

# **Criptografia e Segurança na Informática**



**Pedro A. D. Rezende**

**Ciência da Computação**

**Universidade de Brasília**

“Não existe nada de demoníaco na técnica,  
mas existe o mistério da sua essência.  
É a essência da Técnica,  
enquanto destino de revelação,  
que é o perigo”

**Martin Heidegger, 1995**  
**citado em “Cibercultura”, de André Lemos**  
**Editora Meridional, Porto Alegre, 2002**

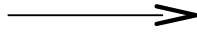
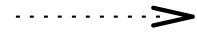
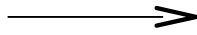

“Quem tem ouvidos, ouça o que o Espírito diz às igrejas.  
Ao vencedor darei um prêmio: o maná escondido.  
Darei também uma pedrinha branca a cada um.  
Nela está escrito um nome novo,  
que ninguém conhece.  
Só quem recebeu. ”

**Apocalipse de São João, Capítulo 2 Versículo 17**

# Índice

<b>0: Introdução</b> .....	ii
<b>1: Fundamentos</b> .....	1
<b>2: Elementos de Protocolos</b> .....	21
<b>3: Protocolos Importantes</b> .....	34
<b>4: Técnicas Criptográficas</b> .....	59
<b>5: Algoritmos Criptográficos Importantes</b> .....	79
<b>6: Enfoques para Implementação</b> .....	123
<b>7: Exemplos de Aplicações</b> .....	155
<b>Apêndices</b> .....	A, B, C

## Notação -

Conceito:	<b>Negrito</b>
Explicação; Termo a definir; Anglicismo:	<i>Itálico</i>
Descrição de algoritmo:	Font Arial
Objeto matemático; Código fonte em C:	Font Courier
Canal de comunicação digital inseguro:	
Canal presumido seguro (sigilo e/ou integridade):	
Passo presumido executado antes do protocolo:	<b>0: ...</b> 
Ponto de confiança para algum agente do protocolo:	
Conjunto das sequências de símbolos do alfabeto $\Sigma$ :	$(\Sigma)^*$
Para todo...; Existe...:	$\forall$ ; $\exists$

# 0: Introdução

No mundo hoje, somos levados a crer que podemos comprar soluções prontas – de preferência feitas sob medida – para atender nossas necessidades. Mas a essência do que se busca nem sempre pode ser encontrada na prateleira, embrulhada em pacote. Essas notas iniciam-se com um comentário sobre a natureza da busca de segurança para a informática. Creio que a essência dessa busca não pode ser encapsulada e apresentada em forma de algoritmo. Por isso, não posso me propor apresentar-lhes roteiros ou receitas para segurança na informática. Proponho, ao invés disso, ajudá-los a desenvolver sensibilidade sobre como usar o conhecimento e as ferramentas até hoje acumulados, nessa busca. Para guiar-nos em minha proposta, compilei nas sessões seguintes um resumo dos conceitos, técnicas e procedimentos mais eficazes hoje conhecidos para a construção de mecanismos criptográficos de proteção a dados digitais, e de meios para esses mecanismos se integrarem a sistemas de informação que se queira proteger. Começamos observando que a segurança de que falamos não é dos bits. Os bits não podem sofrer dano, pois são apenas símbolos. Bits apenas ocorrem em sequências que representam dados. Quem pode sofrer dano são aqueles para quem tais sequências de bits representam valor.

Na sessão 1 veremos os principais conceitos técnicos ou científicos, inclusive de natureza lingüística e psicológica, relacionados com a construção e funcionamento de ferramentas para a proteção a dados – os serviços básicos de segurança computacional. Na sessão 2 abordaremos como mecanismos de proteção contra ameaças – os protocolos – são concebidos, para tornarem viável o uso destas ferramentas. Na sessão 3 estudaremos o funcionamento dos mecanismos que tem se mostrado eficazes, ou seja, os principais protocolos criptográficos de padrão aberto hoje em uso. Em seguida exploraremos na sessão 4 as formas como os serviços básicos são escolhidos e montados em protocolos específicos, para então examinarmos em mais detalhes, na sessão 5, as peças principais desses protocolos – os algoritmos criptográficos – inclusive quanto às formas mais elementares de se avaliar a segurança que oferecerem. Detalhes de segurança externos aos protocolos que influem nas suas escolhas são abordados na sessão 6, e uma visão do estado atual da aplicação da criptografia, principalmente na internet, será oferecida na sessão 7. Assim, faremos uma abordagem top-down até os algoritmos criptográficos e técnicas para sua implementação, e daí retornamos, por via reversa, de novo a uma visão de conjunto do assunto.

Enquanto discorro sobre conceitos, protocolos e algoritmos criptográficos, buscarei situá-los, por meio de comentários, no cenário real, onde deve configurar-se claro o caráter global e integrador do significado da segurança na informática, de cuja compreensão evolui tal sensibilidade. Estas notas representam portanto *apenas* um esforço para resumir descrições de conceitos, protocolos e algoritmos, não havendo a pretensão de fazê-las passar por texto didático autônomo. Referências bibliográficas estão espalhadas ao longo das notas e devem ser buscadas por quem julgar necessária uma apresentação textual mais discursiva do assunto. Espera-se do leitor algum conhecimento da aritmética modular (aritmética dos processadores digitais), do funcionamento básico de sistemas operacionais e das redes de computadores, e noções de complexidade de algoritmos. E por que esse conhecimento é esperado?

A criptografia é uma área de especialização da matemática e da engenharia que oferece técnicas de proteção a mecanismos de acesso e à integridade de dados, e ferramentas de avaliação da eficácia dessas técnicas. Estas técnicas e ferramentas são de natureza puramente sintática, não podendo, portanto, serem destinadas a fornecer ou induzir, por si mesmas, confiança no significado da informação que tais dados supostamente veiculam. A criptografia pode oferecer segurança na informática somente onde e quando a confiança no significado da informação veiculada pelos dados protegidos já tenha sido obtida ou fornecida por outros meios. Portanto, a criptografia não oferece nenhuma solução mágica para problemas de segurança na informática. O que oferece são truques para manipulação de probabilidades que nos permitem escolher o terreno e o maneira como poderemos nos defender no mundo dos bits.

Para explicar minha crença sobre a natureza do assunto que iremos tratar, escolhi um texto de um experiente criptólogo, Bruce Schneier, divulgadas em uma lista de discussão eletrônica em novembro de 96 (schneier@couterpane.com), cuja tradução transcrevo abaixo

## Porque a criptografia é mais difícil do que parece

“Do correio eletrônico à telefonia celular, do acesso seguro a servidores WEB à moeda eletrônica, a criptografia é parte essencial dos sistemas de informação de hoje. A criptografia ajuda a imputar responsabilidade, promover a justiça, prover acurácia e privacidade. Pode prevenir fraudes em comércio eletrônico e garantir a validade de transações financeiras. Usada apropriadamente, protege a anonimidade e fornece provas de identidade de pessoas. Pode impedir vândalos de alterarem sua página na internet e competidores industriais de lerem seus documentos confidenciais. Com o comércio seguindo sua marcha pelas redes de computadores, a criptografia se tornará cada vez mais vital.

Mas a criptografia hoje existente no mercado não fornece a segurança que apregoa seu *marketing*. A maioria desses sistemas são projetados e implementados não por criptógrafos, mas por engenheiros que pensam que a criptografia é como qualquer outra tecnologia de computadores. Não é. Você não pode tornar um sistema seguro simplesmente acrescentando criptografia como uma medida adicional. Você precisa saber o que está fazendo a cada passo do caminho, da concepção até a implementação do sistema.

Bilhões de dólares são gastos em segurança de computadores, e quase todo este dinheiro é desperdiçado em produtos inseguros. Afinal, criptografia fraca parece idêntica à criptografia forte na vitrine de software. Dois produtos de encriptação de correio eletrônico no mercado têm interface de usuário praticamente idênticas, enquanto um deles é seguro e o outro permite bisbilhotagem. Uma tabela contendo comparações entre recursos pode sugerir que dois produtos tenham funcionalidade similar, embora um possa ter furos comprometedores de segurança e o outro não. Um criptógrafo experiente pode reconhecer a diferença. Determinados tipos de criminosos também poderão.

A segurança de computadores hoje em dia é um castelo de cartas; pode se manter de pé por agora, mas não vai durar. Muitos produtos inseguros ainda não foram quebrados porque ainda estão em sua infância, mas à medida em que se tornem mais e mais usados, tornar-se-ão alvos atraentes para criminosos. A imprensa divulgará os ataques, minando a confiança do público nesses sistemas. No final, produtos sobreviverão no mercado de acordo com a robustez de sua segurança.

Os ataques a sistemas de informação são dos mais variados tipos. Toda forma de comércio já inventado tem sido alvo de fraudes, desde as balanças propositadamente descalibradas, o dinheiro falso, as faturas frias, etc. O comércio eletrônico também sofrerá fraudes, personificação, bloqueio de serviço, e falsificações. Não se pode caminhar pelas ruas usando uma máscara que imita o rosto de outra pessoa sem ser percebido, mas no mundo digital é muito fácil personificar outrem. Ocorre que a informatização torna os riscos maiores ainda, permitindo ataques automatizados, impossíveis de serem conduzidos contra sistemas não automatizados. Um ladrão pode se sustentar retirando um centavo por mês de cada dono de cartão de crédito Visa. Apenas com a criptografia forte pode-se proteger tais sistemas contra estes tipos de ataques.

Violações contra a privacidade constituem outro tipo de ataque. Alguns ataques contra a privacidade são direcionados: alguém da imprensa pode tentar ler a correspondência eletrônica de uma figura pública, ou uma companhia pode tentar interceptar as comunicações de um competidor. Pode-se também tentar ataques de colheita, buscando informações interessantes num mar de dados: viúvas ricas, usuários de AZT, pessoas que visitam determinada página na internet, etc.

O vandalismo eletrônico é um problema cada vez mais sério. Já foram pichadas as páginas digitais da agência de serviço secreto dos EUA, enviadas cartas-bomba digitais a provedores da internet, e cancelados centenas de listas de discussão eletrônicas, além de ataques que bloqueiam o acesso a computadores que se comunicam por meio de determinados protocolos. E como divulgado, ladrões e vândalos rotineiramente invadem redes de computadores. Quando as salvaguardas de segurança não são adequadas, os invasores correm poucos riscos de serem flagrados. Os atacantes não seguem regras, podendo atacar sistemas usando técnicas não antecipadas pelos projetistas e analistas de sistemas, como no exemplo de arrombadores que entram numa casa abrindo um buraco na parede, evitando os alarmes e trancas das portas e janelas.

Vândalos cibernéticos também abrem buracos em paredes de bits. Roubam dados técnicos, subornam agentes, modificam programas e mancomunam. Tiram vantagens de tecnologias mais avançadas que a dos sistemas que querem atacar, e até descobrem novos métodos matemáticos para atacá-los. Geralmente dispõem de mais tempo do que alguém honesto normalmente teria para desmontar e examinar um sistema. O SecurID foi usado durante anos até que alguém olhou mais atentamente dentro de seu gerenciador de chaves: seus códigos binários ainda continham rótulos!. As chances favorecem os atacantes, que só precisa encontrar um ponto vulnerável no sistema, enquanto os defensores precisam proteger seu sistema de toda vulnerabilidade possível.

## O que a criptografia pode e não pode fazer

A garantia de 100% de segurança é uma falácia, mas podemos trabalhar em direção a 100% de aceitação de riscos. Fraudes existem nas formas usuais de comércio: dinheiro pode ser falsificado, cheques adulterados ou roubados, números de cartão de crédito copiados. Mesmo assim esses sistemas ainda têm sucesso porque seus benefícios e conveniências compensam as perdas. Cofres, fechaduras e cortinas – mecanismos de privacidade – não são perfeitos mas com frequência são bons o suficiente. Um bom sistema criptográfico atinge o equilíbrio entre o que é possível e o que é aceitável.

A criptografia forte pode resistir com sucesso a ataques que lhe são direcionados até um certo ponto – o ponto onde se torna mais fácil obter, de alguma outra maneira, a informação que ele protege. Um sistema criptográfico, não importa quão seguro, não irá impedir que alguém vasculhe seu lixo. Mas pode perfeitamente prevenir ataques de colheita de dados: ninguém conseguirá vasculhar suficientes latas de lixo para montar a lista de todos os usuários de AZT do país.

A boa notícia sobre criptografia é que já temos os algoritmos e protocolos para proteger nossos sistemas. A má notícia é que esta foi a parte mais fácil: implementações bem sucedidas requerem especialização considerável. As áreas de segurança na informática que interagem com pessoas – gerência de chaves, segurança da interface homem/máquina e controle de acesso – freqüentemente desafiam análise. As disciplinas de infra-estrutura de chaves públicas, segurança do software, segurança de computadores, segurança de redes e projeto de hardware inviolável são também pouco compreendidas.

Companhias muitas vezes fazem mal a parte fácil e implementam algoritmos e protocolos inseguros. Mas mesmo assim, na prática raramente a criptografia é quebrada por causa, ou através, de sua matemática; outras peças do sistema são mais fáceis de serem quebradas. O protocolo mais seguro já inventado poderá facilmente sucumbir a um ataque simples se não for dada atenção a detalhes mais complexos e sutis sobre sua implementação. A segurança do *browser* Netscape 1.0 caiu devido a uma falha no seu gerador de números randômicos. As falhas podem estar em qualquer lugar: no modelo de ameaças, no projeto do sistema, na implementação do software ou do hardware, ou na gerência do sistema. Segurança é uma cadeia, onde um único elo fraco pode quebrar todo o sistema. *Bugs* fatais à segurança podem estar em partes do software distantes dos módulos que implementam serviços de segurança, e uma decisão de projeto que não tenha nada a ver com segurança poderá criar uma falha de segurança.

Uma vez encontrada uma falha de segurança, pode-se consertá-la. Mas encontrar as falhas, para início de conversa, pode ser extremamente difícil. Segurança é diferente de qualquer outro requisito de projeto, porque nele funcionalidade não é igual a qualidade: se um editor de texto imprime corretamente, sabe-se que a função de impressão funciona. Segurança é diferente: só porque um cofre reconhece a combinação correta para abri-lo, não significa que seu conteúdo está seguro contra um chaveiro ou arrombador. Nenhuma quantidade de testes beta revelará todas as falhas de segurança de um sistema, e não haverá nenhum teste possível que prove a ausência destas falhas.

## Modelos de ameaças

Um bom projeto começa por um modelo de ameaças. O que o sistema está sendo concebido para proteger, de quem e durante quanto tempo? O modelo de ameaças deve levar em consideração todo o sistema, não apenas os dados que está sendo projetado para proteger, mas também e principalmente as pessoas que irão usá-lo e como irão usá-lo. O que motivará os atacantes? Que tipo de abusos podem ser tolerados? Deve um tipo de ataque ser prevenido ou basta que seja detectado? Se o pior acontecer e alguma hipótese fundamental sobre a segurança do sistema for violada, que tipo de salvamento pós-desastre pode ser conduzido? Respostas a estas questões não podem ser padronizadas, como os algoritmos e protocolos. São diferentes para cada sistema, e com frequência, projetistas não dedicam tempo a construir um modelo realista das ameaças ou a analisar os riscos.

