Segurança e Privacidade

Sistemas informáticos em todo o lado e interligados: levantou grandes preocupações de segurança, levando investigadores e empresas a criar ferramentas, técnicas, padrões e regulamentos.

Ataque bem-sucedido:

- **Problemas de privacidade:** violação de dados confidenciais;
- **Segurança de dados internos:** danos no sistema, perdas financeiras, perdas de reputação;
- **Segurança de dados externos:** perigo para a vida das pessoas, danificar a natureza.

Fatores para se ter um nível aceitável de segurança num sistema informático:

- Arquitetura complexa dos sistemas;
- Tipo de sistema (crítico de segurança, crítico de negócio, crítico de segurança);
- Mecanismos de defesa implementados;
- Mecanismo de tolerância de falhas implementado;
- Interesses dos atacantes;

Defesa em profundidade: defender um sistema contra qualquer ataque específico usando **vários métodos independentes**.

Rede e internet segura

Rede e internet segura: medidas para **dissuadir**, **prevenir**, **detetar** e **corrigir** violações de segurança que envolvem a **transmissão de informações**.

Deterrence (Dissuasão):

- As estratégias de dissuasão procuram influenciar o comportamento do adversário, desencorajando-o de se envolver em atividades indesejadas;
- Pode ser alcançado influenciando a avaliação dos custos versus ganhos dos potenciais infratores (por exemplo, sanções pesadas).

Prevention (Prevenção):

- Usar técnicas, ferramentas e meios para prevenir um ataque (por exemplo: firewalls, criptografia).

Detection (Deteção):

- Usar ferramentas ou técnicas para monitorar e detetar um ataque ou intrusão (por exemplo, sistemas de deteção de intrusão).

Correction (Correção):

- Utilizar estratégias para mitigar um ataque e fazer correções (remoção de vulnerabilidades) no sistema para se tornar mais seguro.

Objetivos da segurança informática

Confidencialidade:

- **Confidencialidade de dados:** garante que **as informações não sejam disponibilizadas** ou divulgadas a **indivíduos não autorizados**.
- **Privacidade:** garante que os **indivíduos controlem** ou influenciem que informações relacionadas a si próprios podem ser recolhidas e armazenadas, e por quem e para quem estas podem ser divulgadas.

➢ A perda de confidencialidade significa divulgação não autorizada de informações.

Ferramentas:

- Criptografia;
- Access Control;
- Autenticação;
- Autorização;
- Segurança física;
- Anonimato.

Integridade:

- **Integridade de dados:** assegura que os **dados são alterados** apenas de uma **forma autorizada** e especificada.
- **Integridade do sistema:** Assegura que um **sistema desempenha a sua função prevista de forma irrepreensível**, sem manipulação deliberada e não autorizada do sistema.

- Uma perda de integridade significa modificação ou destruição não autorizada de dados ou sistema.

Ferramentas:

- Backups;
- Checksums;
- Códigos de correção de dados.

Disponibilidade:

- Assegura que os **sistemas funcionam prontamente** e que o **serviço não é recusado** a utilizadores autorizados.

- A perda do serviço traduz-se num grande prejuízo financeiro e na perda de clientes.

Ferramentas:

- Proteções físicas;
- Redundâncias computacionais.

Responsabilidade (não-repúdio):

- Rastreia uma violação de segurança até uma parte responsável.
- Os sistemas devem manter os registos das suas atividades para ser possível rastrear violações de segurança.

Ferramentas:

- Registo;
- Assinaturas digitais.

Autenticidade:

- Verificar se os utilizadores são quem dizem ser.

Ferramentas:

- Autenticação de dois fatores (2FA);
- Autenticação multifator (MFA).

Ameaças e ataques

Ameaça: uma **potencial de violação da segurança**, que existe quando há uma circunstância, capacidade, ação ou evento que possa violar a segurança e causar danos. Ou seja, uma ameaça é um **possível perigo** que pode explorar uma vulnerabilidade.

Tipos de ameaças:

- Ameaças naturais (por exemplo: inundações);
- Ameaças não intencionais (por exemplo: um funcionário obtém por engano acesso a informações privadas Informação);
- Ameaças intencionais (por exemplo: spyware, malware, funcionários insatisfeitos, utilizadores mal-intencionados).

Ataque: um ataque à segurança do sistema. Um **ato inteligente** para **fugir dos serviços de segurança** e **violar a política de segurança** de um sistema.

Eavesdropping (Espionagem): intercetação de informação destinada a outrem durante a sua transmissão **através de um canal de comunicação**.

Alteration (Alteração): modificação não autorizada de informações.

- Exemplo: o ataque man-in-the-middle, onde um fluxo de rede é intercetado, modificado e retransmitido.

Denial-of-service (Negação de serviço): a interrupção ou degradação de um serviço ou de dados acesso.

- Exemplo: spam de e-mail, na medida em que se destina simplesmente a preencher uma fila de e-mail e desacelerar um servidor de e-mail.

Masquerading (Mascara): fabrico de informações que supostamente vêm de alguém que não é realmente o autor.

Repudiation (Repúdio): negação de um compromisso ou de uma receção de dados

- Trata-se de uma tentativa de voltar atrás num contrato ou num protocolo que exige que as diferentes partes forneçam recibos que confirmem a receção de dados.

- Normalmente, acontece quando um sistema não adota controlos adequados para rastrear e registar corretamente as ações dos utilizadores, tornando assim possível o repúdio.
- Quando o repúdio é possível, os utilizadores podem manipular os dados sem serem conhecidos.

Correlation and traceback (correlação e rastreio): integração de múltiplas fontes de dados e fluxos de informação para determinar a origem de um determinado fluxo de dados ou informação.

