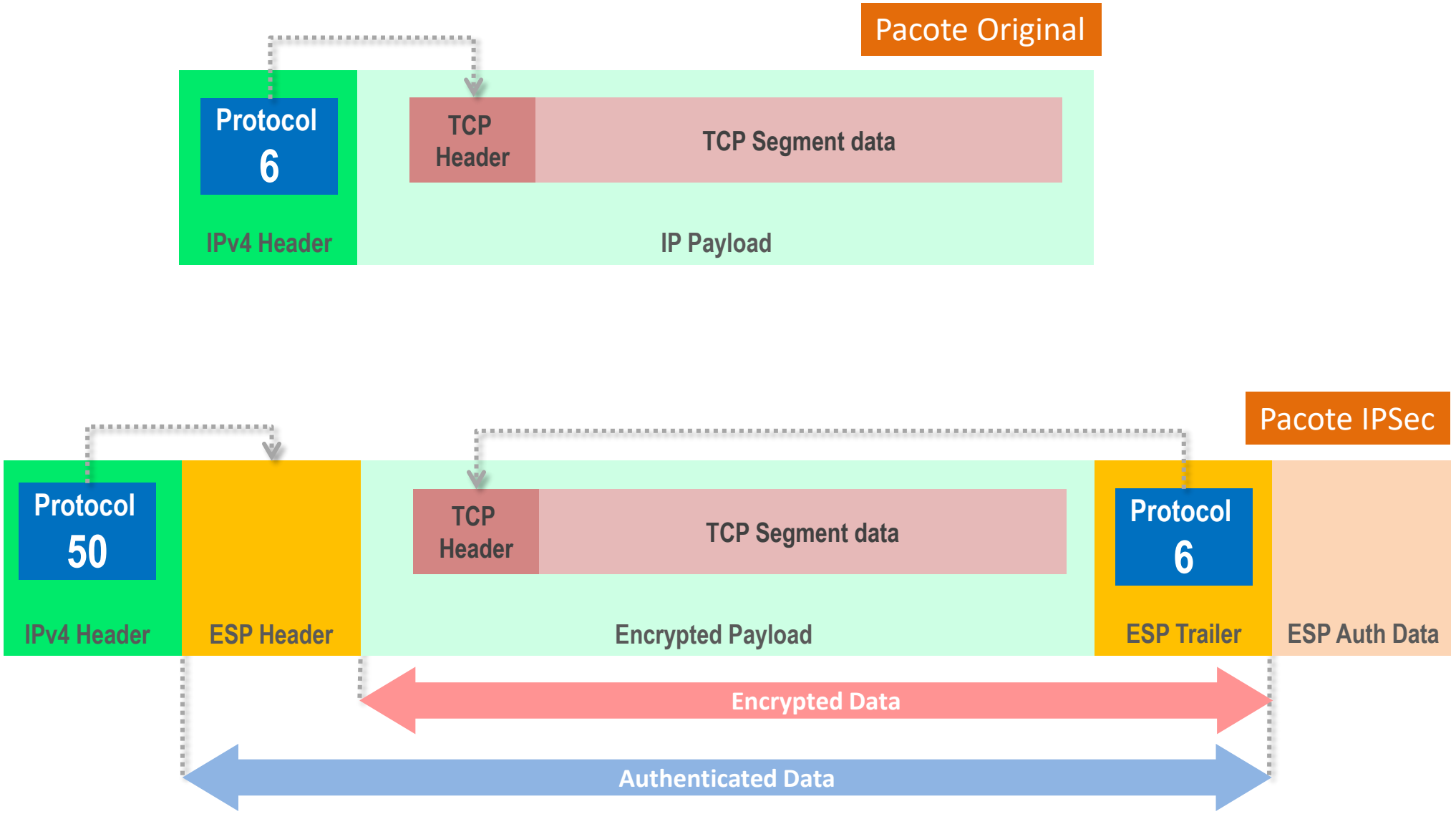
- IKE – Internet Key Exchange (RFC 2401)
  - Protocolo para gerência de chaves, criação e manutenção de Secure Associations
- ISAKMP – Internet Security Association and Key Management Protocol (RFC 2408)
  - Responsável pela criação, modificação e exclusão das Secure Associations
- OAKLEY Key Determination Protocol (RFC 2412)
  - Mecanismo para estabelecer chaves entre dois participantes
  - Utilizado pelo ISAKMP para estabelecer chaves entre Secure Associations
  - É uma variação mais segura do algoritmo Diffie-Hellman (RFC 2631)
- AH – Autentication Header (RFC 4302)
  - Informações para implementação dos serviços de integridade, autenticação e anti-*replay*.
  - Código de protocolo = 51 na pilha TCP/IP
- ESP – Encapsulating Security Payload (RFC 4303)
  - Informações para implementação do serviço de confidencialidade
  - Código de protocolo = 50 na pilha TCP/IP