Modelos de ameaças permitem a desenvolvedores de produtos e consumidores determinar quais medidas de segurança são necessárias: terá sentido encriptar todo seu disco rígido se você não guarda seus documentos de papel num cofre? Como pode alguém de dentro da companhia fraudar o sistema de comércio? Qual é exatamente o custo para se neutralizar a inviolabilidade de um cartão inteligente? Não se pode especificar um sistema seguro sem conhecimento sobre contra o que, e de quem, se deseja protegê-lo.

## Projeto de sistemas

O projeto de um sistema criptográfico seguro deve ser feito somente após o modelo de ameaças ter sido compreendido. Este trabalho é o tema central da criptologia, e é muito especializado. A criptografia mescla várias áreas da matemática: teoria dos números, teoria da complexidade, teoria da informação, teoria da probabilidade, álgebra abstrata, análise formal, dentre outros. Poucos podem contribuir apropriadamente para esta ciência, onde um pouco de conhecimento é muito perigoso: criptógrafos inexperientes quase sempre projetam sistemas falhos. Bons criptógrafos sabem que nada substitui a revisão extensiva feita por colegas e anos de análise. Sistemas de qualidade usam algoritmos e protocolos publicados e bem compreendidos: usar elementos não provados em um projeto é no mínimo arriscado.

O projeto de sistemas criptográficos é também uma arte. O projetista precisa atingir um equilíbrio entre segurança e acessibilidade, anonimidade e responsabilização, privacidade e disponibilidade. A ciência sozinha não garante segurança: somente a experiência e a intuição nascida da experiência podem guiar o criptógrafo no projeto de sistemas criptográficos e na busca de falhas em sistemas existentes.

Bons sistemas de segurança são feitos de pequenos módulos independentemente verificáveis (e que tenham sido verificados!), cada um provendo algum serviço que claramente se resume a uma primitiva. Existem vários sistemas no mercado que são muito grandes para serem verificados em tempo razoável.

## Implementação

Existe uma distância enorme entre um algoritmo matemático e sua implementação concreta em hardware ou em software. Projetos de sistemas criptográficos são muito frágeis. Só porque um protocolo é logicamente seguro, não significa que permanecerá seguro quando o implementador começar a definir estrutura de dados e a descrever a passagem de bits de um lado para outro. “Fechado” nunca será totalmente fechado: esses sistemas têm que ser perfeitamente implementados, senão irão falhar. Uma interface mal projetada pode tornar um encriptador de arquivos de disco completamente inseguro. Uma interface de sincronização mal projetada pode deixar um furo num sistema para comunicações seguras. Confiança excessiva na inviolabilidade de hardware, tais como os chips de cartões selados, pode tornar inútil um sistema de comércio eletrônico. Como estes problemas não aparecem em testes, por vezes aparecem em produtos já lançados no mercado.

Implementadores estão sempre sob pressão de orçamentos e prazos. Cometem os mesmos erros vezes a fio, em muitos produtos diferentes. Usam geradores de seqüências randômicas ruins, não checam condições de erro apropriadamente, e deixam informações secretas em arquivos de *swap*. Muitas destas falhas não podem ser estudadas em livros acadêmicos porque não são tecnicamente interessantes. A única maneira de aprender sobre estas falhas é fazendo e quebrando sistemas de segurança, um após o outro, numa corrida sem fim.

## Procedimentos e Gerência

No final da estória, muitos sistemas de segurança são quebrados por pessoas que os usam, e a maioria das fraudes contra sistemas de comércio são praticadas por quem os opera. Usuários honestos também causam problemas, porque geralmente não ligam para segurança. Eles querem simplicidade, conveniência, e compatibilidade com sistemas legados (inseguros) e em uso. Eles escolhem senhas fracas, anotam-nas, passam-nas para parentes e amigos, largam computadores com sessões abertas, etc. É muito difícil vender fechaduras para pessoas que não querem ser molestadas pela responsabilidade de carregar chaves. Sistemas bem projetados têm que levar em conta as pessoas, e as pessoas são os elementos mais difíceis de serem abstraídos no projeto.

Aí é onde está realmente o custo com segurança. Não está nos algoritmos. A criptografia forte não é mais cara que a fraca. O grosso do custo também não está em projeto e implementação: sai bem mais barato projetar e implementar um bom sistema do que cobrir as perdas com um sistema inseguro. A maior parte de seu custo está em fazer com que as pessoas o utilizem. É difícil convencer o consumidor sobre a importância de sua privacidade financeira, quando o mesmo está disposto a trocar um detalhado registro de suas compras por um milésimo de uma viagem ao Havai. É difícil construir um sistema de autenticação robusto sobre um outro sistema que permite ser penetrado por meio do conhecimento do nome de solteira da mãe de alguém. A segurança é rotineiramente ultrapassada por vendedores, gerentes, executivos e qualquer um que esteja querendo “apenas tocar o serviço”.

Mesmo quando o usuário compreende a necessidade de um sistema de segurança robusto, não terá meios de comparar dois sistemas. Revistas de computação comparam produtos de segurança listando seus recursos e funcionalidade, e não avaliando sua segurança. Propagandas de produtos fazem asserções que simplesmente não se sustentam. Um produto mais robusto, isto é, melhor testado (e portanto mais caro), estará nestas condições em desvantagem para a comercialização. As pessoas confiam no governo para zelar pela sua segurança e bem estar, em coisas para as quais não detêm conhecimento suficiente para fazerem sua própria avaliação – industrialização de alimentos, aviação, medicamentos, medicina, etc. Com a criptografia entretanto, os governos fazem geralmente o contrário.

## Problemas

Quando cai um avião, são abertos inquéritos, feitas análises e laudos técnicos. Informação sobre o acidente é amplamente divulgada, e muitos aprendem algo com o acidente. Pode-se obter das autoridades, laudos sobre acidentes aéreos desde o início da história da aviação. Mas quando o sistema eletrônico de transações financeiras de um banco é penetrado e fraudado, quase sempre o episódio é acobertado. Se alguma informação chega até os jornais, os detalhes são omitidos. Ninguém analisa o ataque, e ninguém aprende nada com os erros. O banco tenta remendar o problema em segredo, na esperança de que a clientela não perca a confiança num sistema que não merece esta confiança.

Remendar sistemas de segurança para tapar furos em resposta a ataques bem sucedidos não é suficiente. A informação move muito depressa. Uma falha em algum sistema, descrita na internet, pode ser explorada por milhares em um dia. Os sistemas para hoje precisam antecipar futuros ataques. Qualquer sistema de grande porte – seja para comunicações autenticadas, armazenamento seguro de dados ou comércio eletrônico – deveria ter vida útil de cinco anos ou mais. Para permanecer seguro, precisa ser capaz de resistir ao futuro: ataques mais inteligentes, com maior capacidade computacional e motivações crescentes para se subverter um sistema que está consolidado por longo uso. Não haverá tempo para se fazer *upgrades* enquanto este estiver em uso.

A história tem nos ensinado: nunca subestime a quantidade de recursos em dinheiro, tempo e esforço que alguém esteja disposto a gastar para subverter um sistema. Use sistemas de defesa ortogonais, com várias maneiras de se fazer a mesma coisa. Autenticação segura pode significar assinaturas digitais pelo usuário via teclado, SSL para proteger a transmissão, IPSec pelo firewall para o destino, junto com pontos de auditoria múltiplos ao longo do caminho para gerar rastros e produzir evidências. A quebra de alguma parte dará ao atacante uma alavanca, mas não causará o colapso de todo o sistema.

È sempre melhor assumir o pior. Assuma que seus adversários são melhores do que realmente são. Assuma que a ciência e a tecnologia poderão em breve fazer coisas que hoje ainda não podem. Dê a si mesmo um margem de erro. Dê a si mesmo mais segurança do que hoje precisa. Quando o inesperado acontecer, você estará contente por ter agido assim”. (Bruce Schneier)

Creio ser a busca de segurança para a informática semelhante à busca metafísica do homem pelo significado da vida. Um movimento de impulso difuso entre o compreensível e o desejável, no horizonte cambiante do possível. Terei atingido meu objetivo se ao final pudermos reconhecer o contexto onde as ferramentas criptográficas podem ser úteis. Este contexto é formado pelas esferas de atitude, motivação e compreensão dos riscos por parte de quem usa a informática, para dela se obter confiabilidade. Poderemos então conviver, e convencer outros da necessidade de convivência, com o dilema que há na versão digital da segunda lei da termodinâmica, expressa pela equação de Nemeth

### **Segurança = 1 / Conveniência**

A sabedoria de cada um será enriquecida na medida em que puder discernir a dose certa com que uma outra força humana relacionada à segurança – a paranóia – pode contribuir ao delicado equilíbrio desta lei.



# 1: Fundamentos

- **Desafios e demandas sociais por segurança computacional:**

**Safety-** proteção contra acidentes (Leis de Murphey)

**Security-** proteção contra ataques e 'incidentes' (Primeira Hipótese Metafísica de Descartes)

## 1 - Security: desafio 1- Padronização de mecanismos e protocolos

- Instituições financeiras...*Transações eletrônicas.*
- Corporações.....*Gerência, Comércio eletrônico, etc.*
- Telecomunicações *Provisionamento de serviços.*
- Comunidades.....*Internet, Redes proprietárias, etc.*
- Governo.....*Administração, Militar, Espionagem, etc.*

## 3 - Security: desafio 2- Norma jurídica e Norma cultural

<b>Classificada</b> <i>Governos</i> <i>Organizações militares</i>	<b>Sensível</b> <i>Comércio, Indústria</i> <i>Comunidades</i>
<b>Virtual Interna</b> <i>Sistemas operacionais</i> <i>Bancos de dados</i>	<b>Virtual Externa</b> <i>Redes de computadores</i> <i>Telecomunicações</i>

## 3 - Security: desafio 3- Mudança no perfil da impunidade

(Estudo por Securicor Consult. Ltd, Londres, 1993)

- Crimes rastreados dentre ocorridos.....~ 1% ;
- Crimes denunciados dentre rastreados.....~15% ;
- Crimes denunciados com suspeito(s).....~61% ;
- Suspeitos julgados e condenados.....~ 3% ;
- Crimes punidos com prisão..... ~**0,0003%**

# Processo de segurança computacional - Security

- **Demanda básica de mecanismos de proteção:**

<b>Tipo de proteção</b>	<b>Ameaça básica</b>	<b>Ação Indevida</b>
Privacidade	Vazamento ou desvalorização	(Read)
Integridade	Fraude, adulteração ou perda	(Write)
Legitimidade	Acesso indevido à execução	(eXec)
Disponibilidade	Bloqueio ilegítimo de serviço	¬(eXec)

- **Componentes principais do processo:**

## 1 - Política de segurança de dados

- Planejamento - Avaliação e análise de riscos e custos.
- Especificação para implementação de salvaguardas e serviços.
- Atribuição documentada de autorizações e responsabilidades.

## 2 - Serviços básicos para segurança computacional (security)

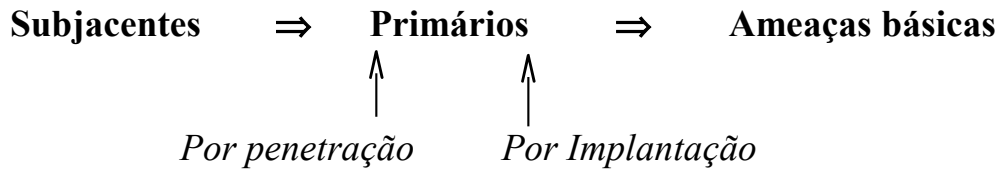
- Autorização.....*identificação para controle de acesso.*
- Cifragem.....*codificação para sigilo ou privacidade.*
- Autenticação.....*validação de origem e/ou integridade de conteúdo.*
- Certificação.....*autenticação recursiva com validação objetiva.*

## 3 - Controle e Auditoria

- Monitoramento.....*gerenciadores (rede, backup) logs, IDS, etc.*
- Rastreamento.....*antivirus, firewalls, proxies, IDS, etc.*
- Avaliação.....*testes de penetração, análise estatística, relatórios, revisão de políticas, de estratégias, etc.*

# Ataques a sistemas informáticos

- **Classificação de ataques (Warwick Ford):**

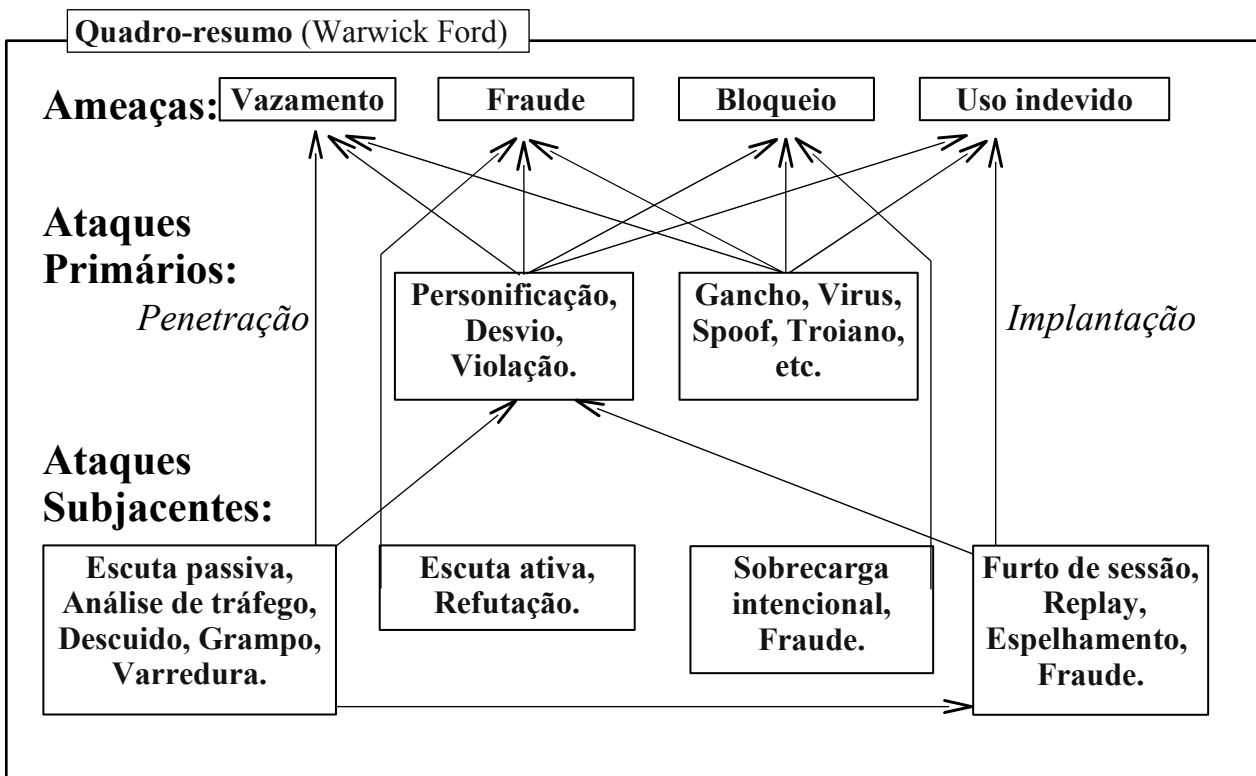


**1 - Ataques primários por penetração -**

- Personificação:.....*invasão no perímetro de login*
- Desvio de controle:.....*exploração de falhas na plataforma*
- Violação de autoridade:.....*uso indevido de privilégios*

**2 - Ataques primários por implantação -**

- Gancho: (*backdoor*).....*uso de recurso não documentado*
- Infecção:.....*exploit, verme, vírus.*
- Embuste:.....*c. de tróia, spoof, smurf, (spear) phishing*



# Vulnerabilidades e pontos de ataque

- **Ataques mais freqüentes a sistemas computacionais em 89-**  
(em ~3000 casos; Computer Security Conference, ordem decrescente)
  - 1º - Violação de autoridade:.....*abuso de usuário legítimo.*
  - 2º - Personificação:.....*uso de senha vazada.*
  - 3º - Desvio de Controle:.....*hacking, cracking.*
  - 4º - Gancho ou Embuste:.....*mascamamento de funcionalidade.*
  - 5º - Grampo, Escuta, Varredura:....*garimpagem no tráfego de dados.*
- **Meios externos de ataque (*Counterintelligence DoD, 94*) -**
  - via Internet (rede aberta).....*80% dos casos*
  - outros meios:.....*20% dos casos*
- **Riscos de invasão de redes (*NCSA, 95*) -**
  - Redes de companhias conectadas à Internet:.....*24% invadidas*
  - Redes privadas não conectadas à Internet:.....*3% invadidas*
- **Recursos que demandam medidas específicas (ver Apêndice B)**
  - Cabeamento
  - Dispositivos de interconexão (*gateways, routers, bridges, etc*).
  - Estações de trabalho.
  - Servidores (de autenticação, de terminais, de aplicativos, etc).
  - Software de rede e aplicativos
  - Arquivos de configuração e de Banco de Dados.