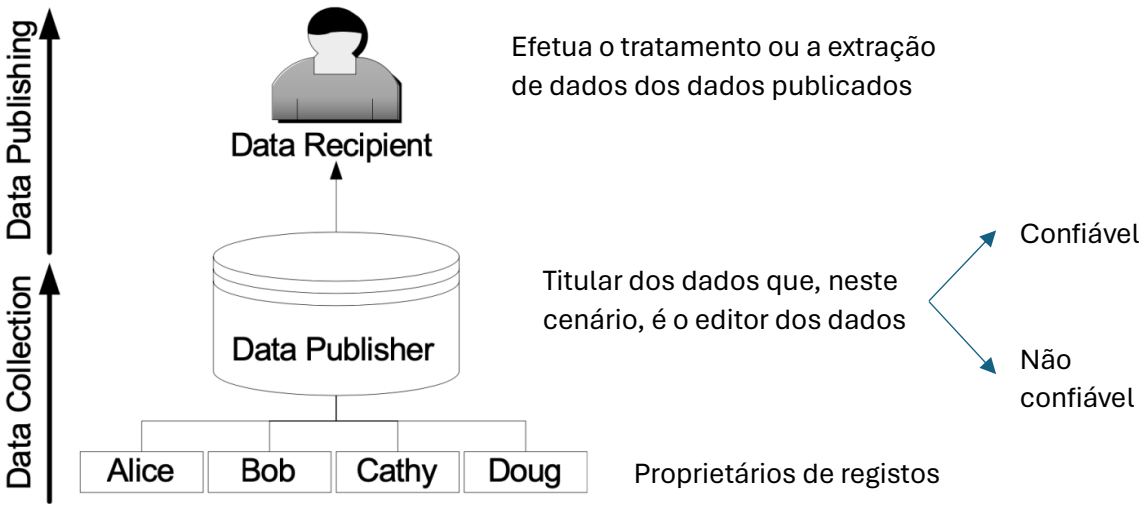
Análise ou extração de dados

Data mining: processo de **extrair informações úteis, interessantes e até então desconhecidas** de grandes conjuntos de dados.

Sucesso depende de:

- Disponibilidade de alta qualidade de dados;
- Partilha eficaz de dados.

Cenário de recolha e divulgação de dados



1. **Fase de recolha de dados:** o titular dos dados recolhe os dados dos proprietários dos registos.

2. **Fase de divulgação de dados:** o titular dos dados divulga os dados coletados a um recetor.

Tipos de titulares de dados:

- **Confiável:** utiliza diferentes técnicas para recolher registos anónimos dos seus proprietários.
- **Não confiável:** pode tentar identificar informações sensíveis dos proprietários dos registos.

Privacidade na divulgação de dados

Objetivo: divulgar recolhas de dados publicamente ou a terceiros para análise **sem divulgar a propriedade dos dados confidenciais**.

Privacy-preserving data publishing (PPDP) – terminologias:

- **Identifier:** atributos com informações que identificam explicitamente/ exclusivamente o proprietário do registo (ex.: nome, número de telefone, endereço e número do seguro social).
- **Quasi-identifier (QID):** atributo que não identifica explicitamente um utilizador, mas pode ser combinado com dados de outras fontes públicas para desanonimizar o proprietário de um registo (ex.**:** ZIP code de 5 dígitos, data de nascimento e sexo).
- **Sensitive attribute:** atributos privados/sensíveis específicos do indivíduo que não devem ser divulgados publicamente (ex.: doenças em registos médicos, salário, situação de incapacidade).
- **Non-Sensitive attribute:** todos os atributos que não se enquadram as três categorias anteriores.

PPDP – Anonimização:

Anonimização: abordagem PPDP que busca ocultar a identidade e/ou os dados confidenciais dos proprietários de registos.

1. **Primeira ideia:** remover informações de identificação pessoal / identificadores explícitos (uma forma de anonimização). Contudo **não é suficiente!**

Ataque de ligação

O **ataque de ligação** ocorre quando um adversário é capaz de ligar o proprietário de um registo a:

- um registo numa tabela de dados publicada → **ligação de registos**;
- um atributo sensível numa tabela de dados publicada → **ligação de atributo**;

- a própria tabela de dados publicada → **ligação de tabelas**.

Nos 3 tipos de ligações, assume-se que o **adversário conhece o QID da vítima**

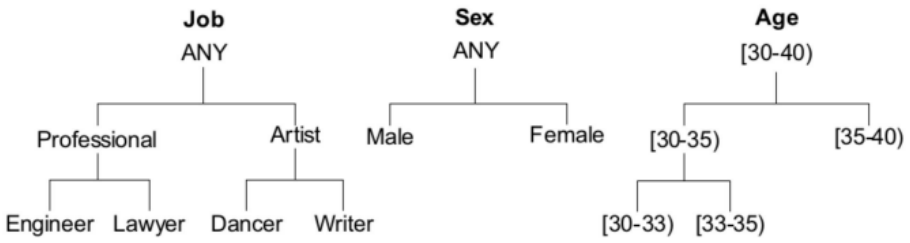
Na **ligação de registos e de atributos** o adversário sabe que o **registo da vítima está na tabela divulgada** e procura identificar o registo da vítima e/ou informações confidenciais na tabela.

Na **ligação de tabelas** o ataque procura determinar **a presença ou ausência do registo da vítima na tabela divulgada**.

Operações de anonimato

Generalização: Substituição de um valor por um mais geral

- Para um **atributo categórico**, um valor específico pode ser substituído por um **valor geral** de acordo com uma determinada taxonomia.
- Para um **atributo numérico**, os valores exatos podem ser substituídos por um **intervalo** que cobre valores exatos.



