2.

① **Aim:** publish a lot of data without harming the privacy

② **Approach:** 1. Naive Anonymization • remove all identifiers linked attacks example: medical records) ③ **k-anonymity:** • 3 types of attributes: – Identifiers (PII) – Quasi-identifiers (can be identifying in combination) **def** Let $T(A_1, \dots A_n)$ be a table. a quasi-id. of T is a set of attributes $\{A_i, \dots, A_j\} \subseteq \{A_1, \dots, A_n\}$ whose release must be controlled. – Sensitive attributes • **Idea:** create anonymity sets, so that each individual cannot be guished from at least k-1 others. (its about quasi-identifiers) • **def** k-anonymity: Let $RT(A_1, \dots A_n)$ be a released table and QI_{RT} be the quasi-id. RT is said to satisfy k-anonymity if and only if for each $QI \subset QI_{RT}$ each sequence of values in $RT[QI_{RT}]$ appears with at least k occurrences in $RT[QI_{RT}]$ • Approaches to make data k-anonymous: 1. Generalization 2. Suppression • Find optimal k-anonymized table with generalization and suppression is NP-Hard ④ **Attacks against k-anonymity**

1. Homogeneity attack 2. Background knowledge attack

⑤ **ℓ-diversity** • A QI-block is ℓ-diverse if it contains at least ℓ well-represented values for the sensitive attribute S. A table is ℓ-diverse if every QI-block is ℓ-diverse

• Types: 1. Probabilistic – Freq. of most frequent value in equivalence class is bounded by $\frac{1}{\ell}$

2. Entropy entropy of the distribution of sensitive values in each equivalence class is at least $\log_2(\ell)$ • $H(X) = -\sum_{x \in X} p(x)\log_2 p(x)$ (level of uncertainty), X – freq. of sensetive values in each class

3. Recursive (c,ℓ)-diversity • $r_1 < c(r_\ell + r_{\ell+1} + \dots + r_m)$ r_i – freq of the i-th most freq • idea: the most freq. value r_1 does not appear too frequently. • Checking – straightforward, creating NP-complete. • **Attacks** 1. Skewness • distribution of sensitive 2. Semantic Similarity 3. Probabilistic Inference

⑥ **m-invariance** (for multy releases) 1. every eq. class E must have at least m rows. 2. the values for sensitive attributes s must all be different 3. set of distinct sensitive values in each eq. class must be the same. • methods: – generalize quasi-identifiers – counterfeit tuples • attacks the same ⑦ **t-closeness** • idea: distribution of sensitive attribute values within each eq. class should be „close" to the distribution of senset. attr. in the entire table. • methods: 1. for numeric attributes: normalize the distance by range $(\frac{i-j}{range})$ 2. for categorical: define distance based on a hierarchy • attacks: – background knowledge

3.

① **Setting:** – database D – query S with the response R_S – mechanism $M: S \to R_S$ ↳ 1. give statistically useful responses (utility) 2. without leaking sensetive info

② **Naive mechanism** a) always report correct and try to limit query types ↳ differencing attack (aggregating over large groups) б) add noise ↳ independent noise → attack with repeated queries (median filter) 2 too little noise → Dinur-Nissim reconstruction attack (• Such mechanisms are called blatantly non-private, • adding $o(\sqrt{n})$ noise is blatantly non-private

③ **Differential privacy** • D, $A: D \to T$ • Adversary get access to T and details of A • adversary should learn nothing new about an individual after seeing A and T, regardless any other background knowledge • **def** A mechanism $M: X \to Y$ is ε-diff. private (ε-DP) if for any two neighboring D_1 and D_2 (the same except one row): $\forall T \subseteq Y$. $P_r[M(D_1) \in T] \le e^{\varepsilon} P_r[M(D_2) \in T]$ • smaller ε ⇒ more privacy; for small ε: $e^{\varepsilon} \approx 1+\varepsilon$ • **Properties:** 1. Safety against post-processing 2. Sequential composition $\varepsilon + \varepsilon + \varepsilon$: $(k \cdot \varepsilon)$ 3. Parallel composition ($X = X_1 \cup X_2 \dots$; $M(X_1), M(X_2)$ – ε-diff.) 4. Group privacy: $P_r[M(D_1) \in T] \le e^{k \cdot \varepsilon} P_r[M(D_2) \in T]$, D_1 and D_2 differ in k rows

④ **Mechanisms for DP:** 1. Laplace mechanism: $A(S) + \eta$, η from $Lap(\mu, b)$, $f(x) = \frac{1}{2b}\exp\left(\frac{-|x-\mu|}{b}\right)$, $\delta^2 = 2b^2$ **def** Sensetivity $S(q)$ is the smallest number so that for $\forall D_1, D_2$: $|q(D_1) - q(D_2)| \le S(q)$ **def** ℓ_1-sensitivity: $\Delta_1^q = \max_{D_1, D_2: d(D_1,D_2) \le 1} |q(D_1) - q(D_2)|$ • so $b = \frac{S}{\varepsilon}$ ⇒ ε-diff. privacy • – Counting query: $S(q) = 1$ • Sum query and Average unbounded • Mechanism: Clipping 2. Exponential mechanism: D, R – set of outputs from which we want to select best element **def** Scoring function: $u: D \times R \to \mathbb{R}$ with sensetivity Δu: $\Delta u = \max_{R, D, D'} |u(D,R) - u(D',R)|$ • Exp. mechanism: given input d, set of output R, scoring function u samples output $r \in R$ with $P_r \sim \exp\left(\frac{\varepsilon \cdot u(d,r)}{2\Delta u}\right)$