# Tipos de Ataque ao TCP/IP

- **Hierarquia de serviços e protocolos hoje usados na Internet-**

<b>Camada</b>											NFS arquiv	PMAP portas	NIS
											XDR		
<b>Aplic</b>	TEL- NET login	FTP transf. arquivo	SMTP e-mail	HTTP www	<i>Gopher</i> dire- tório	DNS nome domino	NTP sincro- nização	TFTP transf. arquivo	RIP rotea- mento	RPC procedimento remoto			
<b>Transp</b>	TCP					UDP							
<b>Rede</b>	IP												
<b>Enlace</b>	ISO 8802-2				HDLC: ISO 3309 .8885	LAP-B: ITU X.25	ITU Q.921/2 Frame Relay	LAP-D: ITU Q.921 ISDN	SLIP	PPP: rfc 1331	ATM: ITU I.361		
	Ethernet	ISO 8802-3 CSMA/CD	ISO 8802-5 Token Ring	ISO 9314 FDDI								Assíncrona	

- **Riscos nos protocolos de enlace -**

- **Escuta passiva** (*sniffers*) .....*via interfaces em modo promíscuo*
- **Sobrecarga** (*denial of service*)....*via escuta ativa (broadcast storm)*

- **Riscos nos protocolos de rede -**

- **Spoofing** de endereço IP:.....*identificação falsa da origem do pacote*
- **Ataques ao ICMP**:.....*uso malicioso de mensagens de controle do IP (Redirect, Destination Unreachable, Source Quench, etc)*
- **Ataques de fragmentação**:.. *subversão dos filtros de firewall em redes cuja implementação TCP pode reconstruir pacotes fragmentados.*

- **Riscos nos protocolos de rede (continua) -**

- **Ataques de roteamento** (*source routing*):.....*uso de opções do IP para habilitar ataques de escuta ativa, espelhamento ou roubo de sessão.*

- **Riscos nos protocolos de transporte -**

- **Ataques de número sequencial:**...*simulação do handshake para abertura de sessão TCP, conjugado ao spoofing de endereço IP.*
- **Spoofing de UDP:**.....*simulação de datagramas para abertura ou roubo de sessão (sequestro) em aplicativos que usam UDP e que não implementam autenticação e criptografia .*

- **Riscos nos protocolos de aplicação -**

- **Ataques a login remoto:**.. *escuta passiva de sessões TELNET ou "serviços r-" vazam senhas que podem habilitar ataques de personificação*
- **Ataques ao DNS:**.....*modificações fraudulentas de tabelas in-addr.arpa, podem habilitar ataques via serviços remotos "r-".*
- **Ataques ao RIP ou EGP:** *roteadores com filtragem deficiente podem sofrer spoofings que habilitam espelhamento e escuta ativa nas redes.*
- **Ataques via SMTP, HTTP:** *falta de autenticação habilita mensagens forjadas. Extensões habilitam ataques por implantação contra servidor e/ou cliente (SQL injection, Cross-site scripting, etc.)*
- **FTP, TFTP:**.....*configuração e filtragem seguras são complexas. Protocolo usado em quase todo ataque externo via IP.*
- **NIS, NFS, NTP:** .....*fraudes no NTP podem habilitar ataques de replay na rede. Serviços baseados em RPC podem ser alvo de sniffers.*
- **X-Windows, Finger, Whois:** *aplicativos que facilitam outros ataques se mal configurados ou indevidamente habilitados.*
- **Etc...**

# Políticas de segurança

- **Roteiro Típico de planejamento para segurança -**

- Quais recursos e ativos virtuais podem ou devem ser protegidos?
- De quem (*security*) e de que (*safety*) se quer protegê-los?
- Qual a probabilidade de acidentes (*safety*) e incidentes (*security*)?
- Como medir o valor a proteger representado nesses recursos e ativos?
- Quais ações podem protegê-los têm custo/benefício aceitável?
- Que planos de contingência, reavaliação, terceirização, etc. decorrem?

- **Salvaguardas não computacionais -**

- 1 - Segurança física:.....*controle de acesso físico, blindagem, etc.*
- 2 - Segurança funcional:.....*recrutamento, treinamento, motivação*
- 3 - Segurança administrativa:.. *auditoria, fiscalização, contingência*
- 4 - Segurança na mídia:.....*backup, destruição de material, etc.*
- 5 - Radiação ou engenharia reversa: *blindagem no encapsulamento*
- 6 - Controle de ciclos:.....*reavaliação da política de segurança*

- **Serviços básicos (para Security) -**

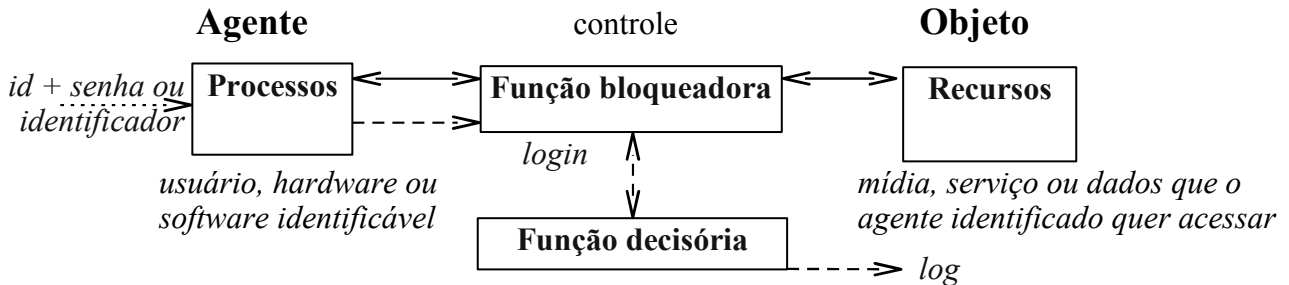
Serviços computacionais básicos, implementáveis com *Primitivas Criptográficas*, para montagem de mecanismos de proteção contra efeitos da Hipótese Metafísica de Descartes. Podem ser classificados:

- 1 - Autorização:.....*serviço básico para controle de acesso.*
- 2 - Cifragem:.....*serviço básico para sigilo ou privacidade.*
- 3 - Autenticação:.....*serviço básico para integridade.*
- 4 - Certificação:.....*serviço básico para autenticação recursiva objetiva.*

# Primitivas criptográficas: Notação

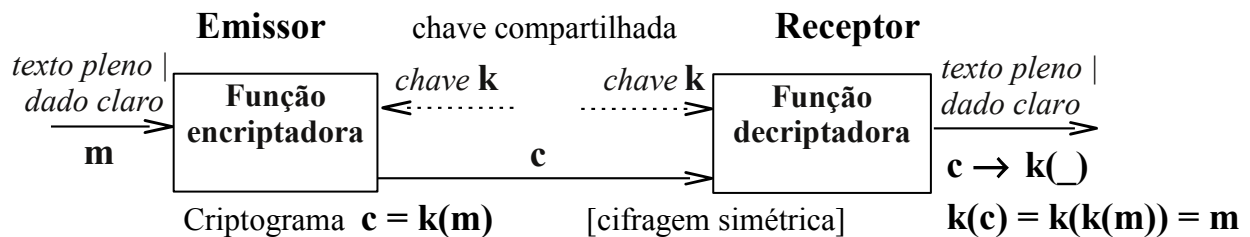
- Autorização:**

para identificação visando principalmente controle de acesso



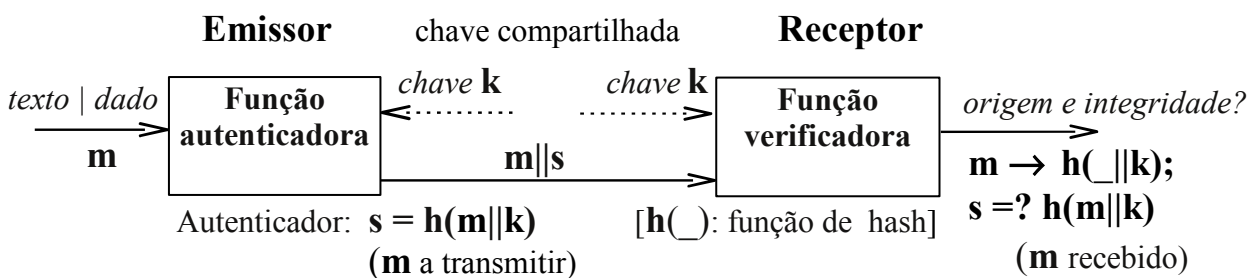
- Cifragem (Encriptação + Decifração) com chave secreta:**

para sigilo durante transmissão – no tempo ou no espaço – de dados



- Autenticação com chave secreta (usando hash criptográfico (MAC) ):**

para validação *subjetiva* de origem e/ou de integridade de dados transmitidos.



**Legenda:**

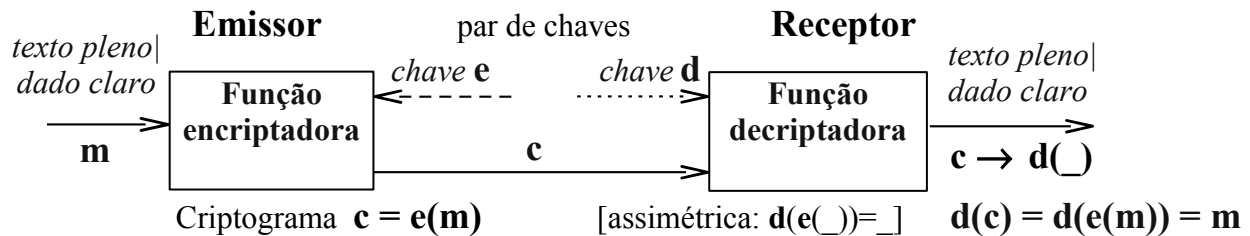
- .....> canais fora de banda
- > canais em banda
- > requer sigilo e integridade
- > requer integridade
- > canal inseguro
- || concatenação
- > 'ou' lógico
- > chave k
- > chave k
- segredo k compartilhado



# Primitivas criptográficas: Notação

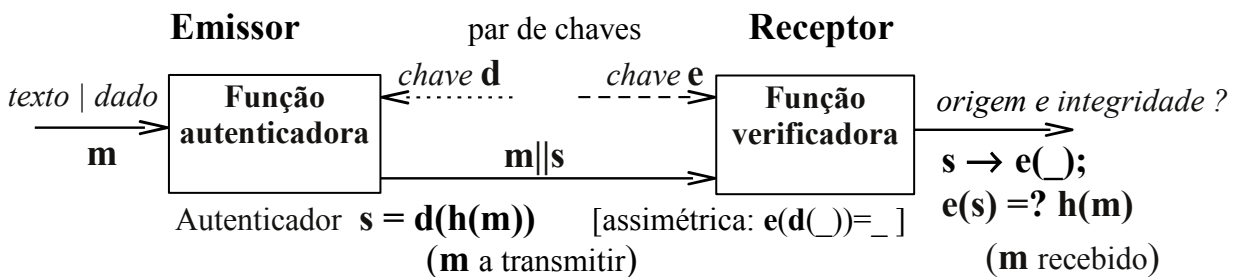
- Cifragem com chaves assimétricas** (pública + privada):

para sigilo durante transmissão de dados sem segredo prévio compartilhado



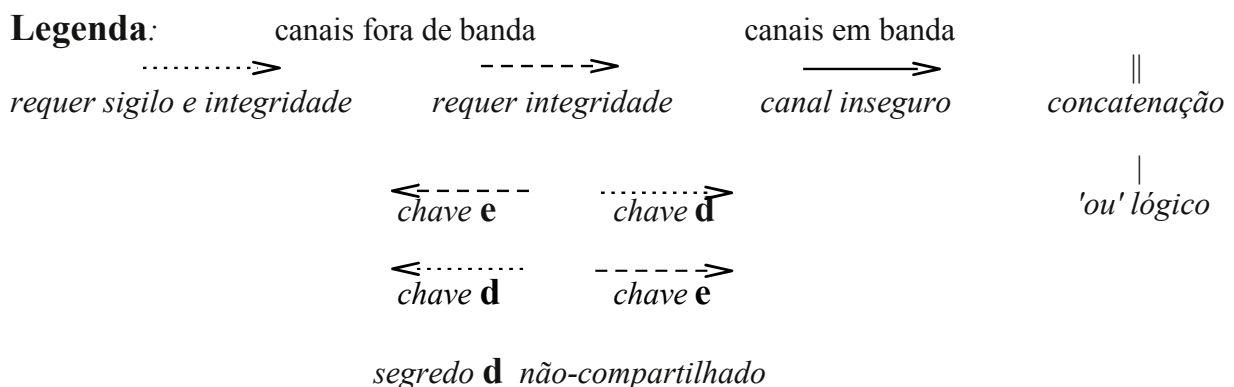
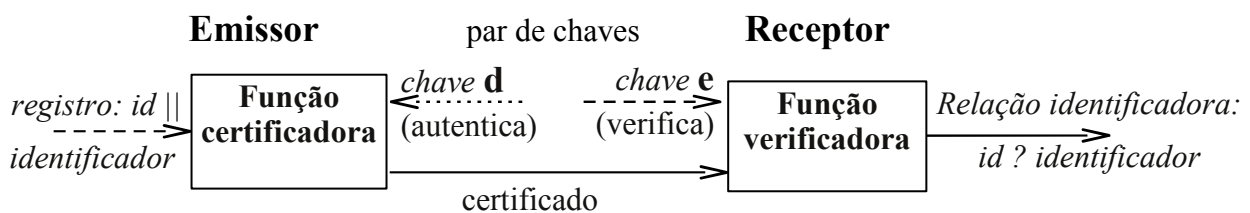
- Autenticação com chaves assimétricas** (privada + pública):

para validação *objetiva* de origem e/ou integridade de dados transmitidos



- Certificação:**

autenticação assimétrica recursiva para transitividade de validações objetivas



# Modelos de Autorização - Controle de Acesso

- **Sistemas de autorização discricionários -**

Baseados no modelo de matriz de acesso de Lampson, vêem o sistema a ser controlado como um conjunto de estados formados por sujeitos, objetos e permissões. As propriedades do mecanismo são definidas como transições de estado permitidas (possíveis autorizações). Aborda a questão pelo aspecto do armazenamento de permissões. (Lampson, B.: "Protection". OS Riview, Jan 1974)

