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”

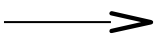
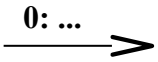
Martin Heidegger, 1995
citado em “Cibercultura”, de André Lemos
Editora Meridional, Porto Alegre, 2002

“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

Notação -

Conceito:	**Negrito**
Explicação; Termo a definir; Anglicismo:	*Itálico*
Descrição de algoritmo:	Font Arial
Objeto matemático; Código fonte em C:	`Font Courier`
Canal de comunicação digital vulnerável:	
Canal presumido protegido:	
Passo inicializador, executado antes do uso pretendido:	
Ponto de confiança para algum agente no protocolo:	
Conjunto das sequências finitas de símbolos do alfabeto Σ:	$(\Sigma)^*$
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 lingúistica 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 ofecerem. 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 freqüência 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 dado 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 freqüência, 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 resuma 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. Ë 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. Ë 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 Havaí. Ë 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 detém 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.

Ë 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 *Provimento de serviços*.
- Comunidades.....*Internet, Redes proprietárias, etc*.
- Governo.....*Administração, Militar, Espionagem, etc.*

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

Classificada *Governos* *Organizações militares*	**Sensível** *Comércio, Indústria* *Comunidades*
Virtual Interna *Sistemas operacionais* *Bancos de dados*	**Virtual Externa** *Redes de computadores* *Telecomunicações*

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:**

Tipo de proteção	Ameaça básica	Ação Indevida
Privacidade	Vazamento ou desvalorização	(**R**ead)
Integridade	Fraude, adulteração ou perda	(**W**rite)
Legitimidade	Acesso indevido à execução	(e**X**ec)
Disponibilidade	Bloqueio ilegítimo de serviço	¬(e**X**ec)

- **Componentes principais do processo:**

1 - Política de segurança (de dados, na informática, etc)

- 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)**:**

Subjacentes ⇒ Primários ⇒ Ameaças básicas

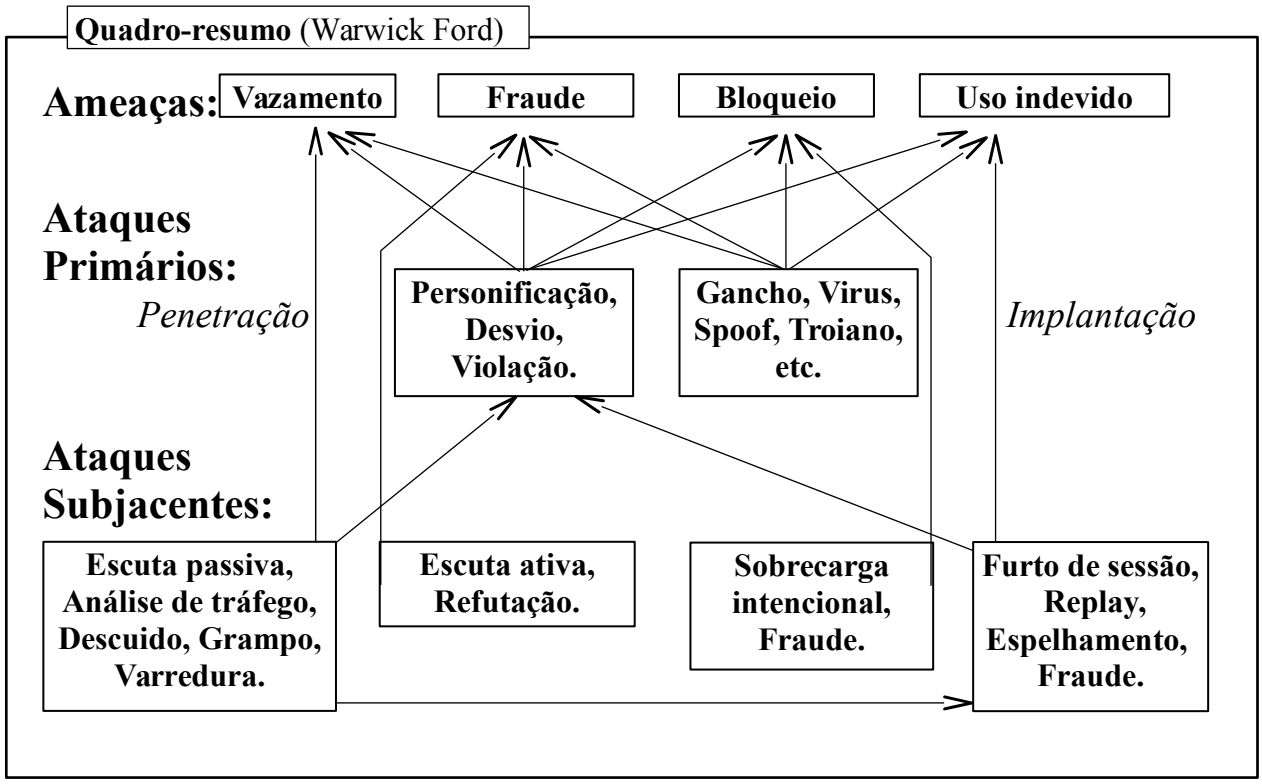
Por penetração *Por Implantação*

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:.....*mascaramento 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**

  - 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-**

<table>
<tr><td rowspan="3">Camada Aplic</td><td colspan="11"></td><td>NFS arquiv</td><td>PMAP portas</td><td>NIS</td></tr>
<tr><td colspan="13">XDR</td></tr>
<tr><td>TEL-NET login</td><td>FTP transf. arquivo</td><td>SMTP e-mail</td><td>HTTP www</td><td>Gopher dire-tório</td><td>DNS nome domino</td><td>NTP sincro-nização</td><td>TFTP transf. arquivo</td><td>RIP rotea-mento</td><td colspan="4">RPC procedimento remoto</td></tr>
<tr><td>Transp</td><td colspan="6">TCP</td><td colspan="7">UDP</td></tr>
<tr><td>Rede</td><td colspan="13">IP</td></tr>
<tr><td rowspan="2">Enlace</td><td colspan="4">ISO 8802-2</td><td rowspan="2">HDLC: ISO 3309 .8885</td><td rowspan="2">LAP-B: ITU X.25</td><td rowspan="2">ITU Q.921/2 Frame Relay</td><td rowspan="2">LAP-D: ITU Q.921 ISDN</td><td rowspan="2">SLIP</td><td rowspan="2">PPP: rfc 1331</td><td rowspan="2">ATM: ITU I.361</td></tr>
<tr><td>Ethernet</td><td>ISO 8802-3</td><td>ISO 8802-5 Token Ring</td><td>ISO 9314 FDDI</td></tr>
<tr><td colspan="2">CSMA/CD</td><td colspan="9">Assíncrona</td></tr>
</table>

.

- **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 nào 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**...

Ver nomenclatura atual padronizada em *https://cve.mitre.org*

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 que 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*

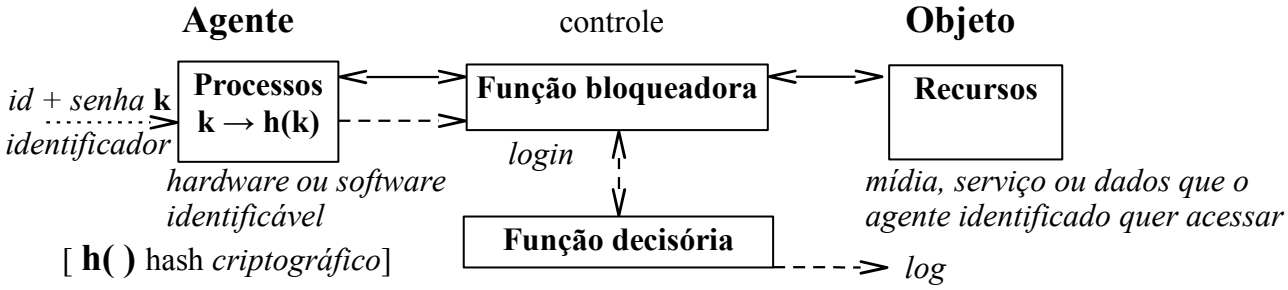
- **Serviços básicos** (primitivas criptográficas de comunicação): Serviços computacionais implementáveis com *Primitivas Criptográficas de comunicação*, para montagem de mecanismos de proteção contra efeitos da Hipótese Metafísica de Descartes. Podem ser classificados em tipos (subtipos):

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

Primitivas criptográficas de comunicação: Notação

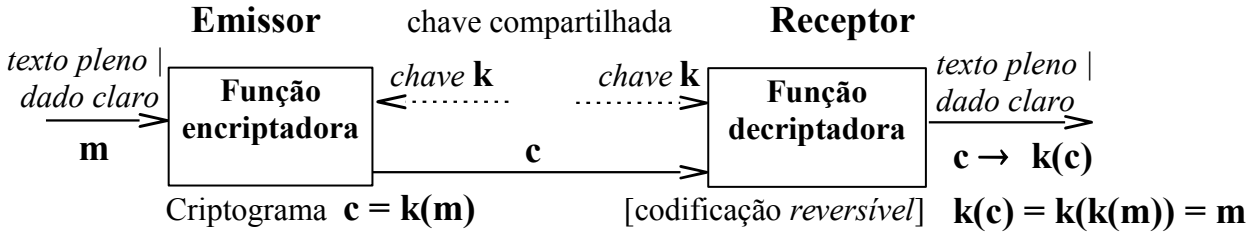
- **Autorização:**

  para identificação e controle de acesso usando codificação não reversível



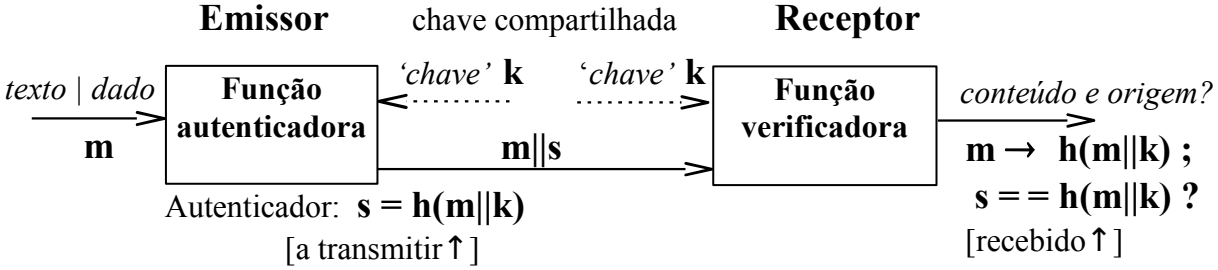
- **Cifragem simétrica** (encriptação e decriptação com chave compartilhada)**:**