Supressão: Remoção de alguns valores de atributos ou registos

- **Supressão de registos (por linha):** supressão/eliminação de um registo inteiro (uma linha).
- **Supressão de valores (em colunas):** supressão/eliminação de todos os valores de um atributo numa tabela (uma coluna).
- **Supressão de células ou supressão local:** supressão/remoção de algumas instâncias de um determinado valor numa tabela.

Anatomização: Desassocia QIDs e atributos sensíveis

- Não modifica o QID ou o atributo sensível, mas desassocia os dois.
- Como é que funciona:

1. Os dados sobre o QID e os dados sobre o atributo sensível são lançados em duas tabelas separadas:
    - Uma tabela de quase-identificadores (QIT) contém os atributos QID;
    - Uma tabela sensível (ST) contém os atributos sensíveis.

Dados Originais

Age	Sex	Disease (sensitive)
30	Male	Hepatitis
30	Male	Hepatitis
30	Male	HIV
32	Male	Hepatitis
32	Male	HIV
32	Male	HIV
36	Female	Flu
38	Female	Flu
38	Female	Heart
38	Female	Heart

QID Atributo Sensível

Tabela QIT

Age	Sex	GroupID
30	Male	1
30	Male	1
30	Male	1
32	Male	1
32	Male	1
32	Male	1
36	Female	2
38	Female	2
38	Female	2
38	Female	2

Tabela ST

GroupID	Disease (sensitive)	Count
1	Hepatitis	3
1	HIV	3
2	Flu	2
2	Heart	2

2. Tanto o QIT como o ST terão um atributo comum, o GroupID.

Vantagem:

- Os dados do QIT e do ST não são alterados.

Desvantagem:

- Aumento do número de tabelas;
- Com os dados publicados em duas tabelas, não é claro como as ferramentas normais de extração de dados podem ser aplicadas aos dados publicados, sendo necessário conceber novas ferramentas e algoritmos.

Permutação-Perturbação: Substituição de dados originais por valores sintéticos

- **Adição de Ruido:** substituir o valor sensível original s por s + r, em que r é um **valor aleatório** retirado de uma qualquer distribuição.

  Frequentemente utilizado para ocultar dados numéricos sensíveis (por exemplo: salário).

  A privacidade é medida determinando a proximidade com que os valores originais de um atributo modificado podem ser estimados.

- **Troca de dados:** troca de valores de atributos sensíveis entre registos individuais, devendo as trocas manter as contagens de frequência para análise estatística.

  A troca de ordem pode preservar melhor a informação estatística do que a troca aleatória de dados.

- **Geração de dados sintéticos:** gerar dados sintéticos que retenham informações estatísticas úteis.

- **Perturbação aleatória**
  Desvantagem: os registos publicados são "sintéticos", pelo que não correspondem às entidades do mundo real representadas pelos dados originais; portanto, os registos individuais nos dados perturbados são basicamente desprovidos de significado para os destinatários humanos.

A operação de **generalização** ou **supressão** esconde alguns **pormenores no QID.**

A **anatomização** e a **permutação** desassociam a correlação entre o QID e os atributos sensíveis, **agrupando ou baralhando os valores sensíveis num grupo de QID**.

Modelos de privacidade

Modelos de privacidade: usados para garantia e medição da privacidade.

Modelos sintáticos (k-anonymity, l-diversity, t-closeness, …):

- Especificam as condições sintáticas para a libertação de dados;
- Pressupostos fortes sobre vetores de ataque.

Modelos semânticos (privacidade diferencial):

- Utilizam informações sobre as características dos próprios dados para adicionar seletivamente ruído ao resultado;
- Muito menos suposições sobre os atacantes.

K-anonymity

Objetivo: evitar vinculação de registos por meio de **QID.**

Classe de equivalência: conjunto de **k registos** que têm o **mesmo QID.**

Passos:

1. **Generalização:** substituir os quasi-identifiers por valores menos específicos, mas semanticamente consistentes, até obter **k valores idênticos**
2. **Supressão:**
   - Omitir um registo inteiro - supressão de linha (comum com “outliers” que são difíceis de anonimizar);
   - Omitir um atributo de todos os indivíduos - supressão de coluna (comum com identifiers).

Ataques ao K-anonymity

O K-anonymity não fornece privacidade se:

- Os valores sensíveis numa classe de equivalência não têm diversidade;
- O atacante tem conhecimentos prévios.

Desvantagem: não tem em conta se os valores dos atributos sensíveis em cada classe de equivalência são distintos.

ℓ-Diversity

Ideia: os **atributos sensíveis** devem ser "**diversos**" dentro de cada classe de equivalência.

Princípio: cada classe de equivalência tem **pelo menos ℓ valores bem representados.**

Desvantagens:

- Não considera a distribuição global de valores sensíveis;
- Não considera a semântica de valores sensíveis.

T-Closeness

A **distribuição** dos valores sensíveis em cada classe de equivalência deve ser "**próxima**" da distribuição correspondente na tabela original.

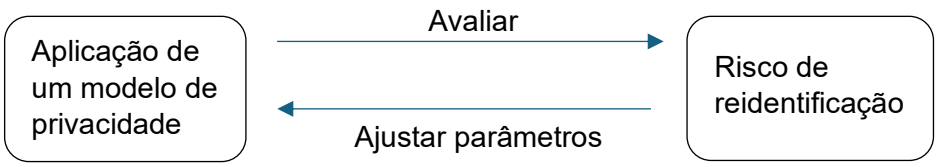
"**Próxima**" significa limitado superiormente por um limite t.