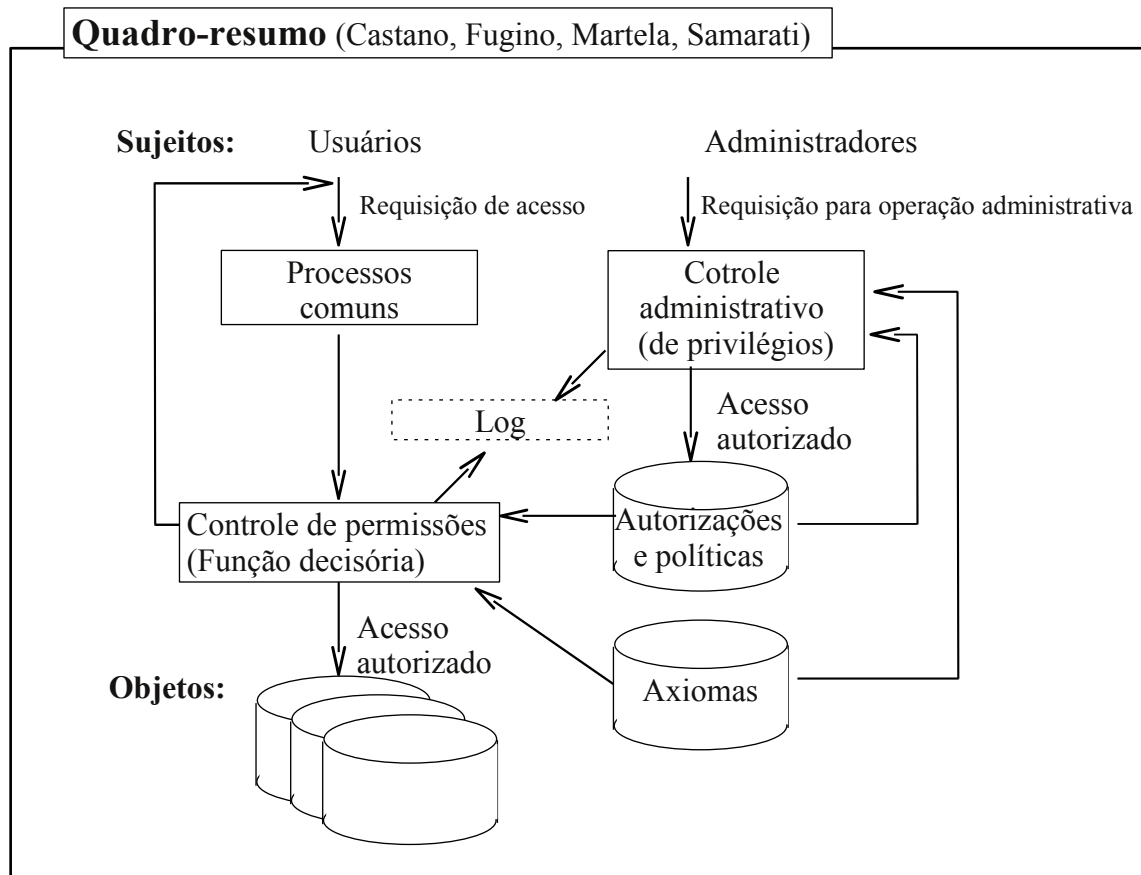
- **Sistemas de autorização mandatórios -**

Baseados no modelo de classificação de relações sujeito-objeto de Bell e LaPadula, vêem o sistema como um conjunto de estados onde as transições possíveis são determinadas por axiomas e regras. Aborda a questão pelo aspecto dos fluxos de informações. (Bell, D. & LaPadula, J.: Lampson: "Security Computer Systems. Mathematical Foundation". MITRE Corp., Bedford, 1974)

- **Elementos de um modelo de controle de acesso -**

- **Sujeitos:**.....*agentes ativos no sistema computacional;*
- **Objetos:**.....*agentes passivos, capazes de representar informações de valor para algum outro agente (inclusive externo ao sistema);*
- **Permissões (modos de acesso):** *formas de acesso que causam fluxo de informações entre objeto e sujeito (e/ou vice-versa) através do sistema.*
- **Autorização:** *conjunto das permissões (modos de acesso), outorgadas a um determinado sujeito do sistema, sobre certo(s) objeto(s) do sistema;*
- **Privilégio (direito administrativo):** *permissão que pode causar modificações em autorizações aplicáveis no sistema.*
- **Política:** *regras para a criação de autorizações, inclusive programáveis para quando forem criados novos objetos ou agentes no sistema;*
- **Axioma:** *política embutida na implementação do controle de acesso.*

# Elementos de controle de acesso



- **Modelos específicos** (C.,F.,M.& S.: "Database Security") -

Modelos para bancos de dados podem exigir adequações, como para tratar múltiplas instâncias de um objeto com distintos requisitos de autorização. Várias extensões dos modelos básicos existem, por exemplo:

- **Harrison-Ruzzo-Ullman**: *modelo de matriz dependente de contexto.*
- **Take-Grant**: *...generaliza o modelo de matriz usando grafos, com enfoque no controle da propagação de autorizações e privilégios;*
- **Wood et al.**: *orientado para a gerência do controle de acesso em BDs multi-camadas que seguem a arquitetura ANSI/SPARC.*
- **Biba, Dion** e outros: *estendem o modelo mandatário para BDs*

- **Modelo híbrido** – RBAC (Role Based Access Control)

Com uma camada adicional, para tipos de sujeitos (papéis), é possível mesclar elementos dos modelos mandatário e discricionário (ex: SELinux)

# Controle de acesso

- **Mecanismos para implementação de controle de acesso:**
  - 1 - **Listas de controle de acesso (ACL):** *banco de dados associado a objetos do sistema, descrevendo autorizações permitidas (controle de acesso com autorização discricionária).*
  - 2 - **Capacidades:**..... *banco de dados associado a objeto(s), descrevendo relações de permissão com outros objetos.*
  - 3 - **Rótulos de segurança:**..... *atributo associado a objeto(s) que determina as possíveis autorizações pertinentes conforme classificação de capacidades (autorização mandatária).*
  
- **Mecanismos para controle de acesso discricionário -**
  - 1 - Alocação de autorizações (*critério de necessidade de conhecimento*).
  - 2 - Autorização e gerência de direitos administrativos (*privilégios*).
  - 3 - Identificação e autenticação de agentes.
  - 4 - Monitoramento de acessos.
  - 5 - Prevenção (on-line) contra acesso não autorizado.
  
- **Mecanismos para identificação:** (ver Apêndice B)
  - 1 - O que (só) o agente pode dizer:....  
*senha, passphrase;*
  - 2 - O que (só) o agente pode fazer:....  
*assinatura de punho digitalizada, timbre de voz, etc;*
  - 3 - O que (só) o agente pode mostrar:....  
*marca biométrica (impressão digital, padrão de retina, etc),  
chave privada, token ou cartão “inteligente”, etc.*

# Classificação de sistemas de controle de acesso

- “**Arco-íris**” (publicações do *National Computer Security Center*)
  - Definem padrões de segurança computacional de sistemas para o *Department of Defense* dos EUA - ISO 15408 (Common criteria)
    - 1 - “**Orange book**” .....DoD 5200.28 STD,.
    - 2 - “**Red Book**” .....NCSC-TG-005,  
*interpreta o orange book no contexto de redes de computadores.*
- **Trusted Computer Standards Evaluation Criteria:**

O *orange book* classifica sistemas stand-alone em classes e níveis.

  - **Classe D - Untrusted**

Nenhuma proteção para o hardware ou para o sistema operacional (Ex. MS DOS, Win95, Win98, MacOS, etc)
  - **Nível C1 - Discretionary Security Protection**

Identifica usuários por login e senha, com permissões de acesso a recursos e dados. Login único “*root*” para administração (Ex.Unix)
  - **Nível C2- Discretionary Access Controls**

C1 com controles adicionais: de acesso por níveis de autorização, de auditoria, e de direitos administrativos. (Ex.Unix comerciais, WinNT)
  - **Nível B1- Labeled Security Protection**

Objetos sob controle de acesso mandatório tem suas permissões pré-codificadas no sistema. (Ex: AT&T V/LMS, UNISYS 1100, HP UX )
  - **Nível B2- Structured Protection**

Todos os objetos acessáveis são rotulados para controle mandatório. Modelo formal de segurança documentado (Ex: Honeywell Multics).

- **Trusted Computer Standards Evaluation Criteria (cont)**

- **Nível B3- *Security Domains Level***

Mecanismos de segurança devem ser modularmente testáveis.

Controle e gerenciamento de memória por hardware.

Mecanismo de restauração e canais de comunicação protegidos em hw.

- **Classe A - *Verified Design Level***

B3 com especificação formal do projeto de software e consistência do modelo de segurança formalmente verificável. Controle na fabricação e transporte do hardware (Ex: Honeywell SCOMP)

- **O nível de segurança C2 -**

Esta classe, que se tornou critério aceitável para padrão de segurança na maioria das aplicações comerciais, tem como característica principal as seguintes propriedades:

- 1 - **Domínio:**.....*sistema operacional protegido, quando em execução, contra vazamentos interprocessos, por meio da compartimentação de memória.*
- 2 - **Kernel do sistema:**.....*protegido contra adulterações em disco.*
- 3 - **Política de segurança:**...*parâmetros configuráveis em níveis de segurança, globalmente aplicáveis no controle de acesso.*
- 4 - **Controle de acesso:**.....*autorização controlada por listas de permissões, com registro configurável de acessos em log*
- 5 - **Autenticação:**.....*com granularidade a nível de objeto, por módulo protegido, com suas operações rastreáveis via log.*
- 6 - **Log:** .....*quando em execução, acesso restrito a níveis de administração e protegido de adulterações em disco.*

# Cifragem para sigilo

- **Área da Criptografia que estuda recodificações reversíveis**

1- **Cifra:** Conjunto  $\mathcal{K}$  de *funções simbólicas inversíveis*  $e : M \rightarrow C$ ;  $e^{-1} : C \rightarrow M$ , onde:

- $m \in M$  pode codificar (representar) um texto  $\mathbf{m}$  numa *linguagem*  $\mathbf{L}$ ;
- $\forall m \in M, \forall e \in \mathcal{K} [e(m) = c \text{ oculta o significado de } \mathbf{m} \text{ em } \mathbf{L}]$ .

- Um par de funções  $e, e^{-1}$  pode ser indexado (ou parametrizado) por uma chave  $\mathbf{k}$  ou por um par de chaves  $(\mathbf{e}, \mathbf{d})$  em  $\mathbf{K}$ .  $M$  é dito *espaço de mensagens*,  $C$ , espaço de criptogramas,  $\mathbf{K}$ , espaço de chaves. [Se  $\mathcal{K}$  é parametrizado, um par de funções  $e, e^{-1}$  pode ser indexado por mais de uma chave  $\mathbf{k}$  (ou par de chaves  $(\mathbf{e}, \mathbf{d})$ ) em  $\mathbf{K}$ ]

2- **Algoritmo de Cifra:** implementação  $f$  de uma cifra  $\mathcal{K}$ ,  $f : \mathcal{K} \times M \leftrightarrow \mathcal{K} \times C$ , onde

- $f$  **encripta**  $m$  :  $f$  calcula  $e(m) = c$  [=  $f(e, m)$ , denotado  $\mathbf{e(m)}$  ou  $\mathbf{k(m)}$ ]; e
- $f$  **decripta**  $c$  :  $f$  calcula  $e^{-1}(c) = m$  [=  $f(e^{-1}, c)$ , denotado  $\mathbf{d(c)}$  ou  $\mathbf{k(c)}$ ].

3- **Criptografia s. s.:** Arte de implementar cifras robustas, i.e., cifras que permitem controle do custo, inviável, para se obter  $\mathbf{m}$  de  $c$  desconhecendo-se  $\mathbf{k}$  (ou  $\mathbf{d}$ ).

4- **Criptoanálise:** Arte de se subverter tal controle (i.e., atacar e *quebrar* cifras).

- **Classificação de Algoritmos de Cifra (por Premissas de Sigilo) -**

1 - **Algoritmo Restrito** (implementa cifra secreta): *Presumido ser desconhecido de quem o ataca: a robustez da cifra depende do sigilo de  $f$  (e das chaves, se existirem, pois neste caso o projeto da cifra ignora o princípio de Kerkhoffs).*

2 - **Algoritmo Simétrico** (implementa cifra de chaves secretas): *Projetado para que a robustez da cifra não dependa do sigilo de  $f$ , mas onde  $\mathbf{d}$  é facilmente dedutível de  $\mathbf{e}$ ; neste caso, um par de funções  $e, e^{-1}$  é geralmente indexado por uma chave  $\mathbf{k}$  (dita secreta): a robustez de  $f$  depende do sigilo de  $\mathbf{k}$ , da equiprobabilidade de qq chave  $\mathbf{k}$  ter sido escolhida em  $\mathbf{K}$ , e de  $\mathcal{K}$  ser grande.*

3 - **Algoritmo Assimétrico** (implementa cifra de chaves públicas): *Projetado para que a robustez da cifra não dependa do sigilo nem de  $f$  nem de  $\mathbf{e}$ ; neste caso, para qualquer par  $(\mathbf{e}, \mathbf{d})$ , tem que ser inviável deduzir-se  $\mathbf{d}$  (chave privada) de  $\mathbf{e}$  (chave pública) ou de  $c$ : a robustez de  $f$  depende do sigilo de  $\mathbf{d}$ , da equiprobabilidade de qq par  $(\mathbf{e}, \mathbf{d})$  ter sido escolhido em  $\mathbf{K}$ , e de  $\mathcal{K}$  ser grande.*

# Criptografia Pré-computacional

- **Cifras de substituição** -

- 1 - **Monoalfabéticas:**  $M=C=(\Sigma)^*$ ;  $\Sigma =$  alfabeto de  $L$  e de  $M$ ,  $f$  substitui cada letra  $m_i$  de  $m$  por  $e(m_i)$ .  $e$  é uma bijeção de  $\Sigma$  no alfabeto de  $C$  (geralmente também  $\Sigma$ ), e  $k$  expressa esta bijeção (por exemplo, por meio de uma lista que emparelha cada letra de  $\Sigma$  com sua substituta, mapeada por  $e(\_)$ ).  
Exemplo: Cifra de César:  $\Sigma = \{ ,a,b,\dots,z\}$ ,  $e =$  rotação de  $\Sigma$  por  $k$  posições.
- 2 - **Poligrâmicas:**  $\dots M=C=(\Sigma^n)^*$ ,  $\Sigma =$  alfabeto de  $L$ ,  $\Sigma^n =$  alfabeto de  $M$  e de  $C = \{\text{blocos de } n \text{ letras de } \Sigma\}$ .  $f$  substitui os blocos de  $n$  letras de  $m$  por outros blocos, conforme mapeados bijectivamente por  $e(\_)$ . (chaves “longas”)
- 3 - **Polialfabéticas:**  $\dots M=C=K=(\Sigma^n)^*$ ,  $f$  é uma cifra poligrâmica que se decompõe em  $n$  substituições monoalfabéticas, indexadas pelas  $n$  letras da chave  $k$ , que se repetem em cada bloco de  $m$  ( $n$  é dito período da cifra).  
Ex: Vigenère (1538):  $e = n$  rotações distintas de  $\Sigma$ , repetidas em cada bloco de  $m$ ; Variante (com XOR):  $f = \text{XOR}$  da chave com cada bloco de  $m$ .
- 4 - **One-time pad (OTP):** polialfabética onde  $n$  é o tamanho da mensagem  $m$ , a chave  $k$  é gerada aleatoriamente e usada apenas uma vez.