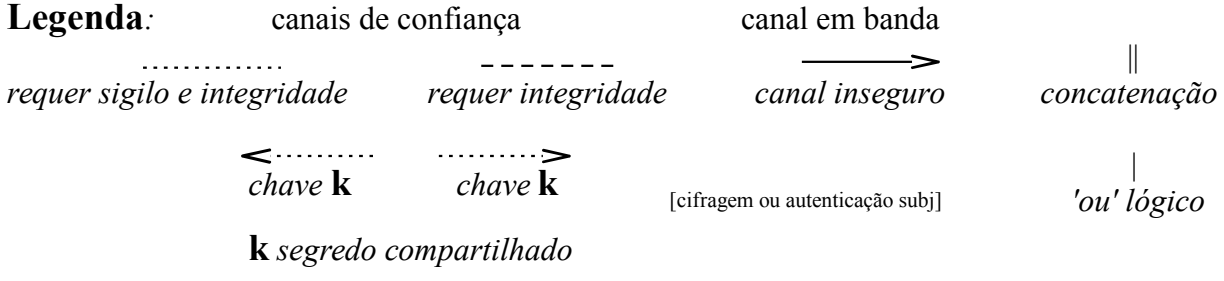
  para sigilo *durante* transmissão – no tempo ou no espaço – de dados



- **Autenticação subjetiva** (com hash criptográfico e segredo compartilhado)**:**

  para validação, *entre interlocutores,* da integridade de conteúdo e da origem de dados transmitidos e recebidos.

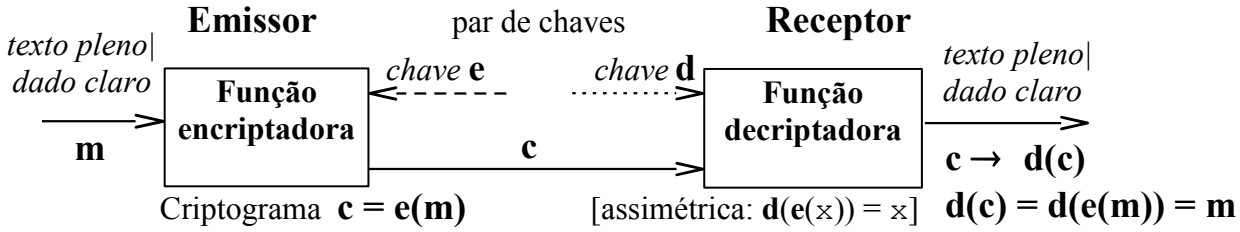




Primitivas criptográficas de comunicação: Notação

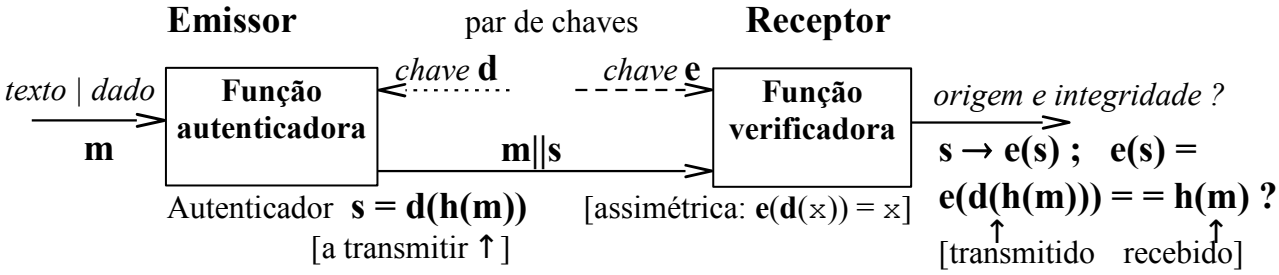
- **Cifragem assimétrica** (com par de chaves pública e privada)**:**

  para sigilo *durante* transmissão *sem prévio* compartilhamento de segredo



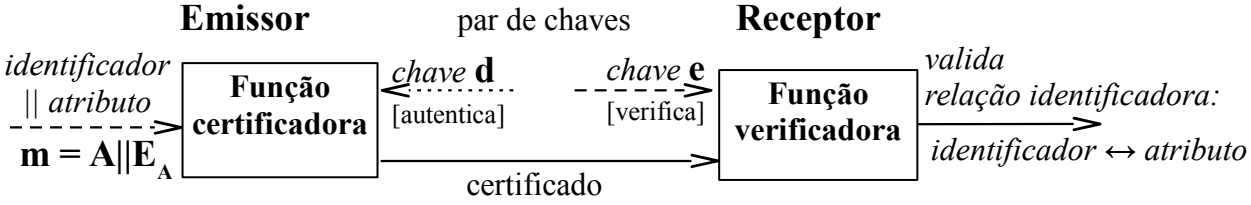
- **Autenticação objetiva** (com par de chaves privada e pública)**:**

para validação, com base em segredo *não compartilhado*, da integridade de conteúdo e da origem de dados transmitidos e recebidos



- **Certificação:**

  autenticação objetiva recursiva para transitividade em validações



Legenda:

canais de confiança		canal em banda	
requer sigilo e integridade	*requer integridade*	*canal inseguro*	*concatenação*
chave **e** / *chave* **d**		[cifragem assimétrica]	*'ou' lógico*
chave **d** / *chave* **e**		[autenticação. objetiva]	

d *segredo não-compartilhado*

Esquemas básicos de Autorização

- **Esquemas de autorização discricionários -**

Baseados no modelo de matriz de acesso de Lampson, que vê o sistema a ser controlado como um conjunto de estados formados por sujeitos, objetos e permissões. O funcionamento do mecanismo é definido por transições de estado permitidas (autorizações). Aborda autorização pelo aspecto do armazenamento editável de permissões. (Lampson, B.: "Protection". OS Riview, Jan 1974)

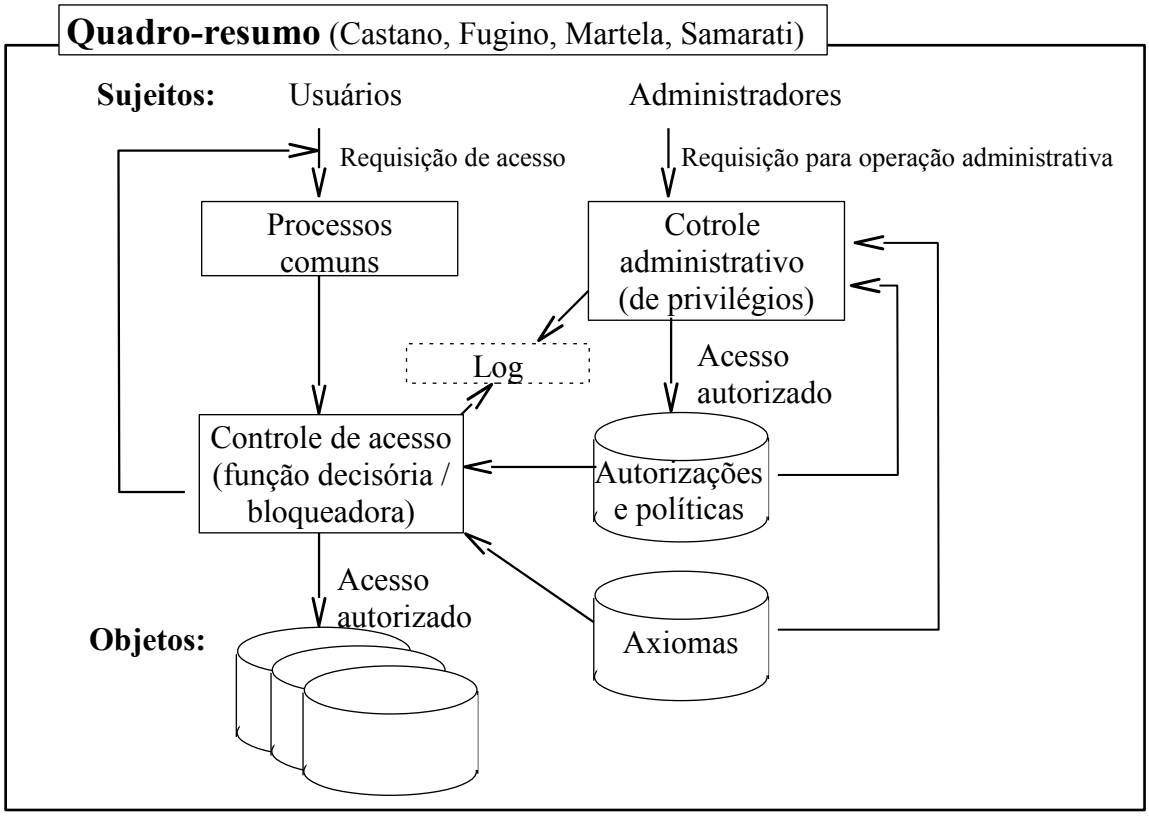
- **Esquemas de autorização mandatórios -**

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

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

  - **Sujeitos**:.....*agentes ou entes ativos no sistema computacional;*
  - **Objetos**:.....*entes passivos, capazes de representar informações de valor para algum outro ente (inclusive externo ao sistema);*
  - **Permissões (modos de acesso)**: *formas de acesso que permitem certo tipo de fluxo de dados entre objeto e sujeito (ou vice-versa) pelo sistema*
  - **Autorização***: conjunto das permissões (modos de acesso), outorgadas a determinado(s) agente(s) do sistema, sobre certo(s) objeto(s) do sistema;*
  - **Privilégio (permissão administrativa)***: 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.*

Modelos de controle de acesso



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

  Subsistemas podem exigir adequações, p.ex. para tratar distintas instâncias de um objeto – nem sempre associado ao conceito de arquivo – com distintos requisitos de autorização. Algumas extensões dos modelos básicos:

  - **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 Banco de Dados (BD) multicamadas que seguem a arquitetura ANSI/SPARC.*
  - **Biba, Dion** e outros: *estendem o modelo mandatório para BDs*

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

  Com uma camada adicional de elementos – a dos "papéis" (*roles*), para tipos de sujeitos –, é possível combinar com mais versatilidade e eficiência elementos dos modelos mandatório e discricionário (exemplo: SELinux)

Controle de acesso

- **Elementos para composição de funções decisória/bloqueadora:**

  - **Listas de controle de acesso (ACL)**: *banco de dados que relaciona agentes e objetos do sistema, descrevendo autorizações como listas de permissões (para controle de acesso discricionário).*
  - **Listas de capacidades**: *banco de dados não editável que relaciona agentes e objetos listando regras para derivação de permissões.*
  - **Rótulos de segurança**: *atributo associado a agente ou objeto (equivalente a "grupo") usado para determinar as possíveis permissões aplicáveis conforme capacidades (controle mandatório).*

- **Elementos para controle de acesso discricionário:**

  1 - Alocação de permissões (*critério de necessidade de conhecimento*).