IPSec – Modo Transporte

- Mantém o cabeçalho IP do pacote original
- Acrescenta um cabeçalho IPSec
- Cabeçalho IPSec pode conter assinatura dos dados
- Permite: verificação da integridade, autenticação da origem, não-repúdio, confidencialidade
- Não esconde informações do cabeçalho original



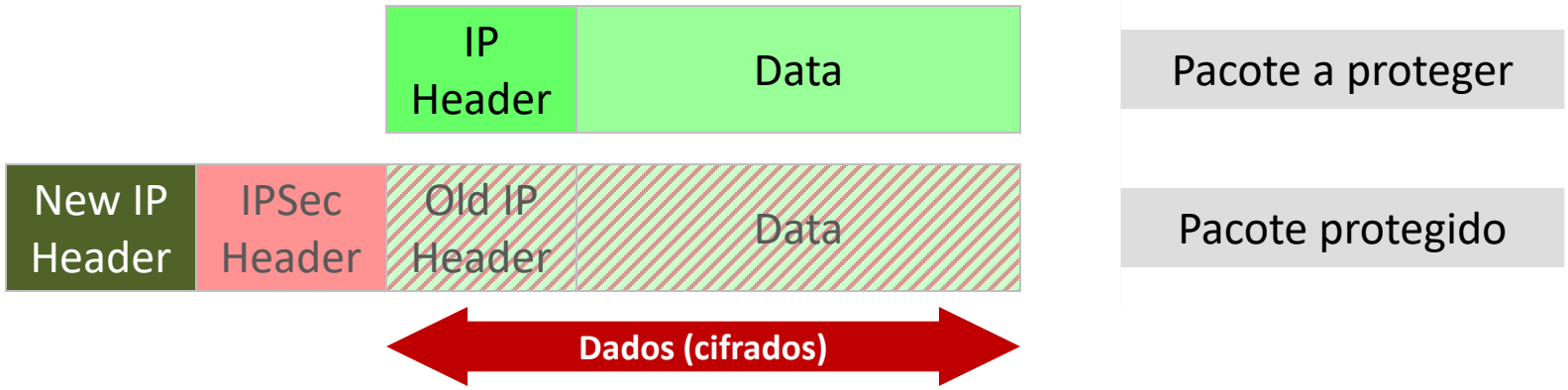
IPSec – Modo Transporte (detalhe)