- **Cifras de transposição** (usadas na 1ª guerra mundial) -

$M=C=(\Sigma^n)^*$ ;  $K \subseteq \sigma(n)$  (permutação de  $n$  posições).  $f$  permuta entre si as letras de cada bloco de tamanho  $n$  de  $m$ , conforme a permutação de posições definida por  $e \in \sigma(n)$ ,  $e$  indexada por  $k$ . ( $d$  representaria a permutação inversa de  $e$ , a qual é facilmente dedutível de  $e$ , portanto a cifra é simétrica e basta um índice  $k$  para representar o par  $e, e^{-1}$ ).

- **Cifras híbridas** (combinam transposições e substituições) -

Entre 1920 e o final da 2ª guerra mundial, prevaleceram implementações eletromecânicas, inspiradas no odômetro. Com o advento do computador, duas estratégias para projeto de cifras simétricas prevaleceram:

**Cifras de Bloco:** procuram simular poligrâmicas, com chaves “menores”.

**Cifras Encadeadas:** procuram simular OTP, com chaves “expansíveis”.



# Exemplo de um algoritmo criptográfico

- **Implementação (em C) da cifra de Vigenère com XOR**

**k(m):**  $f(e, m) = k \text{ XOR } m$  ; **k(c):**  $f(e^{-1}, c) = k \text{ XOR } c$

```
/* uso:cripto chave arquivo_entrada arquivo_saida */
void main (int argc, char *argv[])
{ FILE *entrada, *saida;
  char *key;
  int c;
  /*programa*/
  if ((key = argv[1]) && *key!='\0') {
    if ((entrada = fopen(argv[2], "rb")) != NULL) {
      if ((saida = fopen(argv[3], "wb")) != NULL) {
        while((c = getc(entrada)) != EOF) {
          if (!*key) key = argv[1];
          c ^= *(key++);          /*XOR*/
          putc(c, saida);
        } fclose(saida);
      } fclose(entrada);
    }
  }
}
```

Considerada robusta até 1820, quando foi descoberto método de ataque estatístico baseado no desvio na frequência de coincidências em deslocamentos sucessivos, quando o deslocamento coincide com o tamanho do bloco (ou seja, o tamanho da chave escolhida para cifragem).

# Ataques a Algoritmos Criptográficos

- **Ameaças da Criptoanálise:**

Neutralização do ocultamento do significado de **m** ou do sigilo da chave de decifração **k** (ou **d**), a um custo viável para o atacante.

- **Tipos de ataques a cifras** (em ordem crescente de severidade) -

- 1 - **Criptograma conhecido:** *atacante tem acesso a  $e(m_i)$ ,  $i = 1, 2, 3, \dots$*
- 2 - **Texto pleno conhecido:** *..acesso a instâncias de pares  $m_i, e(m_i)$ , "*
- 3 - **Texto pleno escolhido:** *....acesso a escolhas de  $m_i$  em  $e(m_i)$ , "*
- 4 - **Texto pleno adaptativo:** *.. escolhas iterativas de  $m_i$  em  $e(m_i)$ , "*
- 5 - **Chave comparada:** *.....acesso a instâncias  $m_i, e_1(m_i), e_2(m_i), \dots$*
- 6 - **Chave comprometida:** *....suborno, extorsão, vazamento de **k** (ou **d**).*

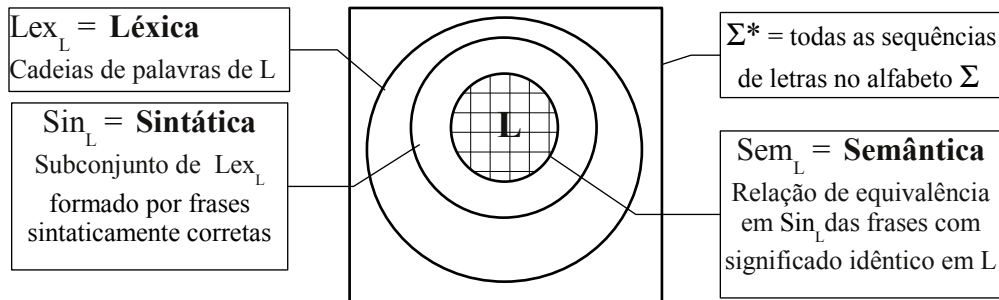
- **Tipos de ameaça a algoritmos criptográficos -**

(em ordem crescente de severidade)

- 1 - **Dedução de informação:** *..criptoanalista ganha informação sobre bits de uma chave, formato ou estatísticas relativas a um texto encriptado.*
- 2 - **Dedução local:** *.....criptoanalista descobre o texto pleno oculto em um criptograma  $c$ .*
- 3 - **Dedução global:** *.....criptoanalista descobre algoritmo que calcula  $e^{-1}(c)$  sem obter ou precisar de conhecer **k** (ou **d**).*
- 4 - **Quebra total:** *.....criptoanalista descobre método eficiente para obter a respectiva chave de decifração **k** (ou **d**).*

# Teoria da Informação aplicada à Criptografia

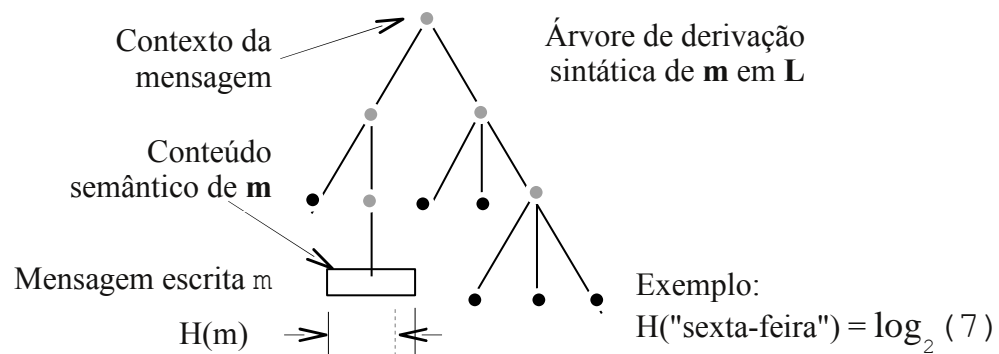
- **Estruturas de linguagens naturais escritas num alfabeto  $\Sigma$ :**



- **Entropia (incerteza) lingüística - (Shannon, 1949)**

## Definições de entropia (como 'medida de informação') numa Linguagem:

- 1 - Definição intuitiva: Dada um texto (sequência de frases)  $m \in L$ , representado por uma escrita  $m \in \Sigma^*$ , a entropia de  $m$  em  $L$ , denotada por  $H(m)$ , é (uma medida do) *espaço mínimo* necessário para se “reescrever”  $m$  numa codificação que, aplicada a  $L$ , seja capaz de distinguir os possíveis significados das frases com a mesma sintaxe de  $m$  em  $L$ .



Intuitivamente, a entropia de uma linguagem  $L$  é o espaço mínimo necessário para codificar mensagens com significado em  $L$

- 2 - Definição formal: Medida logarítmica ponderada do inverso da probabilidade dos possíveis significados de  $m$  em  $L$  dado seu contexto ( $s_i =$  um possível significado de  $m$  conhecido apenas seu contexto).

$$H(m) = -\sum_{i=1}^j p(s_i) \log\left(\frac{1}{p(s_i)}\right)$$

# Entropia em Criptoanálise

- **Taxa  $r$  de uma linguagem escrita  $L$ :**

$$r_L = \overline{H(m) / |m|}$$

Razão média entre a entropia de textos  $m$  de  $L$  e tamanho de suas escritas  $m$ .

- **Redundância  $D_L$  e redundância relativa  $d_L$  de uma linguagem  $L$ :**

$$D_L = \log_2(|\Sigma|) - r_L$$

Redundância relativa  $d_L$  é a normalização de  $D_L$ , dada por  $d_L = D_L / \log_2(|\Sigma|)$ .

$d_L$  é proporcional ao desvio padrão na distribuição de frequência de letras de  $\Sigma$  em textos escritos de  $L$ , e à densidade de suas escritas em  $\Sigma^*$

- **Estimativas de taxa e redundância da lingua inglesa (Schneier)**

$$r_{\text{ENG}} \cong 1.3 \text{ bits / letra } [\Sigma = \{ ,a,b,\dots,z \}];$$

$$D_{\text{ENG}} \cong 3.4 \text{ bits / letra } ; \quad d_{\text{ENG}} \cong 6.7 \text{ bits / character } [\Sigma = \text{ASCII}]$$

- **Distância de Unicidade  $U_{K,L}$  de um sistema criptográfico**

- Um **Sistema Criptográfico** é a aplicação de uma cifra a uma linguagem:

formalmente, um par  $(f : K \times M \leftrightarrow K \times C, L)$ , abreviadamente denotado por  $(K, L)$ .

- A quantidade de distintos significados que poderiam ser ocultados num criptograma gerado com uma chave de  $K$  sobre uma mensagem  $m$  de tamanho  $n$ , isto é, o número de possíveis decifrações em  $L$  de  $e(m)$ , é dada por:

$$2^{(H(K) - d_L * n) + 1}$$

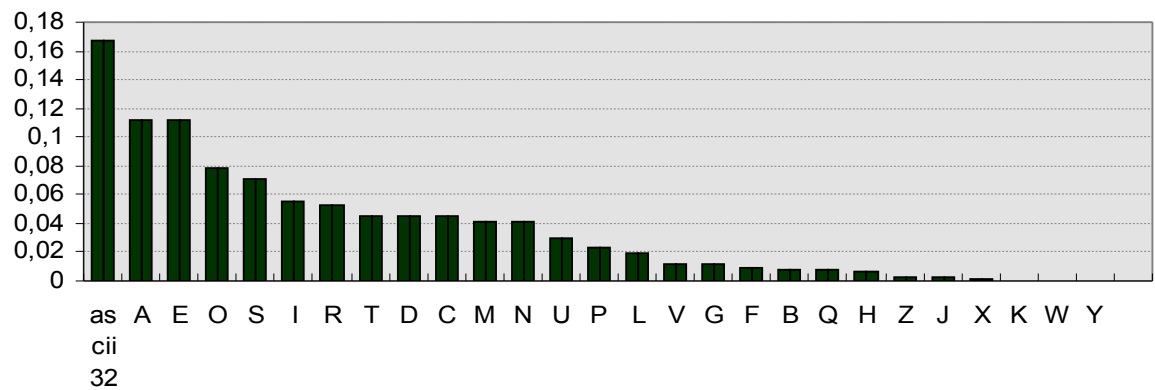
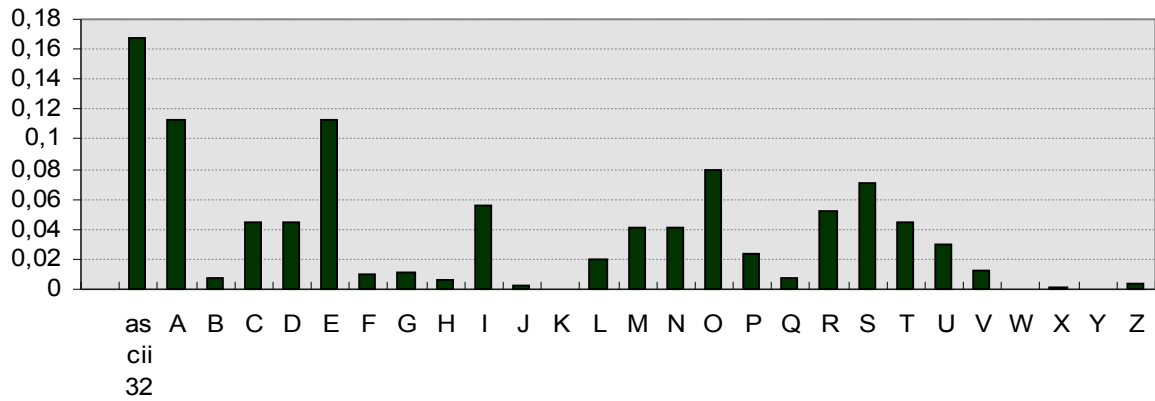
- A **Distância de Unicidade** de um sistema criptográfico, denotada  $U_{K,L}$ , é o limiar para tamanho de mensagens  $m$  acima do qual o número de possíveis significados do correspondente criptograma em  $(K, L)$  é estimadamente  $> 1$ :

$$U_{K,L} = H(K) / d_L$$

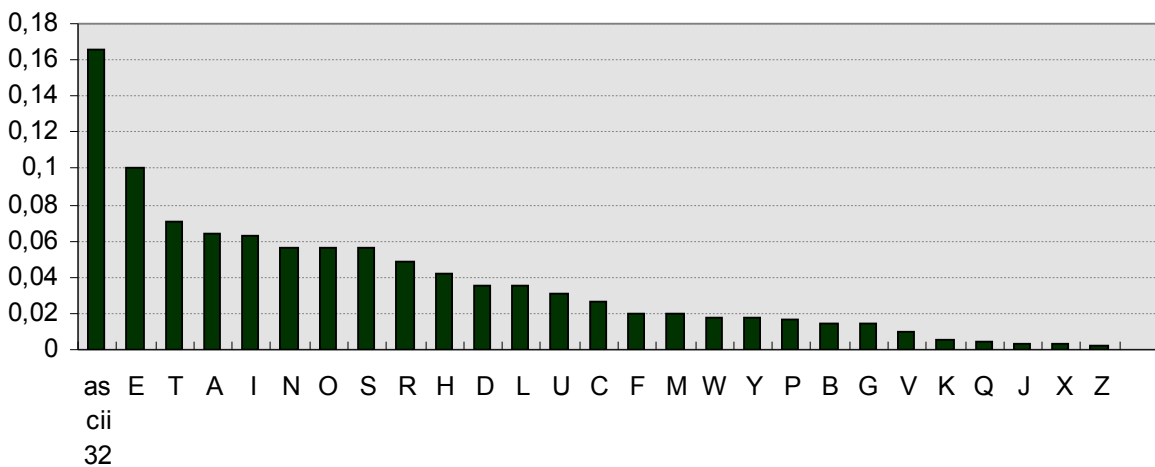
Se  $f$  preserva o comprimento das mensagens, criptogramas de tamanho maior que  $U_{K,L}$  podem, em tese, ser deduzidos estatisticamente; menores, não podem.