  2 - Autorização e gerência de permissões administrativas (*privilégios*).

  3 - Identificação e autenticação de agentes.

  4 - Monitoramento e registro de metadados sobre acessos/tentativas.

  5 - Prevenção – on-line – contra acesso não autorizado.

- **Técnicas para identificação**: (ver Apêndice B)

  1 - Baseados em o que (só) o agente pode dizer:...
  senha, passphrase;

  2 - Baseados em o que (só) o agente pode fazer:...
  assinatura de punho digitalizada, timbre de voz, etc;

  3 - Baseados em o que (só) o agente pode mostrar:...
  marca biométrica (impressão digital, padrão de retina, etc), chave privada, token ou cartão "inteligente", etc.

- **Uso de múltiplos mecanismos**:

  Pesquisar "*multi-factor authentication*"

Classificação de sistemas de controle de acesso

- **"Arco-íris"** (publicações do *National Computer Security Center – NCSC*):
  Define padrões de segurança (controle de acesso) de sistemas computacionais para o *Department of Defense* dos EUA - ISO 15408 (*Common Criteria*)

  1 - **"Orange book"**.....DoD *5200.28 STD,* (para sistemas off-line)
  2 - **"Red Book"**.....*NCSC-TG-005,* (para sistemas em rede)
  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** - *Discrecionary 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**- *Discrecionary 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 a partir da década de 1990, 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

Recodificações reversíveis que atendem especificações criptográficas

- **Cifra:** Conjunto K de *funções simbólicas inversíveis* $e: M \rightarrow C$; $e^{-1}: C \rightarrow M$, onde:
  - $m \in M$ codifica **m** – que representa informação – numa *linguagem* **L**;
  - $\forall\, m \in M$, $\forall\, e \in K$ [$e(m) = c$ *oculta* o que **m** *pode representar em* **L**].

  Um par de funções e, e^{-1} pode ser indexado (ou parametrizado) por chave simples **k** ou par de chaves (**e**,**d**) ∈ **K** . M é dito *espaço de mensagens*; C, *espaço de criptogramas*; e **K** (se K é parametrizado), *espaço de chaves*, onde mais de uma chave **k** ou mais de um par (**e**,**d**) podem indexar o mesmo par de funções e, e^{-1}.

- **Algoritmo de Cifra:** implementação f de uma cifra K, $f: K \times M \leftrightarrow K \times C$, onde
  - f **encripta m :** f calcula $f(e, m) = e(m) = c$ [denotado **k(m)** ou **e(m)**];
  - f **decripta c :** f calcula $f(e^{-1}, c) = e^{-1}(c) = m$ [denotado **k(c)** ou **d(c)**].

- **Criptografia *strictu sensu*:** Arte de implementar cifras *robustas*, ou seja, cifras que permitem o controle do custo (até o inviável) de se obter **m** de c desconhecendo-se **k** ou **d.** Custo controlado, sob condições sadias, pelo status das chaves
  Criptoanálise: Arte de se subverter tal controle (arte de se atacar cifras).

Classificação de algoritmos de cifra segundo suas 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 da chave se existir* (se não, $f \simeq$ **k**), pois no caso o projeto da cifra ignora o princípio de Kerckhoffs.

2 - **Algoritmo Simétrico** (implementa cifra de chaves secretas): *Projetado para que a robustez da cifra independa do sigilo de* f, *mas onde* e^{-1} *é facilmente dedutível de* **k** *ou* **e**; Neste caso, o par de funções e, e^{-1} é indexável por chave simples **k** (dita secreta), *e o limite de robustez da* f *depende do status de* **k**: *do seu sigilo*, *de* **k** *ter sido escolhida aleatoriamente*, *e de* $|K|$ (ser grande o bastante).

3 - **Algoritmo Assimétrico** (implementa cifra de chaves pública/privada): *Projetado para que a robustez da cifra independa do sigilo de* f *e de* **e**; Neste caso, para qualquer par (**e,d**) tem que ser *inviável deduzir-se* **d** (chave privada) *de* c *ou de* **e** (chave pública), *e o limite de robustez da* f *depende do status de* (**e**,**d**): *do sigilo de* **d**, *do par* (**e**,**d**) *ter sido escolhido aleatoriamente, e de* $|K|$ (grande o bastante).

Criptografia Pré-computacional

Classificação de algoritmos de cifra simétrica segundo Tipo de Dados:

- **Cifras simétricas de substituição -**

1 - **Monoalfabéticas:** $M=(\Sigma)^*$; Σ = *alfabeto de* **L** *e de* M, f *substitui cada letra* a_i *de* m *por* $e(a_i)$. $e()$ é uma bijeção de Σ no alfabeto de C (geralmente também Σ), e $\mathbf{k} \in \mathbf{K_\Sigma}$ expressa tal bijeção (por exemplo, por meio duma lista de pares que associam cada letra de Σ à sua substituta, mapeadas por $e()$). Ex: **Cifra de César**: $\Sigma = \{a,b,...,z\}$, $e()$ = rotação de Σ por $k \in \mathbb{Z}$ posições

2 - **Poligrâmicas:** $M=C=(\Sigma^n)^*$; Σ = *alfabeto de* **L**, Σ^n = *alfabeto de* M *e de* C = {blocos de n letras de Σ}, $n \in \mathbb{N}$. f *substitui cada bloco de* m *por outro bloco* conforme bijeção entre blocos mapeada por $e()$. [chaves **k** são "longas"]

3 - **Polialfabéticas:** $\mathbf{K}=\mathbf{K_\Sigma}^n$; K *é cifra poligrâmica onde* $e()$ *se decompõe em* n *monoalfabéticas* (indexadas por n subchaves de **k**), *aplicadas às respectivas* n *letras de cada bloco* m_i *de* m (n, $i \in \mathbb{N}$; n: *período de* K). Ex: **Cifra de Vigenère:** $\mathbf{K}=(\Sigma^n)^*$; $e()$: n rotações de Σ, uma para cada posição do bloco, repetidas a cada bloco m_i (**k** = $e(aa...a)$). Variante com XOR: $f = k \text{ XOR } m_i$
One-Time Pad **(OTP):** $n = |m|$; *Polialfabética onde o tamanho do bloco iguala o de* m*, e* **k** *é gerada aleatoriamente para ser usada uma só vez.*

- **Cifras simétricas de transposição -**

$M=C=(\Sigma^n)^*$, $K \subseteq \sigma(n)$; f *permuta entre si as letras do bloco, conforme a permutação de posições definida por* $e \in \sigma(n)$*, aplicada a cada bloco* m_i *de* m, $|m_i| = n$. (**d** representaria a transposição inversa e^{-1}, que *é facilmente dedutível* de um índice para e, portanto a cifra é simétrica, em cujo caso basta usar chave simples **k** para indexar qualquer par e, e^{-1}).

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

De 1920 ao fim da 2ª guerra mundial, prevaleceram implementações eletromecânicas, baseadas no odômetro (Ex: cifra Enigma). Com os computadores, 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 algoritmo de cifra simétrica:

- **Uma implementação da cifra de Vigenère** (variante **XOR,** em C)