Desidentificação e reidentificação

Desidentificação: processo de remoção ou de mascarar de informações de identificação pessoas de um conjunto de dados (exemplo: nomes, moradas, datas de nascimento, números de segurança social).

Objetivos da desidentificação: **proteger a privacidade** dos indivíduos e **reduzir o risco de reidentificação**, permitindo ainda que os dados sejam utilizados para investigação ou para outros fins.

Processo de desidentificação:



Reidentificação: processo de **correspondência entre dados desidentificados com fontes de informação externas**, a fim de reidentificar indivíduos cujas identidades deveriam ser protegidas pelo processo de desidentificação.

Risco de reidentificação:

- Probabilidade dos dados desidentificados poderem ser reidentificados e associados a um indivíduo ou grupo de indivíduos;
- Surge quando ainda existe informação suficiente no conjunto de dados desidentificados para que alguém possa reidentificar os indivíduos.

Para **calcular o risco de reidentificação**, é necessário determinar a Distinção e a Separação dos QIDs.

Distinção e separação

Distinção: grau em que as variáveis tornam os registos distintos

$$Proporção\ de\ distinção = \frac{\#\text{valores distintos}}{\#\text{todos os valores}}$$

- 1 - proporção de distinção = percentagem de dados a remover para esse QID se tornar uma chave de identificação
- Valores de porporção de distinção mais altos indica QIDs mais prováveis

QID = (age)
Set of all values = {**20**,**30**,**40**,**20**,**40**}, size = **5**
Set of distinct values = {**20**, **30**, **40**}, size = **3**
Distinction ratio = #distinct values / #all values
= **3/5**
= **0.6**

	age	sex	state
1	20	Female	CA
2	30	Female	CA
3	40	Female	TX
4	20	Male	NY
5	40	Male	CA

Separação: grau em que as combinações de variáveis separam os registos

$$Proporção\ de\ separação = \frac{\#\text{tuplos distintos}}{\#\text{todos os tuplos}}$$

QID = (age)

Set of all tuples =

{**(20,30)**,**(20,40)**,**(20,20)**,**(20,40)**,**(30,40)**, **(30,20)**, **(30,40)**, **(40,20)**, **(40,40)**, **(20,40)**}, size = **10**

Set of separated tuples = {**(20,30)**,**(20,40)**,**(20,40)**,**(30,40)**, **(30,20)**, **(30,40)**, **(40,20)**, **(20,40)**}, size = **8**

(20,20) and (40,40) are removed from the first set

Separation ratio = #separated tuples / #all tuples

= **8/10**

= **0.8**

Modelos de ataque

Modelo Prosecutor:

- Tem como alvo **um indivíduo específico**;
- O atacante sabe se o indivíduo-alvo está no conjunto de dados.

Risco de reidentificação no modelo Prosecutor:

$$Menor\ P_{Prosecutor} = \frac{1}{\#\text{maior classe de equivalência}}$$

$$Maior\ P_{Prosecutor} = \frac{1}{\#\text{menor classe de equivalência}}$$

$$Médio\ P_{Prosecutor} = \frac{\#combinações\ diferentes\ de\ QIDs}{\#\text{nº de linhas}}$$

N = 11, **QID** = (**Gender**, **Year of Birth**)

Equivalence classes (**K**)

1. (Male, 1970 - 1979) F_1 = 3, $P_{prosecutor}$ = **1/3 ≈ 0.33**
2. (Male, 1980 - 1989) F_2 = 2, $P_{prosecutor}$ = **1/2 ≈ 0.5**
3. (Female, 1990 - 1999) F_3 = 2, $P_{prosecutor}$ = **1/2 ≈ 0.5**
4. (Female, 1980 - 1989) F_4 = 2, $P_{prosecutor}$ = **1/2 ≈ 0.5**
5. (Male, 1990 - 1999) F_5 = 2, $P_{prosecutor}$ = **1/2 ≈ 0.5**

- Lowest $P_{Prosecutor}$ = **1/3 ≈ 0.33**

For QID_1 = (Male, 1970 - 1979), F_1 = 3

- Highest $P_{Prosecutor}$ = **1/2 = 0.50**

For QID_2 = (Male, 1982), F_2 = 2

Average $P_{Prosecutor}$ = size(K)/ N = **5/11 ≈ 0.45**

OUASI-IDENTIFIERS		
Gender	**Decade of Birth**	**DIN**
Male	1970-1979	2046059
Male	1980-1989	716839
Male	1970-1979	2241497
Female	1990-1999	2046059
Female	**1980-1989**	**392537**
Male	1990-1999	363766
Male	1990-1999	544981
Female	**1980-1989**	**293512**
Male	1970-1979	544981
Female	1990-1999	596612
Male	1980-1989	725765

Disclosed File

Conhecendo o ano de aniversário e o género: QID = (Female, 1987)

$F_{(Female, 1987)}$ = **2**

$P_{Prosecutor}$ = **1/2 = 0.5**

Modelo Jornalist:

- Tem como alvo qualquer indivíduo;
- O atacante seleciona um alvo aleatoriamente porque a reidentificação de qualquer registo permite atingir o objetivo;
- Atacante inteligente: foco em classes de equivalência menores (maior probabilidade de reidentificação).