# Distribuição de frequência de letras

Frequências do Português (arquivo genérico de 64KB)



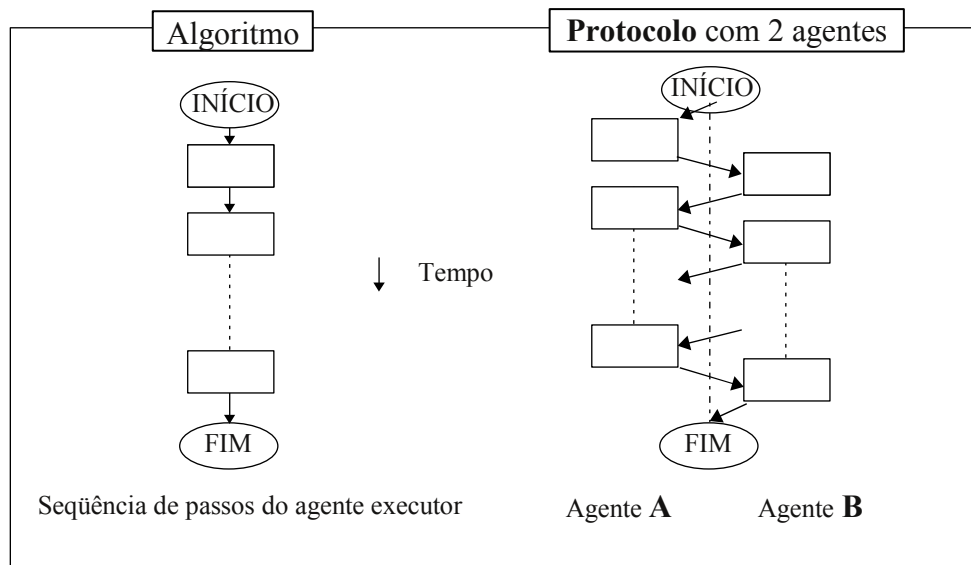
## frequência do inglês



# 2: Elementos de Protocolos

- **O que é um Protocolo?**

É um algoritmo distribuído, em que a execução dos passos é *alternada* entre dois ou mais agentes executores.



- **Premissas implícitas no conceito de protocolo computacional:**

- 0 - Há um *propósito especificado* que motiva a execução do protocolo;
- 1 - Os agentes executores devem *conhecer* a especificação completa,
- 2 - Esses agentes devem concordar em *seguir sua parte* no protocolo;
- 3 - O protocolo deve ser *inambíguo*, sem margem a mal entendidos, e
- 4 - *completo*, com uma ação especificada para cada situação possível;

- **Interface com o Direito:** (Miguel Reale, Marcos da Costa)

- O Direito é composto de três dimensões: Norma, Valor e Fato.
- Com a informática, o que muda é a dificuldade de se estabelecer o Fato
- No Direito, fato de interesse jurídico se estabelece por correlações entre ato, efeito (que causou dano) e autoria (que causou o ato).

# Protocolos Computacionais

- **Especificação de protocolo:**

Uma especificação abstrai o *processo* que realiza o propósito do protocolo, dos mecanismos pelos quais este processo pode ser implementado. Assim, neste capítulo, não mais distinguiremos **m** (texto) de *m* (escrita que codifica **m**), ou **k**, **e** ou **d** (chaves) de *e*,  $e^{-1}$  (funções de cifra correspondentes)

- **Protocolo Criptográfico -**

Tem como propósito proteger algum interesse de certo(s) agente(s) em certos processos de comunicação nos quais esse interesse conflita com alguma característica da comunicação. Usa criptografia para manipular pontos, modos e momentos em que alguma confiança é presumida no processo. Para isso, um protocolo criptográfico pode buscar atingir um ou mais dos seguintes objetivos:

- 1 - Transferir sigilo ou detectar adulteração:.....*relativo ao propósito.*
- 2 - Prevenir vazamentos que o vulneram:.....*durante a execução.*
- 3 - Prevenir ou detectar trapaças ou desavenças: *entre interlocutores.*
- 4 - Prevenir perigosas inferências ou conluios:....*relativo às premissas.*

- **Alocação de Confiança -**

Conforme seu propósito e premissas, um protocolo pode especificar a ação de um agente *auxiliar*, desinteressado a priori no propósito (TTP – ver A-7), para resolver impasses entre agentes *principais* (interessados no propósito).

- 1 - Protocolo **arbitrado**:.....*usa agente auxiliar – TTP – para, durante a execução, impedir ou prevenir trapaças entre principais.*
- 2 - Protocolo **ajuizável**:.....*usa TTP para, após a execução, detectar trapaças entre principais, e/ou para resolver desavenças entre estes.*
- 3 - Protocolo **autoverificável**: *sem TTP, supondo que a execução elimina possíveis vantagens para trapaças ou desavenças entre principais.*

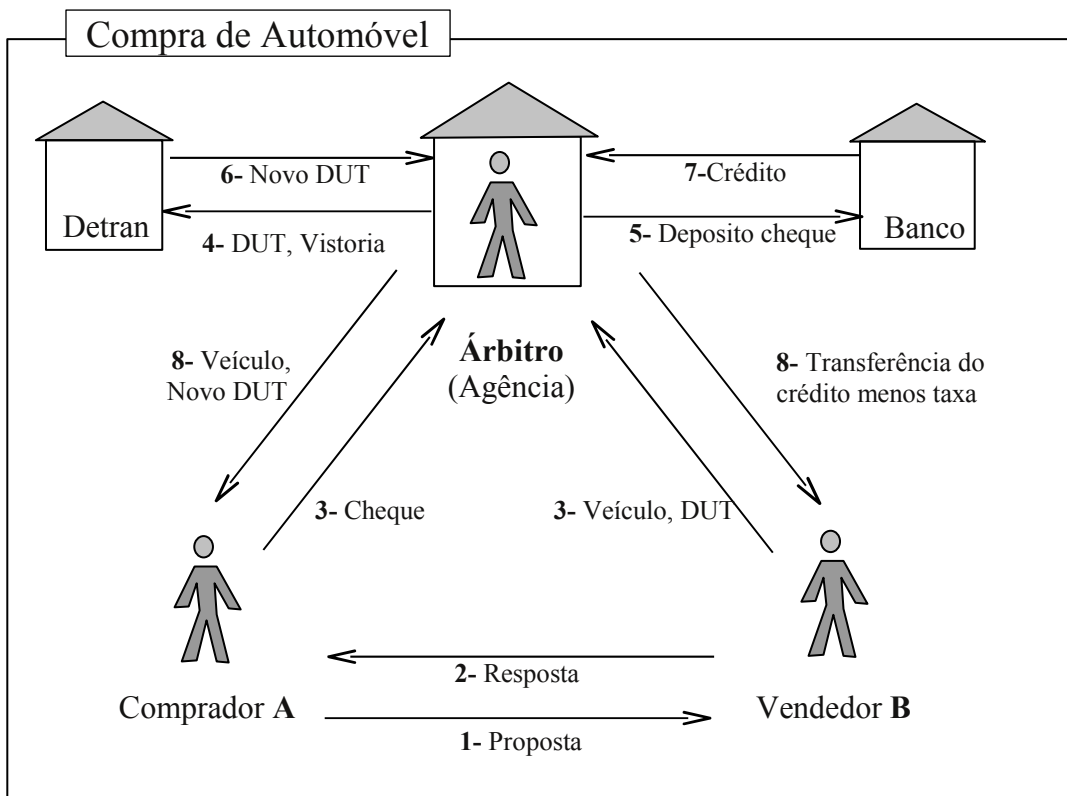
# Ação da Confiança em Protocolos Criptográficos

- **Resolução de impasses -**

Quando o protocolo não possibilita a um agente principal deduzir a lisura dos outros principais ao longo da execução, um TTP tende a nele cumprir função semelhante a uma instituição que goza de 'fé pública' (cartório, tribunal, etc).

- 1- **Arbitragem:**...*decisões do TTP quanto à lisura, em seus papéis, dos agentes principais do protocolo (que não se confiariam em relação ao propósito deste), condicionantes à continuidade da execução do mesmo.*
- 2 - **Mediação:**.....*julgamento pelo TTP de disputas que possam surgir após a execução de um protocolo ajuizável, baseado em evidências produzidas durante execução do mesmo.*

- **Exemplo de protocolo não computacional arbitrado -**





# Protocolos criptográficos

- **Dificuldades da arbitragem em protocolos criptográficos -**

- 1 - Impessoalidade:.....*programas-árbitro nem sempre inspiram a confiança necessária: podem ocultar 'falhas' de programação.*
- 2 - Custo: .....*o custo de criação e manutenção de um serviço de arbitragem geralmente recai no serviço de comunicação.*
- 3 - Gargalo.....*toda transação passa pela arbitragem.*
- 4 - Tempo de execução....*a arbitragem consome tempo considerável.*
- 5 - Vulnerabilidade.....*ponto de alto risco no protocolo, atrai ataques.*

- **Mediação e autoverificação em protocolos criptográficos**

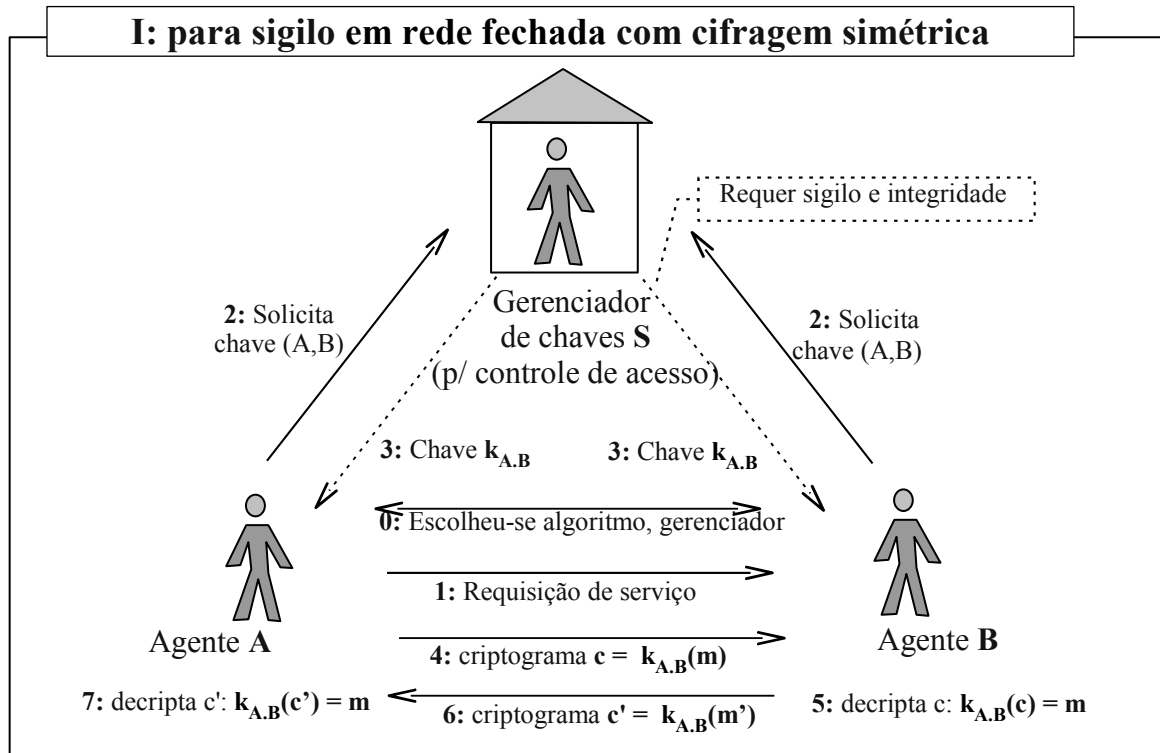
- A mediação apresenta dificuldades listadas acima em 1, 2 e 5.
- A autoverificação apresenta as seguintes dificuldades:
  - 1 - Natureza do problema:..*nem todo tipo de problema de security (premissas de confiança + objetivos de proteção) encontra solução em protocolos criptográficos autoverificáveis.*
  - 2 - Análise exaustiva:.....*a autoverificação só é eficaz enquanto novas formas de ataque que subvertem o protocolo forem desconhecidas.*

- **Conceitos semiológicos fundamentais** (ver Apêndice B)

- **Apresentação:**..*Conhecer algo ou alguém* (por exemplo, via cadastramento);
- **Identificação:**...*Convencer-se de que se re-conheceu algo ou alguém;*
- **Autenticação:**...*Convencer-se, e a outrem(s), de que se reconheceu algo / alguém.*  
**Subjetiva:** *convencer o reconhecido; Objetiva:* *convencer o reconhecido e outros;*
- **Assinatura** (sentido jurídico): *Convencer-se, e a outrem, de que se reconheceu algo que pode representar a manifestação da vontade de alguém.*  
*Para fazer prova jurídica, a assinatura deve convencer terceiros (juiz, juri, etc.)*