  k(m) : `f(e,m) = k* 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;
   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);
      }
   }
}
```

Vigenère

Cifra publicamente conhecida desde 1528 e considerada robusta até ~1850s, quando foi descoberta uma técnica estatística de ataque baseada em análise de frequência de coincidências equidistantes no criptograma (ver Exercicio 1, Ap A-20)

Contando-se as coincidências equidistantes para sucessivas distâncias, a contagem pode superar significativamente a média se a distância amostrada coincidir com o tamanho do bloco (igual ao tamanho da chave escolhida para cifragem) [teste *kappa*]

Ataques a Algoritmos Criptográficos

- **Ameaças da Criptoanálise:**

  Desocultamento da informação expressa por **m**, ou quebra do sigilo de chave **k** ou **d**, ou produção de colisão utilizável para subverter o uso criptográfico de codificações não inversíveis **h**, a custo *viável* para o atacante (isto é, a um custo que lhe compense considerando o valor que estaria sendo protegido pelo sigilo de **m** em c ou pela integridade de transmissões protegidas com **h**).

- **Tipos de ataque a algoritmos de cifra** (severidade crescente) **-**

  1 -**Criptograma conhecido**:. .*atacante tem acesso a* $c_i = e(m_i)$, $i = 1, 2,...$

  2 -**Texto pleno conhecido**: *acesso a instâncias de pares* m_i, c_i, $i = 1, 2,...$

  3 -**Texto pleno escolhido**:.....*acesso a escolhas de* m_i *em* $e(m_i)$, $i = 1, 2,...$

  4 -**Texto pleno adaptativo**:....*escolhas iterativas de* m_i *em* $e(m_i)$, $i = 1, 2,...$

  5 -**Texto pleno comparativo**: *escolhas de instâncias* m_i, $e_1(m_i)$, $e_2(m_i)$,...

  6 -**Chave comprometida**:.....*suborno, extorsão, vazamento de* **k** ou **d**.

- **Tipos de ameaça a algoritmos de cifra** (severidade crescente) **-**

  1 -**Dedução de informação**:.....*criptoanalista deduz informação sobre bits da chave* **k** *ou* **d**, *ou sobre probabilidades relativas ao formato de* m *ou a símbolos da sequência* m *encriptada em* c.

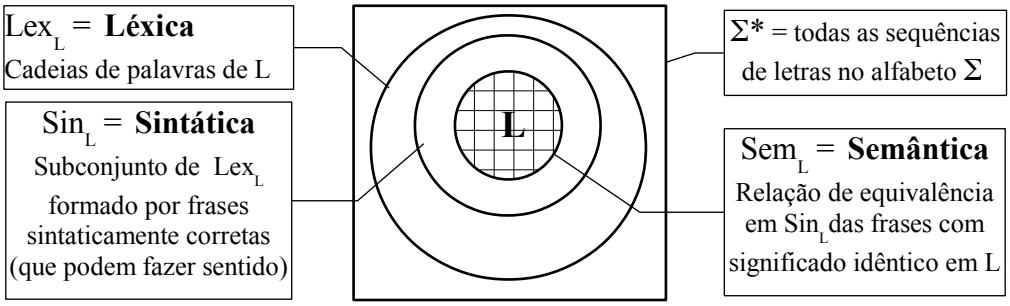
  2 -**Dedução local**:.....*criptoanalista deduz, com probabilidade que lhe satisfaça, o texto pleno* m *que teria sido encriptado em* c.

  3 -**Dedução global**:.....*criptoanalista deduz algoritmo que calcula* $e^{-1}(c)$ *sem precisar obter ou deduzir* **k** ou **d**. (*caso*, *por exemplo, a cifra ignore o princípio de Kerckhoffs e não parametrize chaves*)

  4 -**Quebra total**:.....*criptoanalista descobre método eficiente para obter a respectiva chave* **k** ou **d** *a partir de ataques (tipos 1 a 5).*

Teoria da Informação aplicada à Criptografia

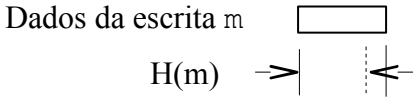
- **Estruturas de linguagens naturais escritas num alfabeto Σ:**



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

Definições de entropia como medida de informação numa Linguagem:

1 - Definição intuitiva: Dado uma mensagem (ou texto) **m** ∈ **L** representável por uma escrita m ∈ Σ*, a entropia de **m** em **L**, denotada por H(m), é uma medida do *tamanho mínimo* necessário para se "reescrever" **m** numa codificação que seja capaz de distinguir, em contexto "geral" de **L**, entre os possíveis significados das frases com mesma "estrutura" (sintaxe) de **m**.



Exemplo:
H("sexta-feira") = $\log_2(7)$

Intuitivamente, a entropia de uma linguagem **L** é uma medida linear de espaço mínimo necessário para se (re)codificar mensagens (ou textos) sem perda de informação em um contexto geral de **L**

2 - Definição formal: Média logarítmica ponderada do inverso da probabilidade dos significados possivelmente expressáveis por m em **L**, considerando-se um contexto minimal anterior à transmissão de m. (Ex: algoritmos de compactação atuam em contexto léxico) [s = possível significado]

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

Entropia linguística aplicada à Criptoanálise

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

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

Razão média entre entropia de mensagens **m** e tamanho de suas escritas m.

- **Redundância D_L (de L escrita com Σ); Redundância relativa d_L:**