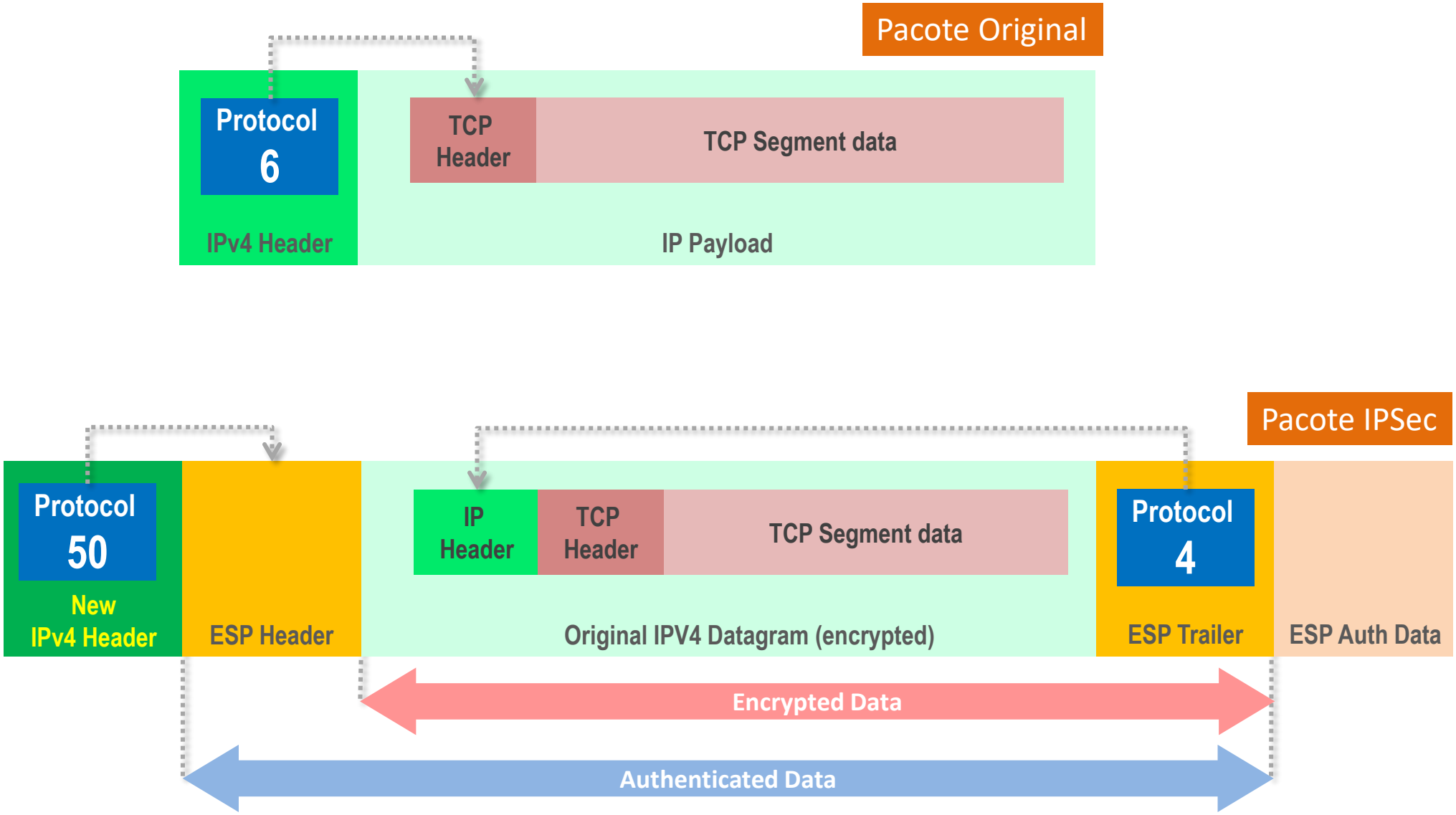
www.tcpipguide.com/free/t_IPSecEncapsulatingSecurityPayloadESP-2.htm

IPSec – Modo Túnel

- Pacote original é tratado como “dados a enviar”
- Acrescenta um novo cabeçalho IP + cabeçalho IPSec
- Permite: verificação da integridade, autenticação da origem, não-repúdio, confidencialidade
- Esconde completamente as informações do cabeçalho original



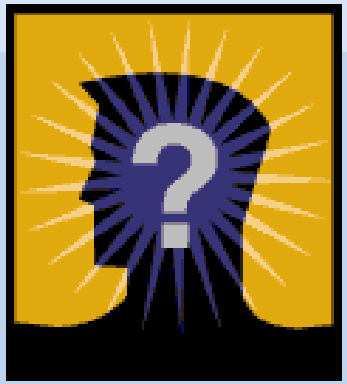
IPSec – Modo Túnel (detalhe)



VPN - Resumo

- Mecanismo para implementar segurança sobre redes inseguras
- Aumenta a complexidade do suporte (configuração, manutenção)
- Precisa de ajustes no firewall, NAT
  - Protocolo 50 - Encapsulating Security Protocol (ESP)
  - Protocolo 51 - Authentication Header (AH)
  - Porta UDP 500 (IKE)
- Aumento no atraso dos pacotes
  - 7 milisegundos adicionais para um túnel com DES em um link de 10Mbps (testes em um Cisco 2600)
- Segurança
  - Usa a Internet que voce já tem
  - Existem boas soluções gratuitas ou de baixo custo
  - Solução para redes de broadcast inseguras (ex: 802.11, Satélite)

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA
PARAÍBA



Segurança de Redes

Tópicos Selecionados

Dênio Mariz
denio@ifpb.edu.br