# Elementos (subrotinas) de protocolos criptográficos

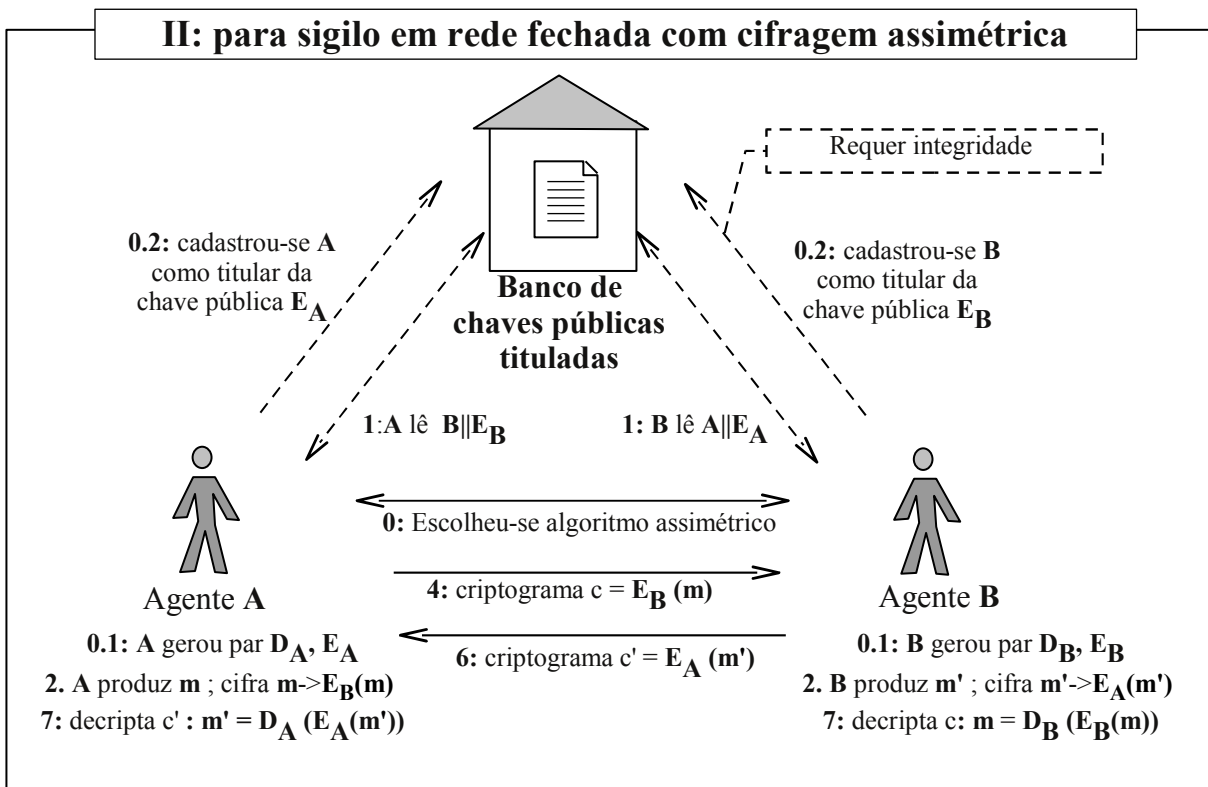
- **Esquema de distribuição de chaves secretas -**



## Premissas e limitações em I:

- a - O esquema I supõe que agentes principais (A e B) confiam um no outro (e no gerenciador de chaves) em relação ao sigilo das chaves.
- b - A transmissão de chaves requer sigilo e integridade. Geradas por um TTP ( $A \neq S \neq B$ ), se o propósito de I incluir controle de acesso (à rede).
- c - I não provê para A a identificação positiva de B, nem vice-versa;
- d - Sem prévia identificação mútua de A e B, ou se uma chave for vazada ou quebrada (comprometida), um atacante poderá personificar A ou B e escalar ataques subjacentes, como escuta ativa, *spoofing*, *replay*, etc.
- e - A privacidade na comunicação em rede com cifra simétrica requer uma chave distinta para cada par de agentes. [ $n^2 - n$  pares, se  $n$  agentes]
- f - A distribuição de chaves pelo esquema I reduz a quantidade de chaves necessárias, mas não pode ser feita pelo canal a ser protegido.

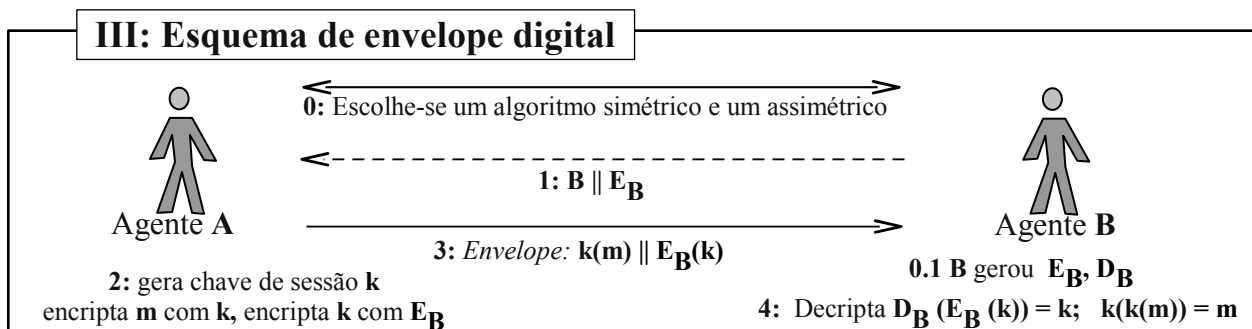
- **Esquema de distribuição de chaves públicas -**



**Premissas e limitações em II:**

- a - A origem e a integridade de  $A||E_A, B||E_B$  (relações identificadoras de A e B) precisa ser certificadas no armazenamento e distribuição das respectivas chaves públicas, para neutralizar ataques de *spoofing*, espelhamento, *replay*.
- b - As chaves públicas distribuídas em II se expõem ao nível 4 de criptoanálise (texto pleno escolhido)
- c - Dentre os algoritmos criptográficos robustos conhecidos, os assimétricos são de  $10^3$  a  $10^4$  vezes mais lentos que os simétricos.

- **Envelope Digital: uso racional das primitivas de cifragem -**

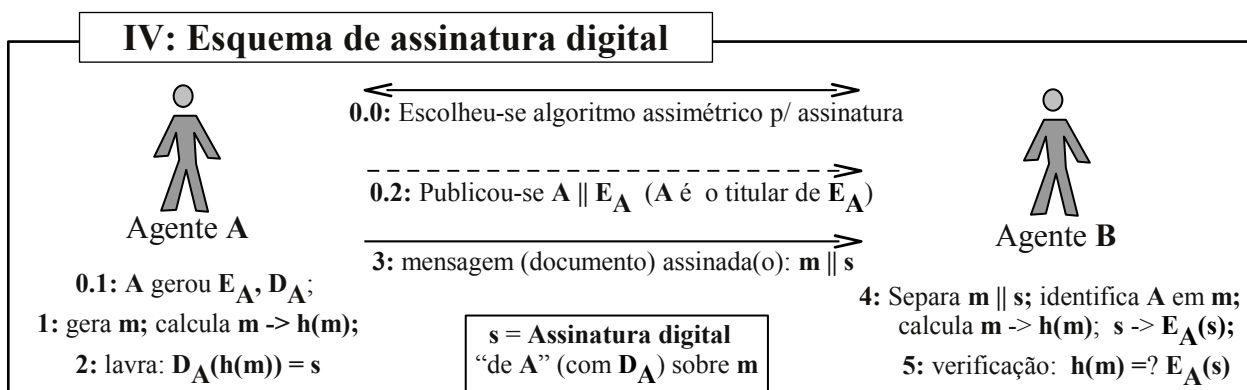


- **Esquema de Assinatura Digital**

**Funções semiológicas da assinatura de punho -**

- 1 - **Inforjabilidade:** *confiança do verificador na identificação de autoria – e do autor, na de possíveis falsificações – de um documento*
- 2 - **Inviolabilidade:** *confiança do verificador na integridade do conteúdo linguístico vinculado, pela assinatura, à vontade do autor.*
- 3 - **Irrecuperabilidade:** *confiança de ambos na inviabilidade de reuso (transferência desse vínculo por troca de conteúdo ou assinatura)*
- 4 - **Irrefutabilidade:** *confiança do verificador na inviabilidade técnica de negação da autoria da assinatura pelo autor (perante ajuizamento)*

- **Autenticação com chaves assimétricas:**



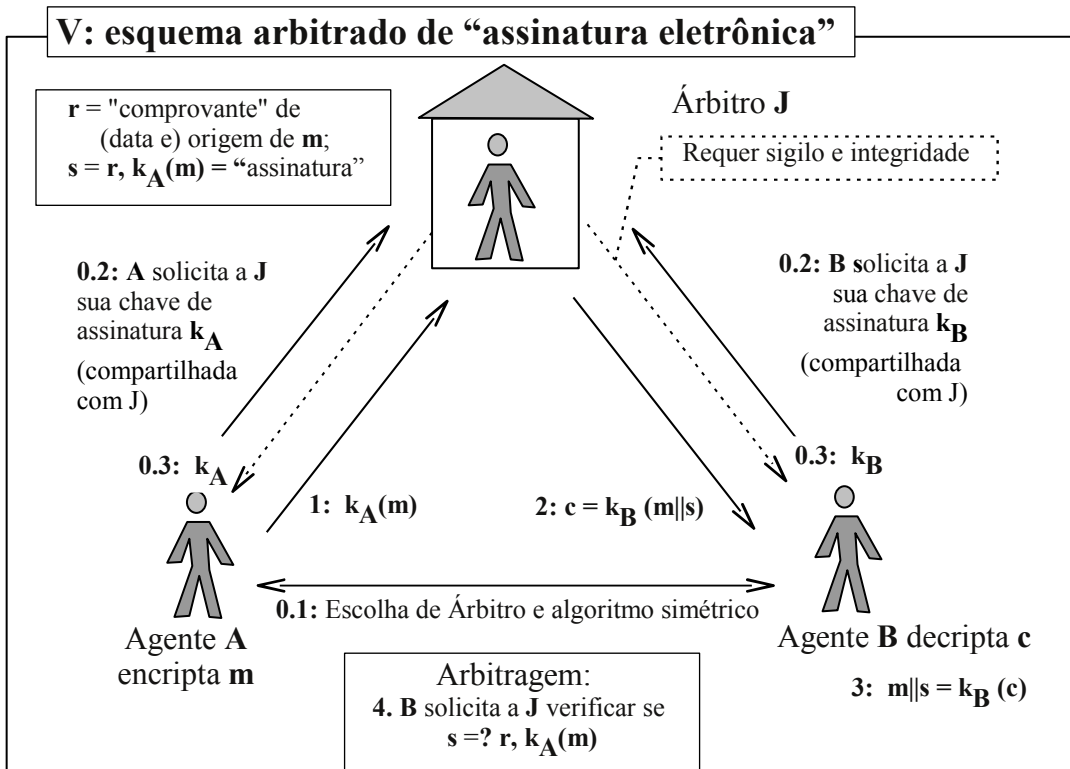
**Premissas de Confiança no esquema de Assinatura Digital:**

- 1- Só o titular de um par de chaves controla o uso de sua chave privada;
- 2- O usuário de uma chave pública confia na titularidade dessa chave;
- 3- Presume-se que o titular manifesta, ou pode manifestar, sua vontade no conteúdo linguístico de um documento assinado com sua chave privada.

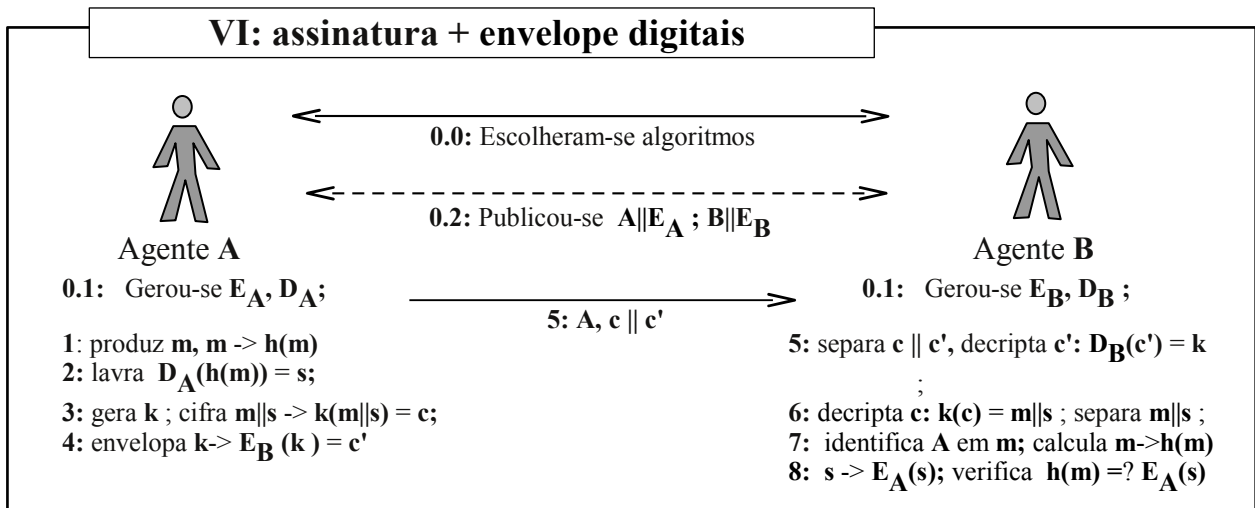
**Objetivo do esquema de Assinatura Digital:**

Autenticação objetiva – no Direito, 'oponível a terceiros' – para validar origem e integridade de mensagens (documentos) assinada(o)s, aproximando as funções semiológicas da assinatura de punho (a irrefutabilidade, apenas precariamente).

- **Autenticação usando cifra simétrica -**



- **Combinando os dois serviços básicos da criptografia assimétrica**



**Premissas de confiança e limitações dos esquemas V e VI:**

V: Árbitro J opera como caixa-preta para verificação indireta (subjetiva).

VI: II.a,b,c, com II.c agravado. Irrefutabilidade relativa, como em IV.

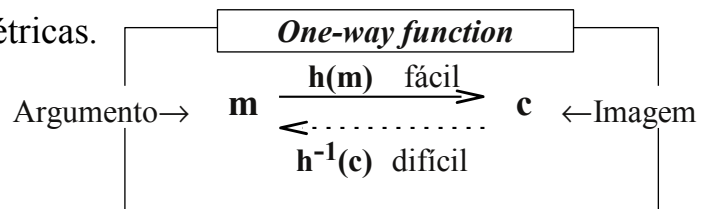
# Cifragem unidirecional

- **Função Unidirecional** (*One-way function*) -

Função simbólica com a qual é fácil (viável) calcular a imagem de um argumento, e difícil (inviável) calcular um argumento (pré-imagem) de uma imagem. Servem para cifragem unidirecional (p. ex., hash criptográfico).

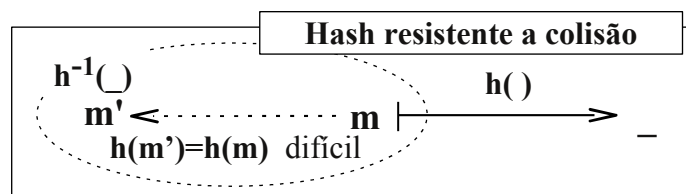
- **Função unidirecional com segredo** (*trapdoor function*) -

É uma função unidirecional para a qual existe alguma informação acerca da imagem que torna fácil o cálculo de sua inversa. É conceito fundamental para geração de chaves assimétricas.



- **Hash criptográfico** -

- **Sinônimos:** *Digest, Fingerprint, ou Checksum Criptográfico; Cheque de Integridade, Cifra Unidirecional, etc.*
- **Definição:** Função unidirecional  $h( )$  com imagens de tamanho fixo e argumentos de tamanho variável, custo computacional de inversão estatisticamente controlado (a níveis inviáveis), e *resistente a colisão*.
- **Resistência a colisão:** Quando é difícil (isto é, estatisticamente controlado a níveis de custo inviáveis) encontrar dois argumentos que produzam o mesmo valor (imagem) dela função de hash (vide *ataque de aniversário*)



- **Código de autenticação de mensagem (MAC):** Esquema em que um hash criptográfico  $h( )$  é usado com chave secreta  $k$ , para validação subjetiva (entre emissor e receptor) da integridade de  $m || h(m||k)$  durante transmissão. (Se um terceiro também conhece  $k$ , essa integridade não pode ser verificada por MAC)

# Autenticação: Resumo sobre uso adequado

- **Checksum, CRC** (*Check Redundancy Code*):

Destinado a detectar erros involuntários de transmissão ou armazenagem de dados (*safety*). Inócuo em canais sujeitos a escuta ativa (ex: rede aberta).