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

Redundância relativa d_L é a normalização binária de D_L: $d_L = D_L / \log_2(|\Sigma|)$. d_L é proporcional ao desvio padrão na *distribuição de frequência de letras* de Σ nos textos produzíveis em linguagem **L**, e à densidade de suas escritas em Σ*

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

$r_{ENG} \cong 1.3$ bits / letra ; $D_{ENG} \cong 3.4$ bits / letra ; [Σ = { ,a,b,...,z}] ;

$d_{ENG} \cong 6.7$ bits / caractere [considerando Σ = ASCII]

- **Sistema criptográfico (K,L) e sua Distância de Unicidade $U_{K,L}$**

- Um ***Sistema Criptográfico*** é a aplicação de uma cifra a uma linguagem: formalmente, um par ($f: KxM \leftrightarrow KxC$, **L**), abreviadamente denotado por (**K,L**)
- Em (**K,L**), a estimativa da quantidade de distintas mensagens de tamanho n que podem, *em tese*, se ocultar num mesmo criptograma gerável com distintas chaves de **K**, i.e. o número de possíveis decriptações de c em **L**, é dada por:

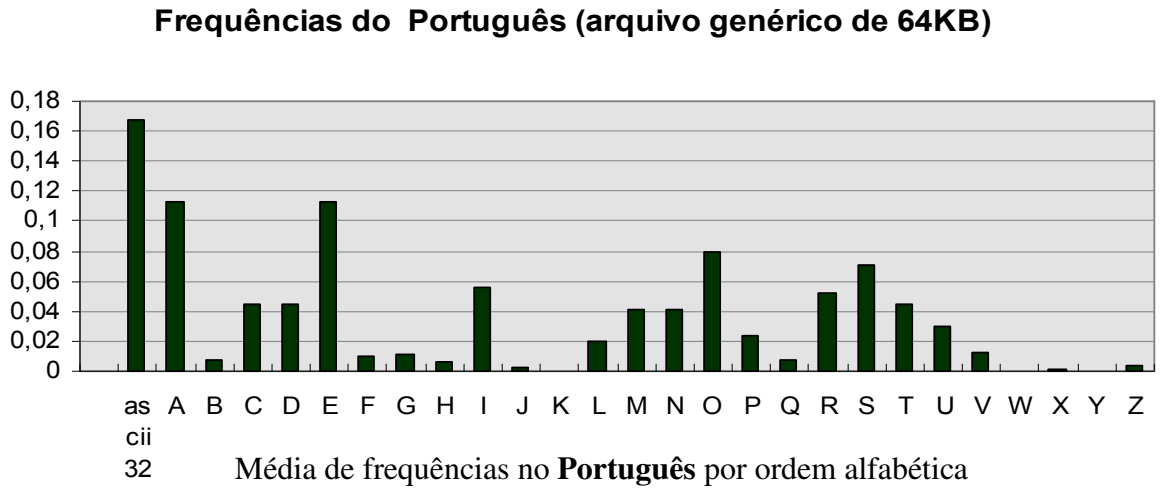
$$2^{(H(K) - d_L * n)} + 1$$

- A ***Distância de Unicidade*** de um sistema criptográfico, denotada $U_{K,L}$, é o limiar para tamanho de mensagens $|m|$ abaixo do qual o número de possíveis mensagens sob um mesmo criptograma c em (**K,L**) é significativamente > 1:

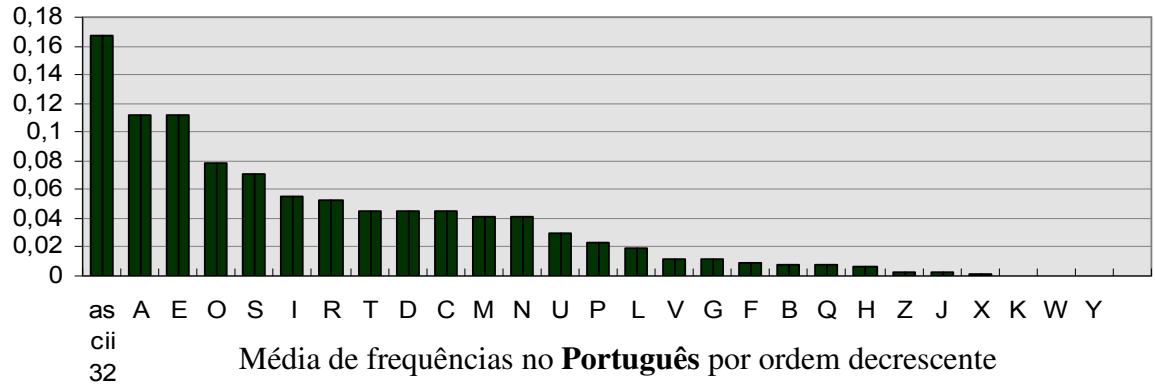
$$U_{K,L} = H(K)/d_L$$

Se f preserva em c o comprimento de m, criptogramas onde $|c| > U_{K,L}$ podem, *em tese*, ter seus **m** deduzidos estatisticamente; e onde $|c| < U_{K,L}$ não podem.

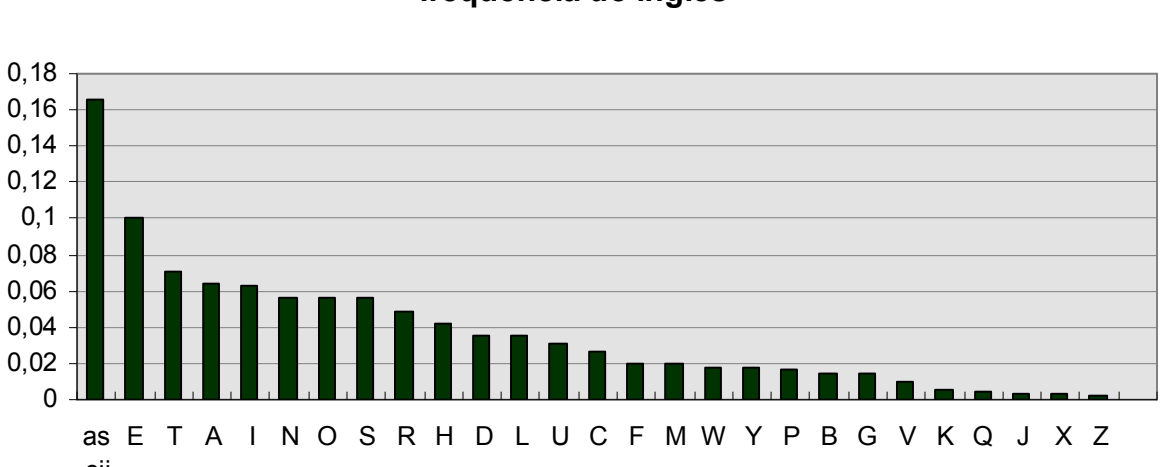
Exemplos de distribuição de frequência de letras



Média de frequências no **Português** por ordem alfabética



Média de frequências no **Português** por ordem decrescente

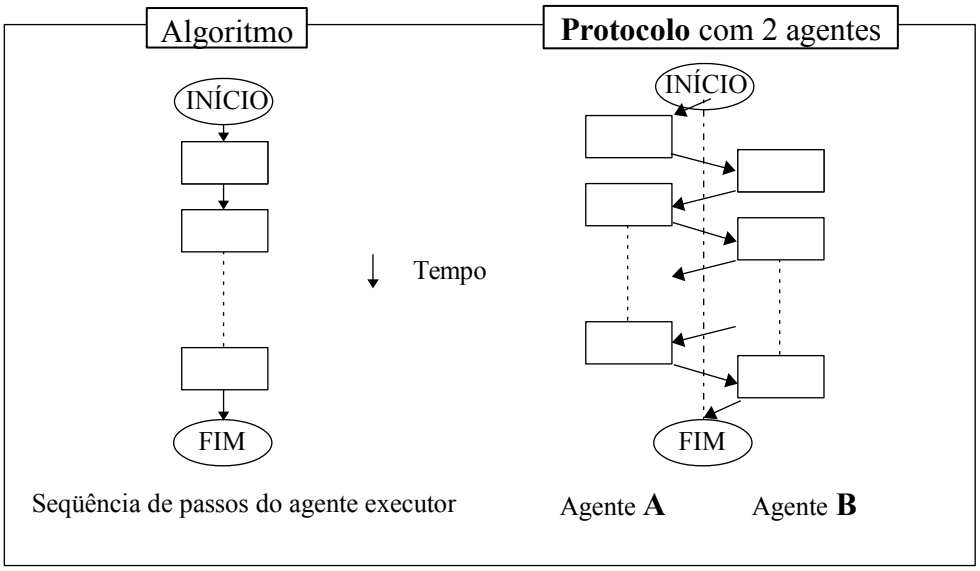


Média de frequências no **Inglês** por ordem decrescente

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 em três dimensões: Norma, Valor e Fato.
  - Com a informática, o que muda é a maior dificuldade em se estabelecer o Fato (em coerência com a definição Deleuziana de "virtual")
  - 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** (mensagem) de $\mathtt{m}$ (escrita que expressa **m**), ou **k**, **e** ou **d** (chaves) de $\mathtt{e}$ ou $\mathtt{e}^{-1}$ (funções de cifra correspondentes)

- **Protocolo Criptográfico -**

  Tem como propósito proteger algum interesse de certo(s) agente(s) em certas situações comunicativas nas quais esse interesse conflita com alguma característica da comunicação. Usa criptografia para manipular pontos, modos e momentos no processo comunicativo em que confiança é presumida ou necessária. Para isso, o protocolo criptográfico pode buscar atingir objetivos de um ou mais desses tipos:

  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*, a priori* desinteressado nesse propósito – o TTP (ver A-7), para superar impasses entre agentes *principais* (interessados no propósito)

  1 - Protocolo **arbitrado:**.....*usa 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** (também chamado ***trustless***)**:**……*sem TTP, supõe que a execução* ***exclui*** *possíveis vantagens para trapaças ou desavenças entre principais.*

Ação da Confiança em Protocolos Criptográficos

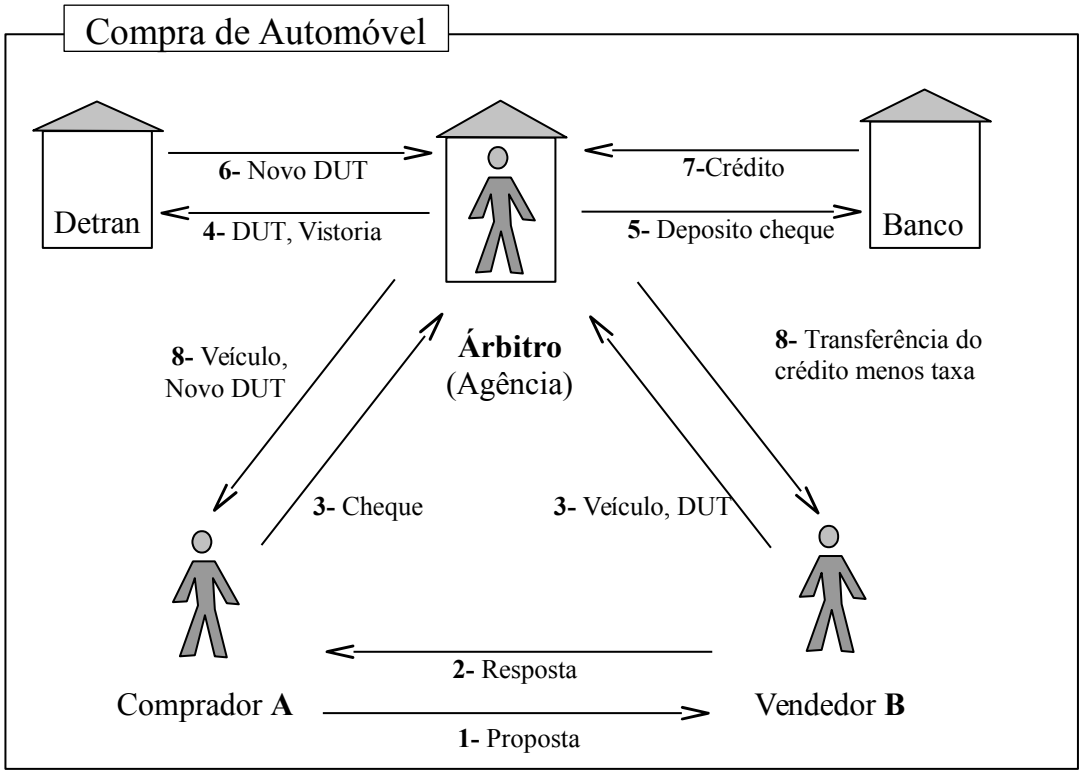
- **Resolução de impasses -**

  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 arbitrado (não computacional) -**



TTP (agência) se propõe a resolver impasses a partir de interesse supostamente "ortogonal" ao de seus clientes (o que quer vender e o que quer comprar carro)

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 falha única 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 limitações:

    1 - Natureza do problema:. *nem todo tipo de problema de security (premissas de confiança + objetivos de proteção) encontra solução em protocolos criptográficos autoverificáveis.*

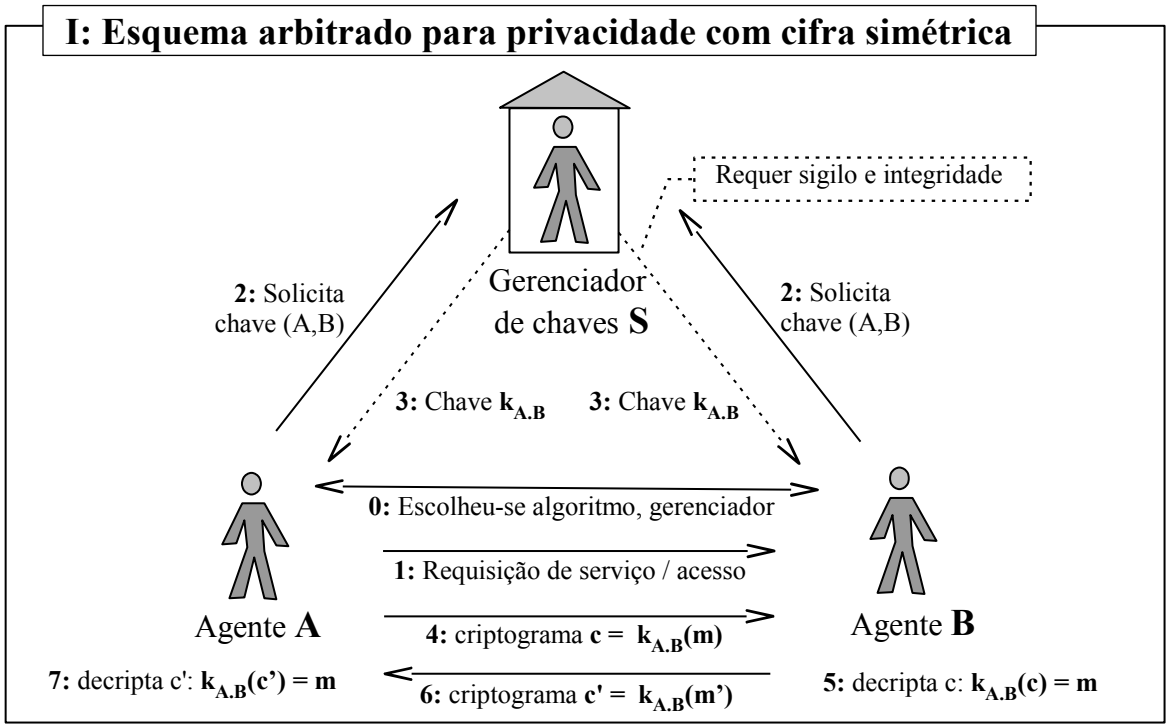
    2 - Análise exaustiva: *a autoverificação só é eficaz enquanto novas formas de ataque que subvertem o protocolo forem desconhecidas.*

- **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 de que se reconheceu algo ou alguém, e ter como convencer a outrem ou a outrens deste reconhecimento.* Subtipos:
  Subjetiva: *outrem: o identificado (interlocutor com quem se compartilha* **k***);*
  Objetiva: *outrens: o identificado e terceiros (que confiam na integridade de* **e***)*
- **Assinatura** (sentido jurídico): *Convencer-se, e a outros se necessário, 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 de protocolos criptográficos
Esquemas (subrotinas comuns)

- **Distribuição de chaves secretas em rede fechada -**



Premissas e limitações em I:

a - O esquema supõe que agentes principais (**A** e **B**) confiam um no outro, e no gerenciador de chaves, em relação ao sigilo das respectivas chaves.

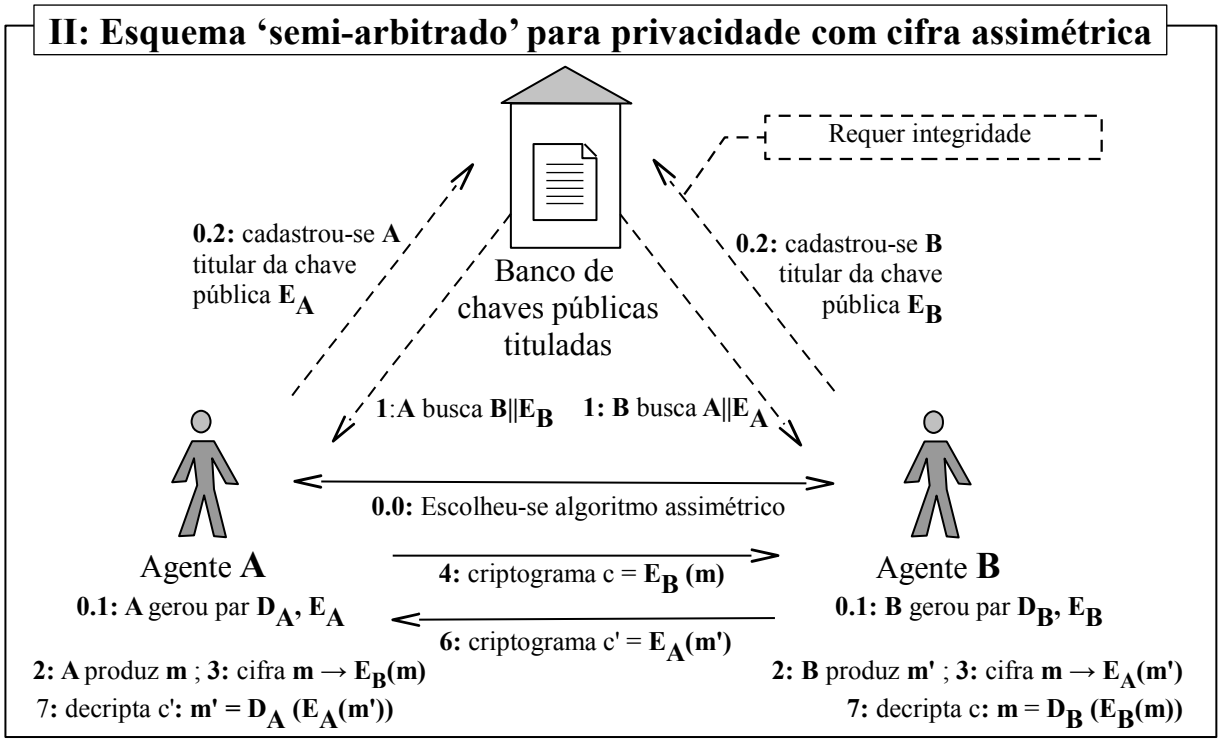
b - O esquema não provê para **A** a identificação positiva de B, nem vice-versa;

c - Sem prévia identificação mútua de A e B, ou se uma chave for comprometida, o atacante pode personificar A ou B e escalar ataques (escuta ativa, etc)

d - A privacidade em rede (= sigilo um-a-um) com cifra simétrica requer uma chave distinta para cada par de agentes. [n agentes $\Rightarrow$ n^2- n pares]; a distribuição de chaves neste esquema reduz (comparativamente a redes *peer-to-peer*) a demanda por Canais de Confiança necessários para distribuir chaves destinadas aos canais em banda [n canais de confiança ; n^2- n em banda].

e - As chaves $\mathbf{k_{A,B}}$ precisam ser geradas por um TTP (**S**) se o propósito do protocolo contendo I incluir controle de acesso aos recursos em rede (Cap. 3).

- **Distribuição de chaves públicas em rede fechada -**



Premissas e limitações em II:

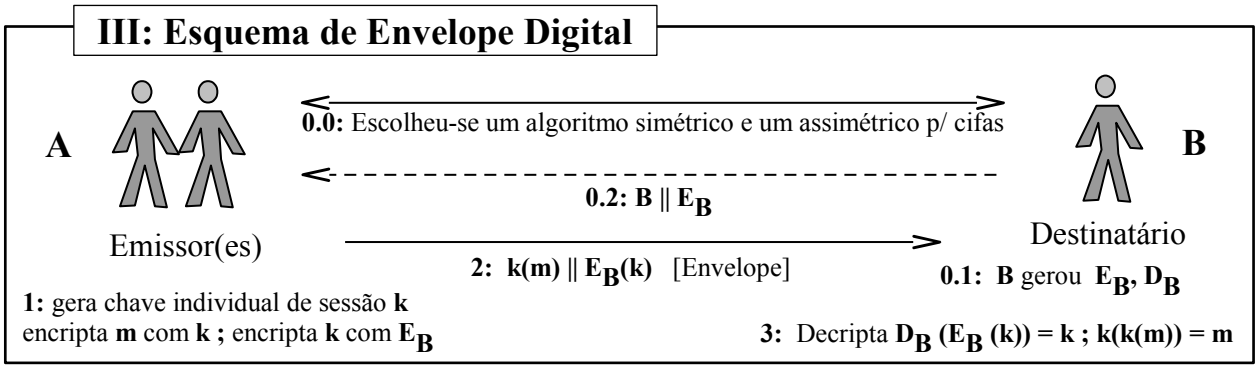
a - A origem e a integridade das relações identificadoras $\mathbf{A}||\mathbf{E_A}$, $\mathbf{B}||\mathbf{E_B}$ (de **A** e **B**) devem ser certificadas para armazenamento e distribuição das respectivas chaves públicas, se houver risco de ataques de *spoofing*, espelhamento, *replay.*

b - As chaves públicas cadastradas se expõem a ataques de tipo 4 (pag. 18).

c - Dentre os algoritmos criptográficos robustos em uso, os assimétricos são de 10^3 a 10^4 vezes mais lentos (computacionalmente + caros) que os simétricos.

- **Envelope Digital -**

Variante virtual com uso racional (II.c) **das 2 primitivas de cifragem**

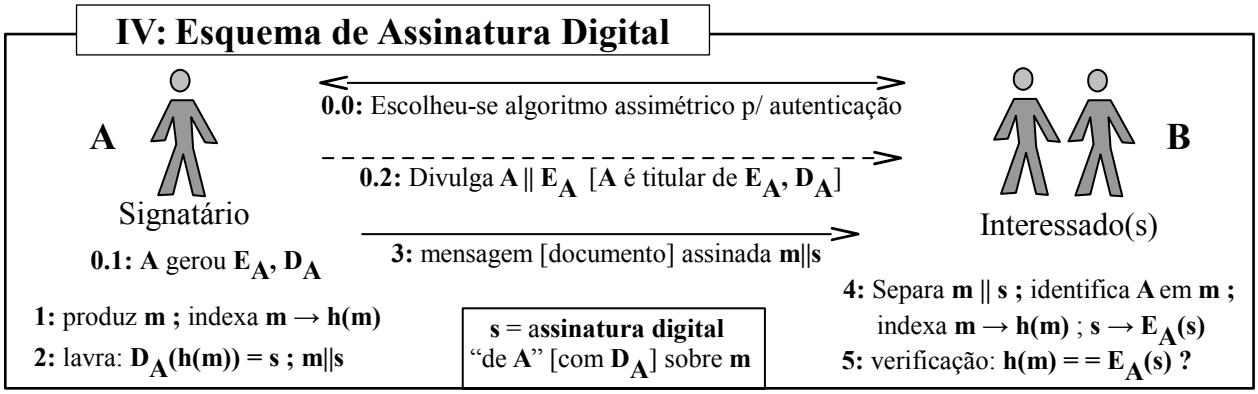


• Esquema de Assinatura Digital

Funções semiológicas tradicionais da assinatura manual (de punho)

- - **Inforjabilidade**: *confiança do verificador na identificação de autoria da assinatura em um documento, e na integridade deste desde então.*
- - **Inviolabilidade:** *confiança do verificador na integridade do conteúdo linguístico do documento, vinculado, pela assinatura, à vontade do autor.*
- - **Irrecuperabilidade:** *confiança de ambos na inviabilidade de reuso (transferência desse vínculo por "troca" de conteúdo ou de assinatura).*
- - **Irrefutabilidade:** *confiança do verificador na inviabilidade técnica de negação da autoria da assinatura pelo autor, perante ajuizamento.*

Autenticação objetiva virtualizada - funcionamento



Premissas de Confiança do esquema de Assinatura Digital:

a- Só o titular de um par de chaves controla o uso de sua chave privada;

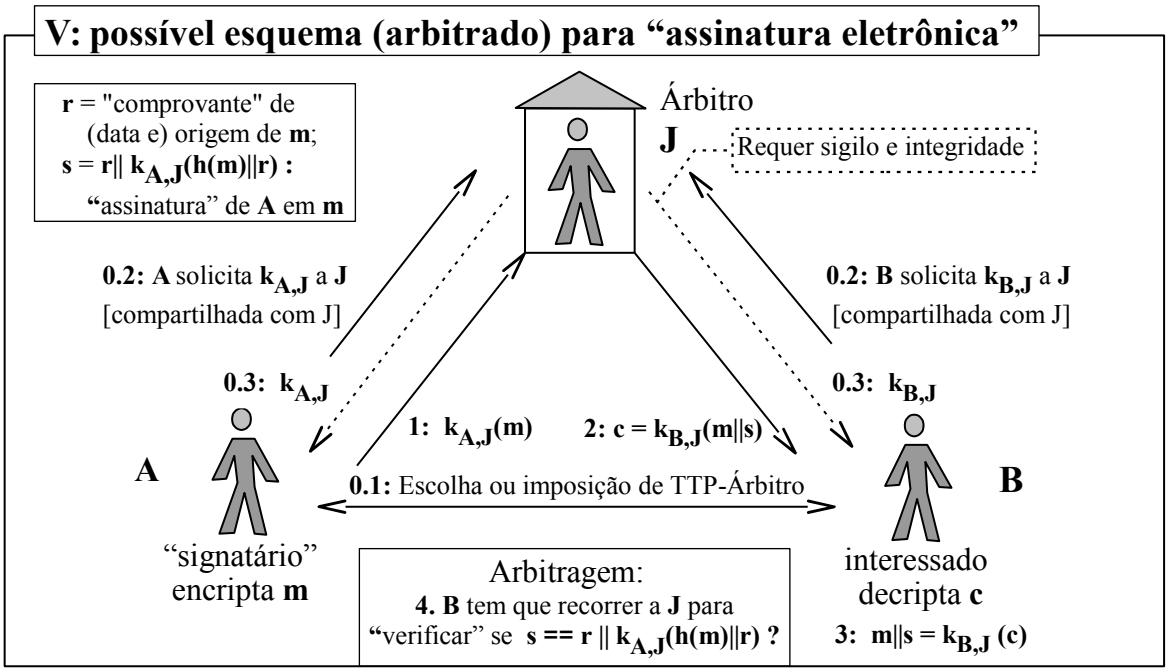
b- O usuário de uma chave pública confia na titularidade dessa chave;

c- Presume-se que o titular manifesta, ou pode manifestar, sua vontade no conteúdo linguístico de um documento assinado com sua chave privada (requer ambiente computacional sadio na execução dos respectivos passos)

Propósito e limitações do esquema de Assinatura Digital:

Autenticação objetiva – no Direito: 'oponível a terceiros' – para validar origem e integridade de mensagens ou documentos assinados, virtualizando as funções semiológicas da assinatura manual; com a irrefutabilidade, precariamente (IV.a,c)

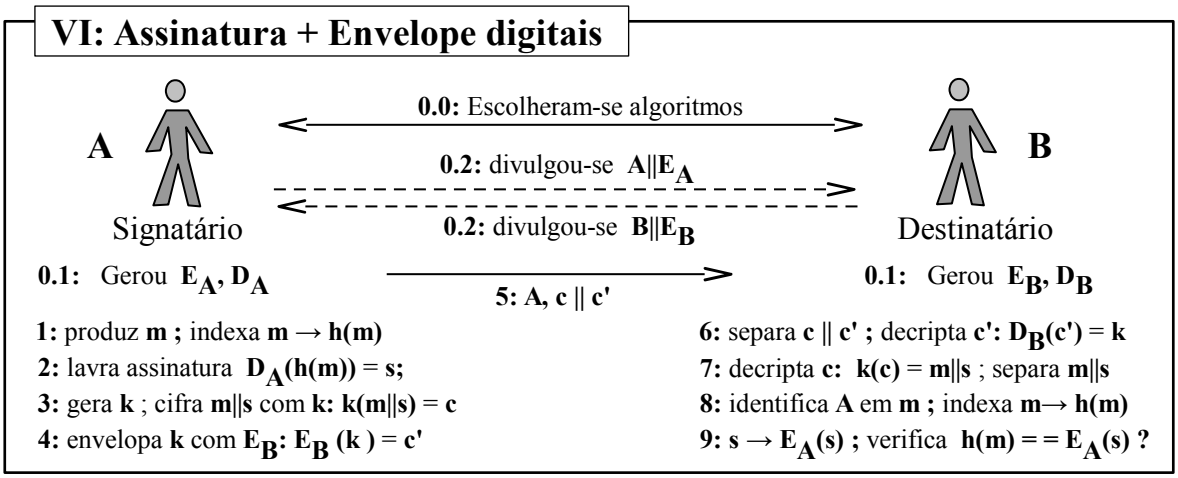
- **Autenticação arbitrada usando cifra simétrica**



Premissas de confiança e limitações em esquemas de "assinatura eletrônica"