Risco de reidentificação no modelo Jornalist:

$$P_{Jornalist} = \frac{1}{\#\text{menor classe de equivalência}} = Maior\ P_{Prosecutor}$$

Modelo Marketer:

- Tem como alvo o maior número possível de indivíduos
- Um ataque é considerado bem-sucedido se uma grande parte dos registos puder ser reidentificada

Risco de reindentificação no modelo Marketer:

$$P_{Marketer} = \frac{\#combinações\ diferentes\ de\ QIDs}{\#\text{nº de linhas}} = Médio\ P_{Prosecutor}$$

Formas do atacante saber que o alvo está no conjunto de dados:

- O conjunto de dados representa **toda a população**
- O conjunto de dados é uma **amostra conhecida** de uma população (ex.: adolescentes)
- Os participantes **revelarem** que fazem parte de amostra

Utilidade de dados vs privacidade

A anonimização dos dados causa perdas de informação, o que pode comprometer a utilidade dos dados.

Objetivo: obter a máxima privacidade e a máxima privacidade (situação ideal), contudo é um objetivo impossível de atingir

Objetivo real: perda mínima de dados (utilidade aceitável – dados úteis para análise) e um risco muito pequeno de reidentificação.

Métricas de utilidade:

- Precisão / distorção mínima
- Perda de informação

- Descritibilidade
- Tamanho médio da classe de equivalência

Computação multipartidária

Computação distribuída:

- Considera o cenário em que vários dispositivos informáticos (ou partes) distintos, porém conectados, pretendem efetuar um cálculo conjunto de alguma função.
- Lida classicamente com questões de computação sob a ameaça de avarias de máquinas e/ou falhas inadvertidas.

Computação multipartidária segura: preocupa-se com a possibilidade de um **comportamento malicioso** por parte de uma entidade adversária.

Objetivo: permitir que as partes realizem essas tarefas de computação distribuída de forma **segura**.

Na computação multipartidária segura:

- Parte-se do princípio de que a execução de um protocolo pode ser "atacada" por uma entidade externa, ou mesmo por um subconjunto das partes participantes;
- O objetivo deste ataque pode ser obter informações privadas ou fazer com que o resultado do cálculo seja incorreto.

Requisitos importantes em qualquer protocolo de computação seguro:

- **Privacidade:** nenhuma parte deve saber **nada para além do resultado prescrito** (ex.: o licitante mais alto no caso de um leilão).

- **Correção:** cada parte tem a garantia de que o resultado que recebe está **correto**. (ex.: a parte com a licitação mais elevada tem a garantia de ganhar e nenhuma parte, incluindo o leiloeiro, pode influenciar este facto).

Mais requisitos:

- **Independência das entradas:** as partes corrompidas devem escolher os seus inputs independentemente dos inputs das partes honestas. (ex.: as licitações são mantidas em segredo e as partes devem fixar as suas licitações independentemente das outras).

- **Entrega garantida de resultados:** as partes corrompidas não devem ser capazes de impedir que as partes honestas recebam os seus resultados. Por

outras palavras, o adversário não deve ser capaz de perturbar a computação através de um ataque de "**negação de serviço**".

- **Equidade:** As partes corrompidas devem receber os seus resultados se e só se as partes honestas também receberem os seus resultados.

Abordagem heurística para segurança:

1. Construir um protocolo;
2. Tentar quebrar o protocolo;
3. Consertar o intervalo;
4. Voltar para 2.

➢ Esta abordagem não funciona para a privacidade de um indivíduo, pois uma vez que esta é violada não há como voltar atrás.

Perigos da Abordagem Heurística:

- Os hackers farão tudo para explorar uma fraqueza - se existir uma, será facilmente descoberta
- Os verdadeiros adversários não lhe dirão que violaram o protocolo
- Nunca se pode ter a certeza de que o protocolo é seguro
- A segurança não pode ser verificada empiricamente

Estratégias de corrupção:

- **Modelo de corrupção estático:**
  - O adversário recebe um conjunto fixo de partes que controla;
  - As partes honestas permanecem honestas durante todo o processo, enquanto as partes corrompidas permanecem corrompidos.

- **Modelo de corrupção adaptativo:**
  - Em vez de ter um conjunto fixo de partes corrompidos, os adversários adaptativos têm a capacidade de corromper partes durante a computação;
  - A escolha de quem corromper e quando pode ser decidida arbitrariamente pelo adversário ou pode depender da sua visão da execução;
  - Quando uma parte é corrompida, permanece corrompida a partir desse momento.

- **Modelo proativo de corrupção:**
  - Considera a possibilidade das partes serem corrompidas apenas durante um determinado período de tempo;
  - As partes honestas podem ser corrompidas ao longo da computação, mas as partes corrompidas também podem tornar-se honestas.

Comportamento dos adversários:

- **Adversários semi-honestos:**
  - O adversário **obtém o estado interno** de todas as partes corrompidas e tenta usá-lo para obter informações que devem permanecer privadas;
  - As partes corrompidas seguem corretamente a especificação do protocolo;
  - São também designados por "**honestos-mas-curiosos**" e "**passivos**".

- **Adversários maliciosos:**
  - As partes corrompidas desviam-se da especificação do protocolo;
  - Os adversários maliciosos são também designados por "**ativos**".

Transferência inconsciente (OT)

Um remetente transfere uma das muitas potenciais informações para um recetor, mas permanece alheio à informação que foi transferida (se é que foi transferida alguma). Isto é, partilha informações sem saber quais é que foram partilhadas.

Prova de segurança OT:

- A visão do remetente consiste apenas em duas chaves públicas pk0 e pk1. Portanto, não aprende nada sobre esse valor de α;
- O recetor só conhece uma chave secreta e por isso só pode saber **uma mensagem**;
- Nota: isto pressupõe um comportamento **semi-honesto**. Um recetor malicioso pode escolher duas chaves juntamente com as suas chaves secretas.