- **Hash, Checksum, Digest ou Fingerprint Criptográfico:**

Função unidirecional  $h()$  com imagem de tamanho fixo e resistente a colisão, usada em esquemas de autenticação que empregam índices para representar mensagens (ou documentos) em primitivas criptográficas.

MAC: mensagem  $\rightarrow$   $\mathbf{m} \parallel \mathbf{h}(\mathbf{m} \parallel \mathbf{k}) \leftarrow$  autenticador da mensagem

Assinatura digital: mensagem  $\rightarrow$   $\mathbf{m} \parallel \mathbf{D}_A(\mathbf{h}(\mathbf{m})) \leftarrow$  autenticador da mensagem

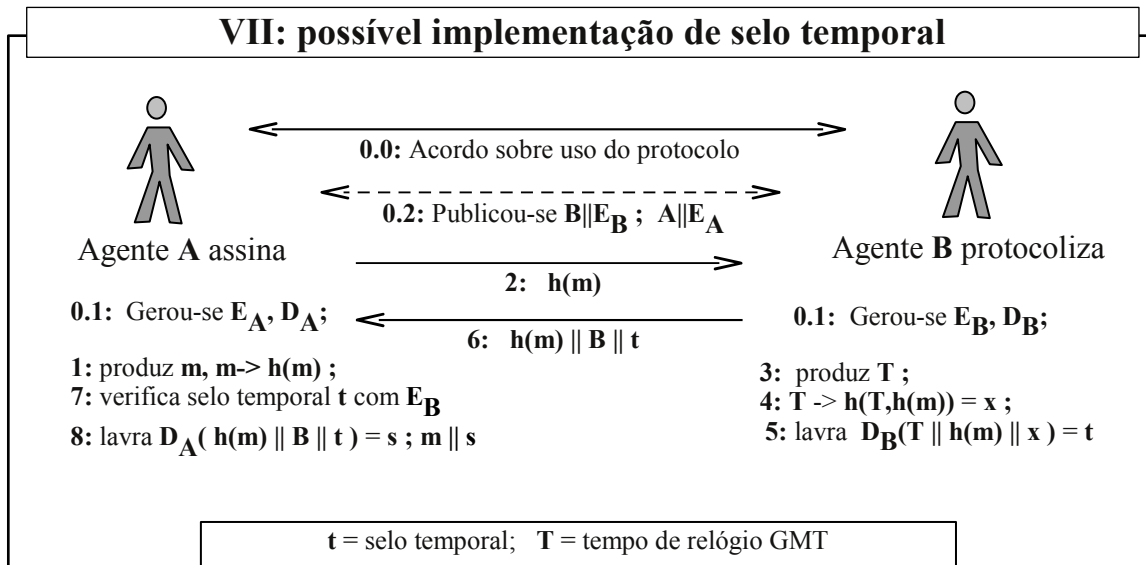
- **MAC** (*Message Authentication Code*):

Esquema de autenticação adequado a protocolos cujo propósito permita que a validação de origem e integridade do conteúdo autenticado possa ser verificado subjetivamente, entre os agentes principais, portanto, com base em segredo compartilhado ( $\mathbf{k}$ ) presumidamente *apenas* entre emissor e receptor. Por exemplo, em protocolos destinados a sistemas ou redes fechadas.

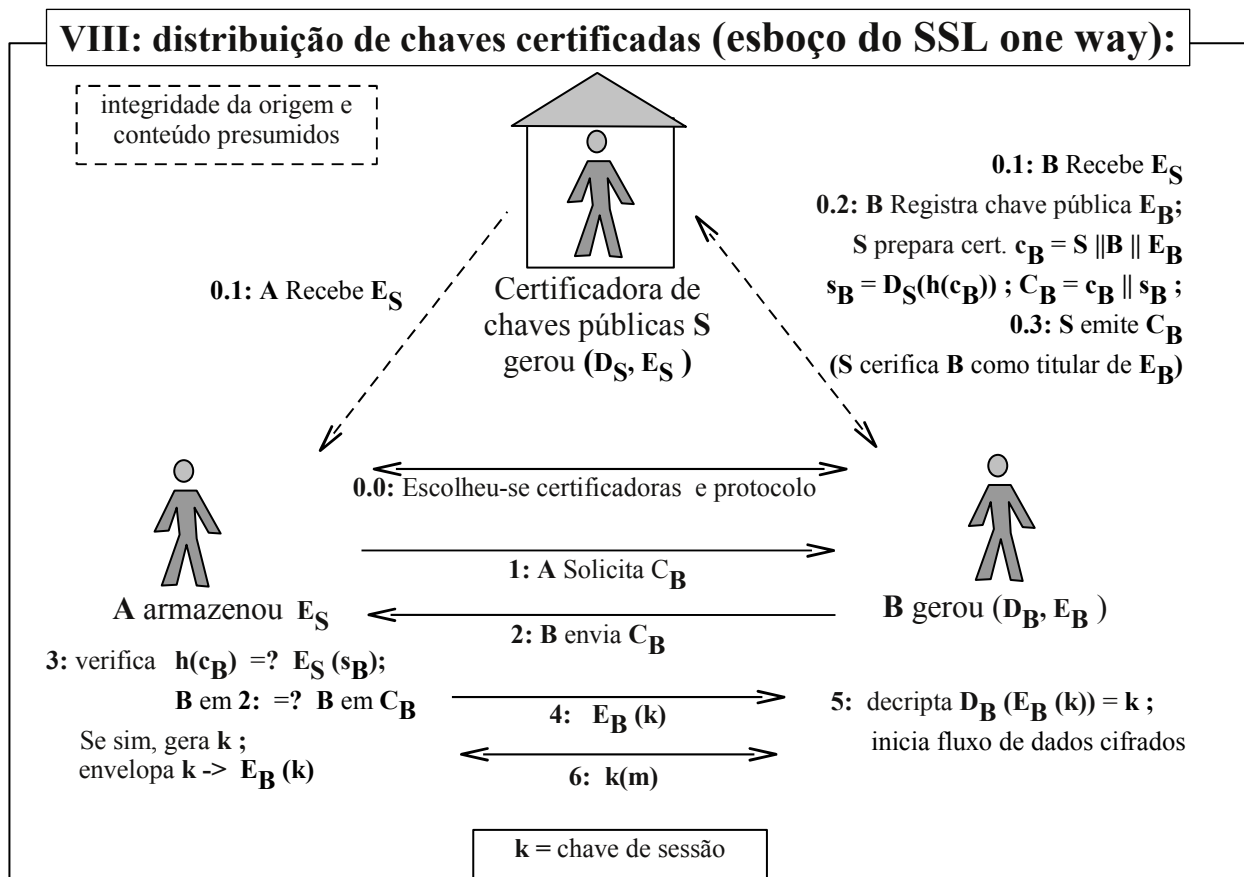
- **Assinatura digital:**

Esquema de autenticação adequado a protocolos cujo propósito exija que a validação de origem e integridade do conteúdo autenticado seja verificável objetivamente, isto é, também por terceiros além dos principais. Portanto, adequado onde tal validação não se baseie em compartilhamento de segredo, e sim em segredo não compartilhado ( $\mathbf{d}$ ). Por exemplo, em protocolos para rede aberta, ou protocolos que devam produzir elementos de prova jurídica. OBS: a *irrefutabilidade* da assinatura digital é de natureza técnica: sustenta-se nas premissas de confiança do esquema e de sanidade da implementação. *Se presentes*, sustentam a *irretratabilidade jurídica* da autoria da assinatura.

- **Prevenção contra ataque de *replay* e certas fraudes temporais:**

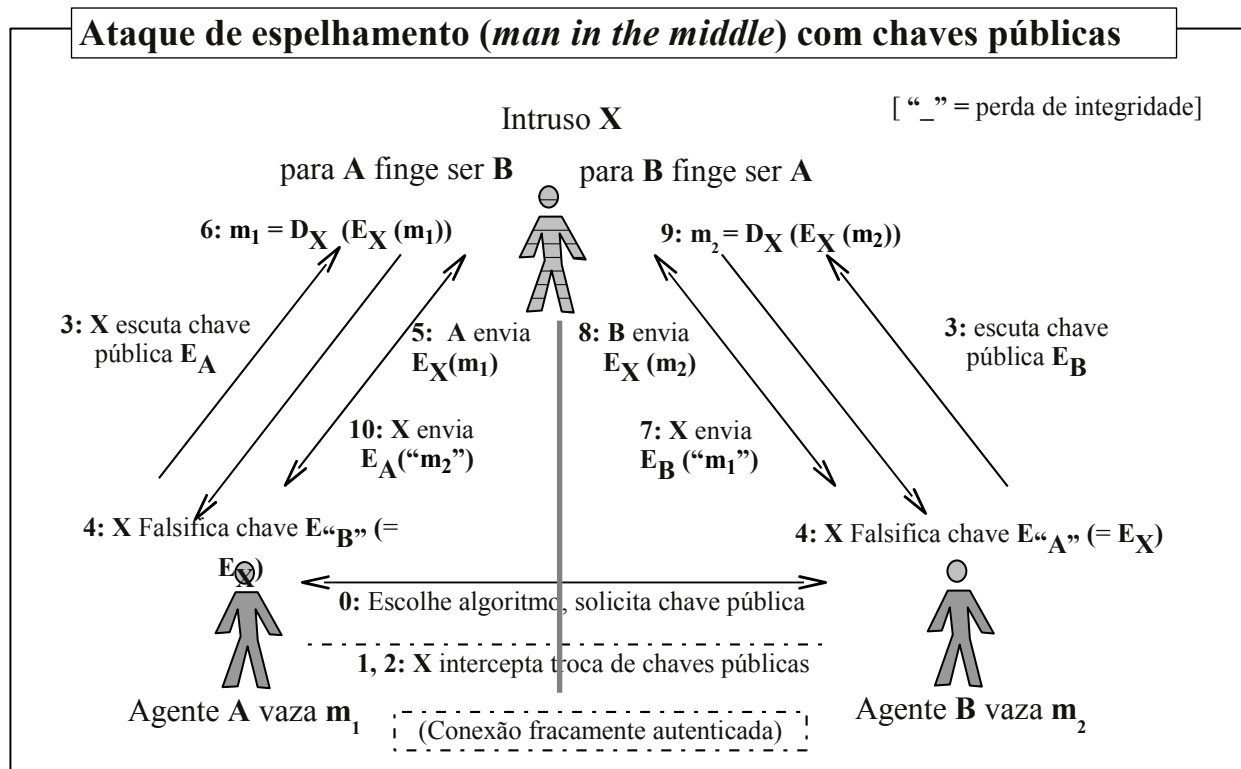


- **Prevenção contra personificação e replay em II.b e III.b**





- **Limitações da autenticação em rede aberta**



- **Resumo de problemas na distribuição de chaves criptográficas**

- a - Espelhamento em I e V pode ser feito por personificação do gerenciador de chaves, e em II, III, IV, VI e VII, por escuta ativa. A eficácia dos protocolos que usam chaves públicas depende de autenticidade destas.
- b - No esquema VIII, o intruso precisa antes atacar S falsificando a titularidade das chaves de A e B, ou fazer *spoofing* do servidor S forjando  $E_S$ , para habilitar o espelhamento entre A e B.

- **Certificado digital:** é um documento eletrônico, emitido e digitalmente assinado por uma entidade certificadora, cujo propósito é atestar uma relação identificadora, representada em seu conteúdo. No padrão X.509 esta relação é entre uma chave pública e um nome: o do titular (*subject*) desta chave e da chave privada correspondente. Neste caso o certificado serve (também) para distribuir esta chave pública, nele contida e titulada, junto a outros atributos dessa titulação, como por exemplo o uso pretendido para esta chave pública.

# Seqüências randômicas

- **Conceito**

Uma definição formal de randomicidade ou aleatoriedade, se existisse, seria paradoxal (o padrão de não seguir nenhum padrão?) A ocorrência ou não de seqüências aleatórias na natureza é uma questão de natureza filosófica, mas que produz uma lista de propriedades necessárias à aleatoriedade. Algumas dessas propriedades são mensuráveis (i.e.: entropia máxima, aperiodicidade, etc.), enquanto outras não (i.e., irreprodutibilidade).

- **Seqüências pseudo-randômicas (B. Schneier)**

São seqüências que exibem as propriedades mensuráveis de aleatoriedade conhecidas. São usadas na criptografia, geralmente onde um protocolo requer um segredo inicial. As principais propriedades mensuráveis são:

- **Distribuição homogênea:** satisfaz medidas estatísticas uniformes, incluindo os modelos de Markov de qualquer ordem.

<i>Subseqüências de mesmo comprimento (bits)</i>	<i>Probabilidade de ocorrência</i>
0, 1	1/2
00, 01, 10, 11	1/4
000, ..., 111	1/8
0000, ..., 1111	1/16
0...00, ..., 1...11	1/2 <sup>n</sup>

- **Entropia máxima:**.....a seqüência não deve ser comprimível por um algoritmo de compressão genérico. (exceto à sua semente, por seu *gerador pseudo-randômico*, se houver)
- **Aperiodicidade:**.....a seqüência não deve conter período (repetições de uma subseqüência de tamanho próximo ao da seqüência)

# 3: Protocolos Importantes

- **Necessidade de protocolos criptográficos**

Criptografia não cria confiança a partir do nada. Suas primitivas são capazes apenas de transformar certas premissas de confiança, em mecanismos de proteção a certos valores, em situações onde tais premissas estejam presentes.

**Autorização**.....*transforma premissa de sigilo e integridade (da senha) ou de posse de um entificador único (token ou chave) em identificação de um agente cadastrado, para fins de atribuição de permissões de acesso.*

**Cifragem**           *transforma premissa de sigilo e integridade de uma chave para a do sigilo de uma mensagem cifrada, durante a transmissão desta.*

**Autenticação**.....*transforma premissa de sigilo (de uma chave secreta ou privada) e de integridade (destas ou de uma chave pública) em validação de origem e integridade em transmissões do conteúdo autenticado.*

**Certificação**.....*transforma premissas de sigilo e integridade (de chaves assimétricas) em autenticação recursiva com validação objetiva.*

Um protocolo criptográfico processa suas premissas de confiança, quando presentes em situações que demandam proteção adequada a seu propósito, para defesa de quem as controla, onde a sua implementação for eficaz.

- **Protocolos “básicos” - Autenticação para controle de acesso**

O desenho de um protocolo básico presume condições mínimas de confiança presentes onde o mesmo irá operar. Essas condições, por sua vez, dependem da arquitetura da plataforma onde a proteção básica é almejada. Protocolos básicos podem ser classificados conforme o tipo de plataforma a que servem

**Login:** autenticação e controle de acesso para *sistema fechado*;

**Distribuição de Chaves:** autenticação e controle para *rede fechada*;

**Public Key Infrastructure:** autenticação e controle para *rede aberta*.