Intermediador **J** opera como caixa-preta para verificação *indireta*, subjetiva (ver http://cic.unb.br/~rezende/trabs/laws.htm). Em situações onde a semântica de **s** é opaca para **B**, o esquema pode [no teatro] até ser 'vendido' (inclusive usando chaves assimétricas) como se fosse 'assinatura digital' com verificação intermediada, o que vem sendo feito por tribunal(is) no Brasil (ver www.ibadpp.com.br/1127).

- **Esquema para combinar as 2 primitivas de algoritmo assimétrico**



A ordem eficaz para encadear os 2 esquemas virtuais (com suporte invariante simbólico) é a mesma dos correspondentes esquemas tradicionais (com suporte físico).

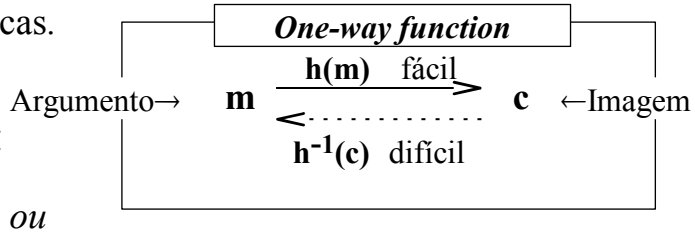
Codificação unidirecional e hash criptográfico

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

  Função simbólica com a qual é fácil (viável) calcular a imagem de um argumento, e difícil (inviável) deduzir um argumento para uma imagem dada (pré-imagem). Condição necessária – mas não suficiente – para hash criptográfico.

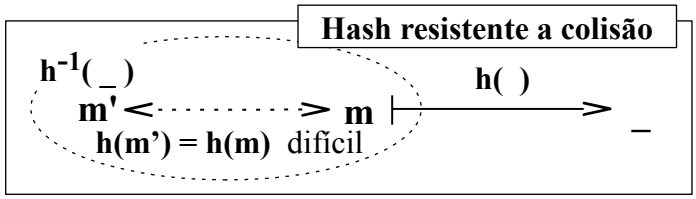
- **Função Unidirecional 'com Tranca'** (*Trapdoor function*):

  Função simbólica unidirecional para a qual existe um tipo de metadado sobre a imagem que torna fácil calcular uma pré-imagem sua. São fundamentais para geração de chaves assimétricas.



- **Hash criptográfico** [**h( )**]:

  - **Sinônimos**: *Digest, Fingerprint ou Checksum Criptográfico*; *Cheque de Integridade, Cifra unidirecional*, etc.
  - **Definição**: Função simbólica com argumentos de tamanho variável e imagens de tamanho fixo equiprováveis (*hash*), unidirecional (de custo para se obter pré-imagens estatisticamente controlado a inviável), e *resistente a colisões.*
  - **Resistência a colisões**: Quando é difícil (i.e., estatisticamente controlado a nível de custo inviável) obter-se *colisões livres*, i.e, encontrar dois argumentos que produzam a mesma imagem por **h( )** [ver "*ataque de aniversário"*]



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

  Esquema em que um hash criptográfico **h( )** é usado com 'chave' secreta $\mathbf{k_{A.B}}$ previamente compartilhada entre emissor e receptor, para autenticação de **m** entre ambos durante a transmissão de **m || h(m||$\mathbf{k_{A.B}}$)**. (Se um terceiro vier a conhecer $\mathbf{k_{A.B}}$, a origem e integridade de **m** não poderão mais ser validadas)

Autenticação: Resumo sobre uso adequado

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

  Função simbólica destinada a detectar erros (involuntários) de transmissão ou armazenagem de dados – para *safety*: Inócua em canais sujeitos a escuta ativa.

- **Hash** (ou *Checksum* ou *Digest* ou *Fingerprint*) **Criptográfico h( ):**

  Função simbólica com imagens de tamanho fixo equiprováveis (*hash*) unidirecional e resistente a colisões. Útil em esquemas de autenticação que empregam índices para representar mensagens em primitivas criptográficas:

  MAC: mensagem → $\mathbf{m \,||\, h(m||k_{A.B})}$ ← autenticador da mensagem

  Assinatura digital: mensagem → $\mathbf{m \,||\, D_A(h(m))}$ ← autenticador da mensagem

- **MAC:**

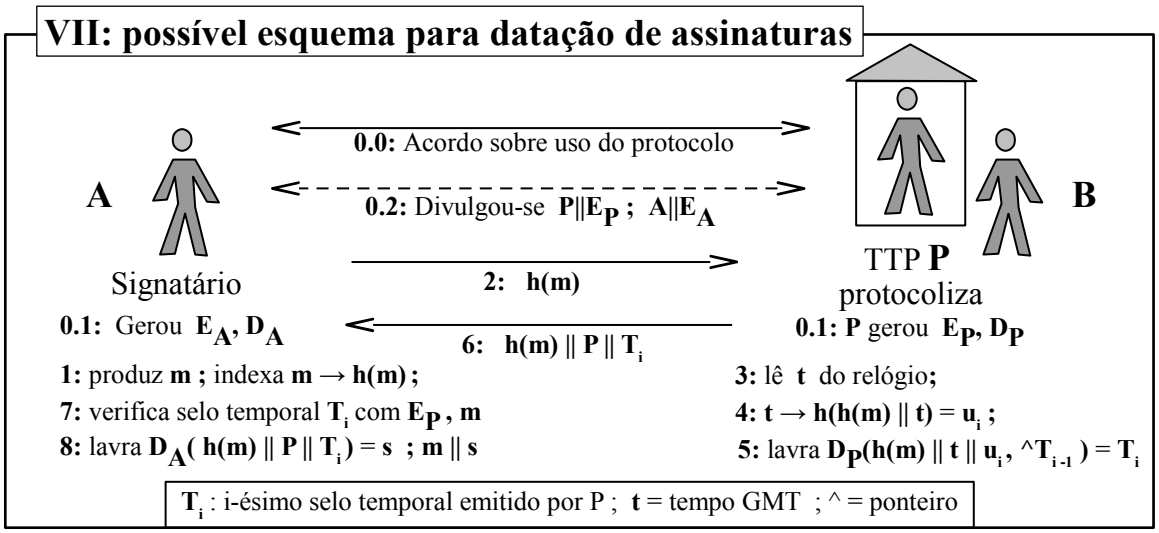
  Esquema para autenticação subjetiva, adequado a protocolos cujo propósito exija validação de origem e integridade (do conteúdo autenticado) *apenas* entre interlocutores, baseado em hash criptográfico e segredo previamente compartilhado entre os agentes principais **($k_{A.B}$)** nas situações abordadas. Por exemplo, em protocolos destinados a operar apenas em sistemas ou redes fechadas.

- **Assinatura digital:**

  Esquema para autenticação objetiva, adequado a protocolos cujo propósito exija validação de origem e integridade (do conteúdo autenticado) *também* por terceiros, além dos agentes principais. Adequado, portanto, onde tal validação não se baseie em compartilhamento de segredo, e sim em segredo não compartilhado (**D_A**) e criptografia assimétrica. Por exemplo, em protocolos para rede aberta, ou protocolos que devam produzir prova jurídica oponível a principais.

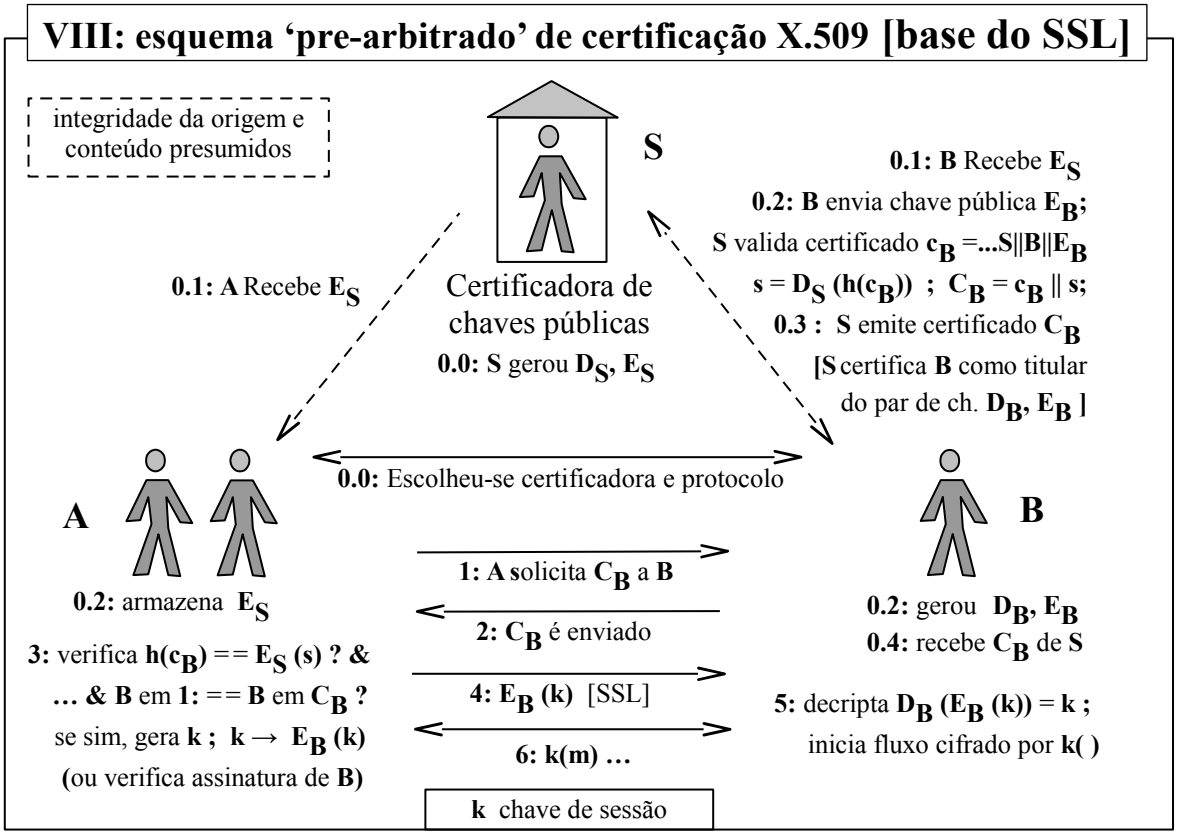
  OBS: a *irrefutabilidade* da assinatura digital é de natureza técnica: sustenta-se nas premissas de confiança do esquema e da sanidade da implementação. *Se presentes*, sustentam a *irretratabilidade jurídica* na autoria da assinatura.

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