Compromisso de bits (BC)

Incorpora 2 fases:

1. **Fase de compromisso:** o emissor compromete-se enviando um token ao recetor.
   - O emissor tem um bit α;
   - O emissor envia ao recetor uma cadeia de compromisso *c* (compromisso com o bit α)
2. **Fase de revelação:** o emissor revela o bit.
   - O emissor envia uma mensagem de descomprometimento ao recetor.
   - O recetor utiliza a mensagem de anulação e *c* para obter α.

Conhecimento zero (ZK)

Conhecimento nulo: o verificador não aprenderá nada para além do facto de que a afirmação estar correta.

Solidez: o provador não será capaz de convencer o verificador de uma afirmação incorreta.

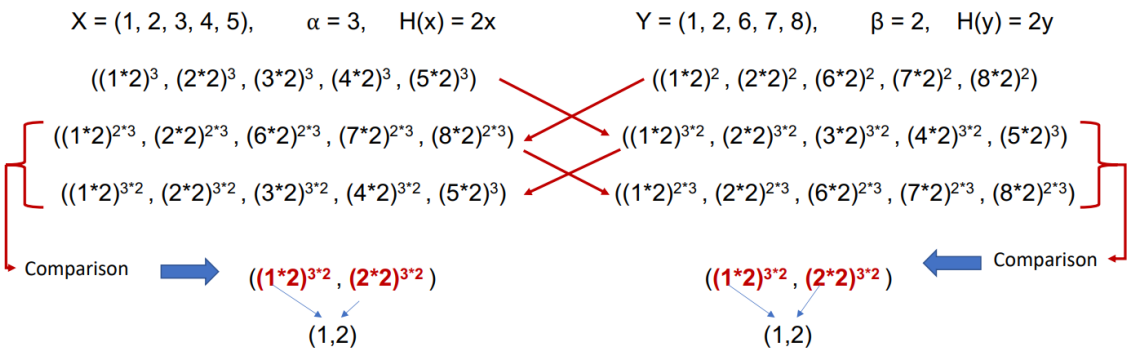
Interseção de conjuntos privados (PSI)

Protocolo Naive hashing (soluções de hash ingénuas)

'A' compara os seus hashes com os enviados por 'B' e reenvia a interseção entre estes, mais especificamente reenvia os dados que apresentam valores em comum em 'A' e em 'B'.

Problema: não protege a privacidade de B se as entradas não tiverem uma entropia considerável

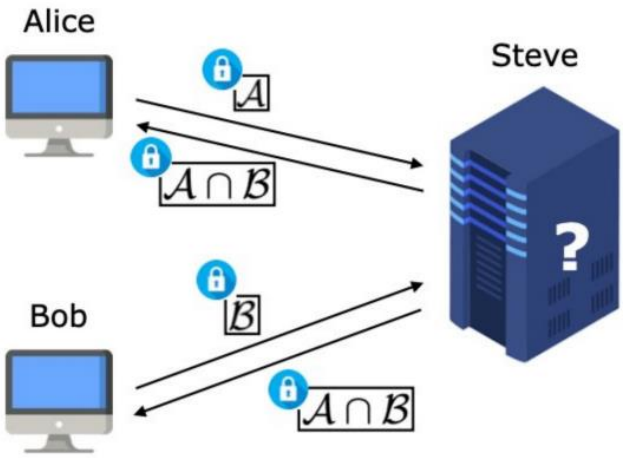
Protocolo Diffie-Hellman-based



Limitações:

- Este protocolo pressupõe um elevado grau de confiança entre 'A' e 'B';
- 'A' pode falsificar uma correspondência com 'B', enviando-lhe de volta a mensagem que recebeu na terceira etapa do protocolo;
- 'B' (ou 'A') também pode descobrir se tem uma correspondência sem a revelar a 'A', enviando lixo no último passo do protocolo;
- Não protege a privacidade de 'A' e 'B' se as entradas não tiverem uma entropia considerável (semelhante à solução anterior).

Protocolo auxiliado por servidor



- Há um grupo adicional, chamado Steve;
- Alice e Bob confiam em Steve.

Requisito de privacidade: Steve **não deve obter informações** sobre os itens de Alice e Bob.

Protocolo privado:

- A única informação que passa para Steve é o número de itens que a Alice e o Bob têm originalmente e o tamanho da intersecção, |A'∩B'|;
- **Steve não aprende nada sobre o valor dos objetos** (nem os da intersecção nem os restantes) porque tudo o que vê são resultados de E, e estes parecem-lhe aleatórios, uma vez que ele não conhece K;
- A Alice e o Bob aprendem apenas quais os objetos que estão na intersecção e nada mais.

Segurança do protocolo:

- Depende, se assumirmos que o **Steve segue o protocolo fielmente,** então **sim**.
- Se suspeitarmos que o **Steve tenta fazer batota** (ou seja, enganar a Alice e o Bob para concluir com uma intersecção errada), então **não**

Protocolo OT-based

Exemplo:

- 'A' quer receber uma de quatro mensagens de 'B';
- 'A' envia chaves públicas para 'B', que usa essas chaves para criptografar suas mensagens;
- 'A', tendo apenas uma chave privada correspondente, pode desencriptar uma mensagem sem que 'B' saiba qual foi a mensagem escolhida.

Criptografia

Criptografia simétrica: utilizada para ocultar o conteúdo de blocos ou fluxos de dados de qualquer tamanho, incluindo mensagens, ficheiros, chaves de encriptação e palavras-passe.

Requisitos para um uso seguro:

- Um algoritmo de encriptação forte;
- O emissor e o recetor devem ter obtido cópias da **chave secreta de uma forma segura** e devem manter a chave segura.

Criptografia assimétrica: utilizada para ocultar pequenos blocos de dados, tais como chaves de encriptação e valores de funções hash, que são utilizados em assinaturas digitais.

Algoritmos de integridade de dados: utilizada para proteger blocos de dados, tais como mensagens, contra alterações.

Protocolos de autenticação: esquemas baseados na utilização de algoritmos criptográficos concebidos para autenticar a identidade de entidades.

Sistemas Criptográficos:

Caracterizados em 3 dimensões independentes:

- O tipo de operações utilizadas para transformar texto simples em texto cifrado:
  - Substituição;
  - Transposição.
- O número de chaves utilizadas:
  - Encriptação simétrica, de chave única, de chave secreta, convencional;
  - Encriptação assimétrica, de duas chaves ou de chave pública.
- A forma como o texto simples é processado:
  - Cifra de bloco;
  - Cifra de fluxo.

Segurança do esquema de encriptação:

- **Incondicionalmente seguro:** independentemente do tempo de que o adversário disponha, é-lhe impossível decifrar o texto cifrado simplesmente porque a informação necessária não existe.

- **Computacionalmente seguro:**
  - O custo de quebrar a cifra **excede o valor da informação encriptada**;

- O tempo necessário para quebrar a cifra **excede o tempo de vida útil da informação**.

Criptoanálise:

- O ataque baseia-se na natureza do algoritmo e em algum conhecimento das características gerais do texto simples;
- O ataque explora as características do algoritmo para tentar deduzir um texto simples específico ou para deduzir a chave que está a ser utilizada.

Ataque de força bruta:

- O atacante **tenta todas as chaves possíveis** num pedaço de texto cifrado até obter uma tradução inteligível em texto simples;
- Em média, metade de todas as chaves possíveis têm de ser tentadas para se obter sucesso.

Algoritmo de Cifra de César: substituição de cada letra do alfabeto pela letra que está três casas mais à frente no alfabeto (ex.: A torna-se D). Não é seguro pois é fácil decifrar a mensagem.

Cifra Monoalfabética: é usado um alfabeto cifado para substituição da mensagem. Não é seguro porque é relativamente fácil identificar as letras com maior ocorrência e assim decifrar a mensagem.

Cifra Playfair:

Baseada no uso de uma matriz de letras 5x5 construída usando uma palavra-chave. Esta matriz é inicialmente preenchida com as letras da mensagem (sem repetições) e depois pelas restantes letras do altabeto.

Pesquisa na Matriz:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

- As **letras repetidas** que se encontram no mesmo par são separadas por uma letra de preenchimento, como x. Por exemplo: balloon -> balxloon;
- Duas letras que se encontram na **mesma linha** da matriz são substituídas pelas letras à direita. Por exemplo: ar -> RM;

- Duas letras de texto simples que se encontrem na **mesma coluna** são substituídas pela letra que se encontra por baixo. Por exemplo: mu -> CM;
- Caso contrário, cada letra é substituída pela letra que se encontra na sua própria linha e na coluna ocupada pela outra letra. Por exemplo: hs -> BP e ea -> IM (ou JM, como o cifrador desejar).

Cifra Polialfabética: melhora a técnica monoalfabética simples usando **diferentes substituições monoalfabéticas** à medida que avançamos na mensagem de texto simples.

Cifra de Vigenère: cifras de substituição polialfabética mais simples, com um deslocamento de 3.

Cifra de Vernam: escolher uma palavra-chave que seja tão longa como o texto simples e que não tenha qualquer relação estatística com ele.

Bloco único (Melhoria da cifra Vernam):

- Utiliza uma chave aleatória tão longa como a mensagem, para que a chave não precise de ser repetida;
- A chave é utilizada para cifrar e decifrar uma única mensagem e depois é descartada;
- Cada nova mensagem requer uma nova chave com o mesmo comprimento da nova mensagem.
- ➢ O esquema é inquebrável

Cifra de transposição: o texto simples é escrito como uma sequência de diagonais e depois lido como uma sequência de linhas. Ex.:

To encipher the message "***meet me after the toga party***" with a rail **fence of depth 2**, we would write:

m e m a t r h t g p r y
e t e f e t e o a a t

Encrypted message is:
MEMATRHTGPRYETEFETEOAAT

Cifra de Fluxo: gera uma corrente contínua de bits pseudorrandômicos, chamada de fluxo, que é combinada **bit a bit** com a mensagem original para criar a mensagem cifrada.

Cifra de Bloco: divide a mensagem original em **blocos fixos de dados** e **cifra cada bloco separadamente** usando uma chave. Cada bloco de dados é tratado independentemente durante o processo de cifragem.

Cifra Feistel: uso de uma cifra que alterna substituições e permutações.

Padrão de criptografia de dados (DES): é um algoritmo de criptografia simétrica, o que significa que a mesma chave é usada para cifrar e decifrar a mensagem, os dados são encriptados em blocos de 64 bits utilizando uma chave de 56 bits. (16 rondas)

Padrão de criptografia avançado (AES): é um algoritmo de criptografia simétrica, o que significa que a mesma chave é usada tanto para cifrar quanto para decifrar a mensagem, funciona no modo de cifra de bloco, onde a mensagem é dividida em blocos fixos e cada bloco é cifrado separadamente. (múltiplas rondas)

Algoritmo Rivest-Shamir-Adleman (RSA):

$$C = M^e \bmod n$$
$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

- Tanto o emissor como o recetor devem conhecer o valor de n
- O emissor conhece o valor de e, e só o recetor conhece o valor de d
- ➢ Abordagem mais usada para criptografia de chave publica

Criptografia Homomórfica

Um homomorfismo é um mapa que preserva a estrutura entre duas estruturas algébricas do mesmo tipo, como grupos. Qualquer função pode ser expressa por adições e multiplicações.