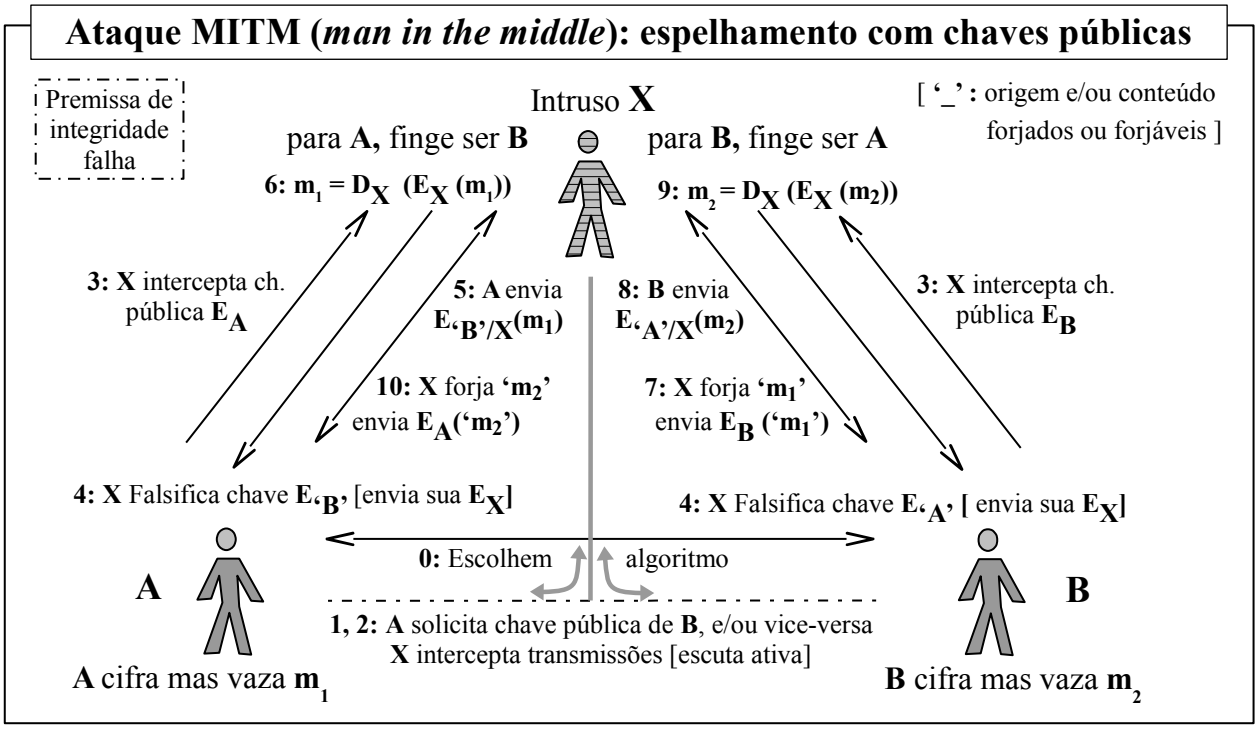
Encadeamentos semelhantes a VII.5 e VII.8 compõem esquemas em protocolos de *blockchain*, que formam a infraestrutura criptográfica de certas criptomoedas

- **Distribuição de chaves públicas tituladas em rede aberta**



O SSL *one way* requer certificado X.509 apenas do servidor **B** (o *two way*, também de clientes **A**); O SSL é gerenciado por aplicativo, e o TLS, pelo serviço TCP local

- **Limitações da autenticação em rede aberta**



Alguns problemas e riscos na distribuição de chaves públicas

a - A eficácia de protocolos que usam chaves públicas deriva da autenticidade destas, que, quando certificadas supõem uma cadeia de certificação **C:** ($\mathbf{C_B}$, $\mathbf{C_S}$ **...**)

b - Em VIII, para agir como MITM, **X** precisa antes falsificar a origem das chaves de **B** (e/ou **A**), ou fazer *spoofing* de **S** forjando $\mathbf{E_S}$. A eficácia da certificação que detectaria isso requer integridade em VIII.0.1 para a "última $\mathbf{E_S}$", *raiz* da cadeia **C**.

- **Certificado digital – o que é, para que serve?**

- Documento eletrônico emitido e assinado digitalmente por uma entidade certificadora, que serve para atestar uma relação identificadora, representada em seu conteúdo. No padrão X.509 esta relação é entre uma chave pública e uma identidade (nome): do titular (*subject*) desta chave – e da ch. privada correspondente

- Um certificado X.509 serve também para transportar a chave pública titulada junto a outros atributos dessa titulação (ex: uso pretendido para a chave). A verificação da assinatura em um certificado requer outro certificado: da entidade emissora do mesmo, os quais formam assim uma cadeia **C**: ($\mathbf{C_B}$, $\mathbf{C_{S_1}}$, $\mathbf{C_{S_2}}$, **...**).

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 sequências aleatórias na natureza é uma questão de cunho filosófico, 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 ou pseudoaleatórias** (Schneier)

  São sequências que exibem as propriedades conhecidamente mensuráveis de aleatoriedade. 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.

Subsequências de mesmo comptimento (bits)	*Probabilidade de ocorrência*
0, 1	1/2
00, 01, 10, 11	1/4
000,.., 111	1/8
0000,.., 1111	1/16
0..00,.., 1..11	$1/2^n$

  - **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 sequência não deve conter período (repetições de uma subsequência de tamanho próximo ao da sequência)

3: Protocolos Importantes

- **Necessidade de protocolos para a utilidade da Criptografia**

Criptografia não cria proteção do nada. Suas primitivas são capazes apenas de processar certos modos de confiança, por meio de mecanismos criptográficos que assim protegem certos valores, em situações onde tais modos ocorrem:

Autorização….*transforma presunção de sigilo e integridade externos (de senha, token ou chave) e integridade de canais internos, em identificação de um agente cadastrado, para aplicação de permissões de acesso*.

Cifragem.....*transforma presunção de sigilo e integridade de chave(s) em presunção do sigilo de mensagem cifrada, durante a transmissão desta.*

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

Certificação…*transforma presunção de sigilo e/ou integridade (de chaves assimétricas) em autenticação recursiva, para validação objetiva.*

Um protocolo criptográfico processa os modos de confiança que presume, se presentes em situações comunicativas que demandam proteção, fornecendo-a se seu propósito for adequado à demanda, e se sua implementação estiver sadia.

- **Protocolos "básicos" - Autenticação para controles fundamentais**

O desenho de um protocolo criptográfico básico presume condições mínimas de confiança "como entrada". Esse mínimo depende da arquitetura da plataforma onde se propõe buscar uma proteção básica. Tais protocolos podem então ser classificados conforme o tipo de arquitetura a que servem *adequadamente*:

Login: autenticação para controle de acessos *em sistema fechado;*

Distribuição de Chaves: autenticação p/ controle de seções *em rede fechada;*

PKI ou **ICP:** autenticação para controle de identidades *em rede aberta*.