Criptografia: usada para proteger dados em repouso ou em trânsito.

Criptografia Homomórfica: suporta cálculos em dados criptografados.

Homomorfismo Aditivo: Permite realizar operações de adição nos dados cifrados.

Homomorfismo Multiplicativo: Permite realizar operações de multiplicação nos dados cifrados.

Propriedade homomórficas do RSA: homomorfismo multiplicativo – texto cifrado vs texto cifrado.

Propriedades homomórficas de Goldwasser-Micali: homomorfismo aditivo – texto cifrado vs texto cifrado.

Propriedades homomórficas de Elgamal: homomorfismo multiplicativo – texto cifrado vs texto cifrado e texto cifrado vs texto simples, e aditivo se o valor não for muito grande.

Propriedades homomórficas de Paillier: homomorfismo aditivo – texto cifrado vs texto cifrado e texto cifrado vs texto simples e multiplicativo para texto cifrado vs texto simples.

Homomorfismo Total: Combina as propriedades aditiva e multiplicativa, permitindo realizar uma ampla variedade de operações. Pode ser considerado como **homomorfismo de anel**.

Esquema de chave secreta baseado em número inteiro:

- Geração de uma chave secreta – um grande número inteiro;
- **Encriptação:** c = pq + 2r + m;
- **Desencriptação:** m = (c (mod p))(mod 2) = (pq + 2r + m (mod p))(mod 2) = (2r + m)(mod 2).

Machine Learning

3 dimensões de ataque:

1. **Tempo de ataque:** tempo de decisão vs. tempo de treino.

   Ataques no momento da decisão:

   - O invasor ataca o modelo, não o algoritmo

   Ataques no momento do treino (ataque por envenenamento)**:**

   - O atacante adiciona aos dados não treinados manipulação maliciosa, resultando no envenenamento dos dados não treinados, o que leva a que o treino seja afetado

2. **Informações do invasor:** white-box vs. black-box.

   Ataques de white-box:

   - O invasor sabe tudo o que precisa saber sobre o sistema.

Ataques de black-box:

- Qualquer limitação da informação ao atacante;
- Não conhece exatamente o modelo, nem o algoritmo, nem as características; normalmente tem alguma informação sobre eles.

3. **Objetivos do ataque:** direcionado vs. confiabilidade.

Ataques de confiabilidade:

- Maximizam o erro de previsão (principalmente na aprendizagem supervisionada);
- Tornam a aprendizagem pouco confiável.

Ataques direcionados:

- O atacante tem uma instância de destino x com um rótulo verdadeiro y e pretende fazer com que o classificador a rotule erradamente como z
  Por exemplo: o atacante deseja que um sinal de paragem seja erradamente rotulado como um sinal de limite de velocidade

Ataques a modelos de machine learning que podem violar a privacidade:

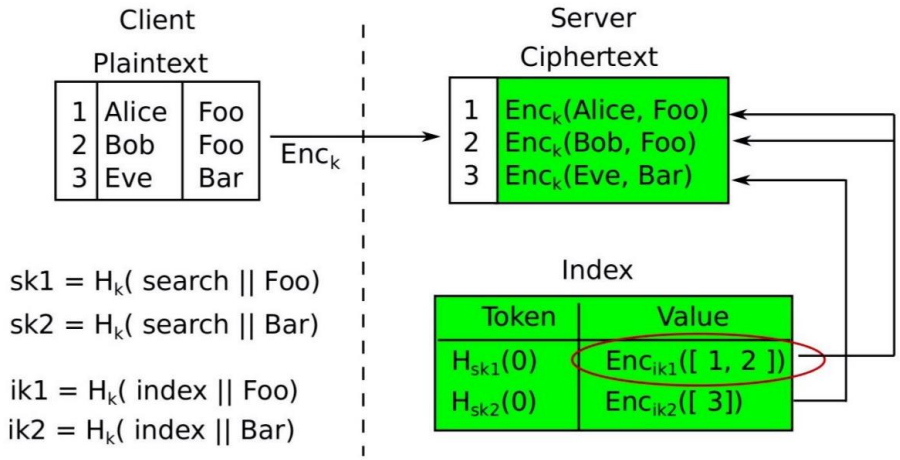
Inversão do modelo:

- Um adversário tenta reconstruir os dados de treino sensíveis observando o resultado de um modelo ML treinado;
- Ao consultar o modelo com uma entrada cuidadosamente elaborada, o atacante pode obter informações sobre os dados de treino que foram utilizados para criar o modelo.

Inferência de membros:

- Este ataque tem como objetivo determinar se um ponto de dados específico fazia parte do conjunto de dados de treino utilizado para treinar o modelo de ML;
- Ao observar o resultado do modelo para diferentes entradas, um atacante pode inferir informações de associação.

Criptografia Pesquisável



1. **Tokenizar** os dados de forma determinística (usando hash)

2. Enquanto os dados vão sendo armazenados no server, vai ser criada outra **tabela com o objetivo de armazenar os indexes.**

3. **Na procura** são geradas chaves para **gerar o token (s_k)** e **desencriptar o valor (ik).**

Por exemplo: No caso da procura pelo atributo do tipo Foo, é gerado sk_1 e ik_1, que são enviados para o server. Este depois usa sk_1 para encontrar o valor na tabela de index correspondente, que depois desencripta usando ik_1. O cliente por fim recebe o array com os indexes correspondentes às linhas onde o atributo é Foo, no caso [1, 2].