⑤ **Unit of Privacy:** – One persone – Person-day ⑥ **Some examples:** 1. U.S. Broadband coverage dataset (unit – one device, ε = 0.2) 2. Post-secondary employment outcomes (unit – person, ε = 3) ⑦ **Approximate differential privacy** $\forall T \subset Y$ $P_r(M(D_1) \in T) \le e^{\varepsilon} P_r(M(D_2) \in T) + \delta$ ($1-\delta$ – DP; δ – no privacy) • Typical choice: $\delta < \frac{1}{n^2}$ $|D| = n$

• Properties: same as DP (Seq. composition, Post-processing, parallel composition) • Mechanism 1. Gaussian mechanism • for a query function $f(x)$: $G(x) = f(x) + \mathcal{N}(\sigma^2)$ • Gaussian M satisfies (ε,δ)-diff. privacy if $\delta^2 = \frac{2s^2(\ln\frac{1.25}{\delta})}{\varepsilon^2}$, where – s is the sensitivity of $f(x)$ – $\mathcal{N}(\delta^2)$ – gaussian distribution ⑧ **Vector-valued functions** • $f: D \to \mathbb{R}^k$ • examples: ML, histograms • sensitivity Δ_1^q: use maximum magnitude $\Delta_1^q = \max_{D_1, D_2: d(D_1,D_2) \le 1} |q(D_1) - q(D_2)|$, | | – chosen norm • Magnitude of a vector: 1. L_1 norm: $\|V\|_1 = \sum |V_i|$ ← Laplace mechanism 2 L_2 norm: $\|V\|_2 = \sqrt{\sum V_i^2}$ ← Gaussian • L_2 sensetivity generally smaller ⇒ more privacy ⑨ **Examples** 1 Linkedin's audience engagements api (user-month-analyst) 2. Urban mobility data (unit: whether an individual made a trip from A to B during week

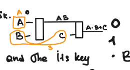
⑩ **Advanced composition** 1. k-fold adaptive composition • seq. of mechanism • each m_i is chosen adaptively, based on the outputs of m_{i-1} • examples: loops, recursive f • Advanced composition theorem: • if each m_i ε-DP then k-fold adaptive composition satisfies (ε',δ')-DP for δ' ≥ 0, $\varepsilon' = 2\varepsilon\sqrt{2k\log(\frac{1}{\delta'})}$ if m_i (ε,δ)-DP, then (ε', kδ+δ')-DP, δ' ≥ 0 and $\varepsilon' = 2\varepsilon\sqrt{2k\log(\frac{1}{\delta'})}$ • Advanced better then Sequential with bigger k ⑪ **Local Sensitivity** • $LS(f,x) = \max_{x': d(x,x')} |f(x) - f(x')|$ – max. difference on an actual dataset • cannot be used to achieve DP ⑫ **Propose-test-release** • analyst proposes an uper bound on local sensitivity • framework test whether the dataset is sufficienty far from a dataset with higher local sensitivity; 1. $A(f,x,k) = \max_{y: d(x,y) \le k} LS(f,y)$ 2. $D(f,x,b) = \arg\min_k A(f,x,k) > b$ 3. Test: $D(f,x,b) + Lap(\frac{1}{\varepsilon}) < \frac{\log(\frac{2}{\delta})}{2\varepsilon}$ 4. $f(x) + Lap(\frac{b}{\varepsilon})$ • PTR satisfies (ε,δ)-DP ⑬ **Smooth Sensitivity** • use an approximation of local sensetivity to calibrate noise • considers local sensetivity of all datasets in the neighborhood of the x • $\beta = \frac{\varepsilon}{2\log(2/\delta)}$; $S = \max e^{-\beta k} A(f,x,k)$ release $f(x) + Lap(\frac{2S}{\varepsilon})$ • larger then local ⑭ **Renyl dyfferential privacy** (RDP) $D_\alpha(P \| Q) = \frac{1}{\alpha - 1}\ln E_{x \sim Q}\left(\frac{P(x)}{Q(x)}\right)^\alpha$, α = ∞ ⇒ DP • Randomized mechanism M satisfies $(\alpha, \bar{\varepsilon})$-RDP if for $\forall x, x'$ $D_\alpha(M(x) \| M(x')) \le \bar{\varepsilon}$ • Any M which is $(\alpha, \bar{\varepsilon})$-RDP also is (ε,δ)-DP for any δ, $\varepsilon = \bar{\varepsilon} + \frac{\ln(\frac{1}{\delta})}{\alpha - 1}$ • To achive this we can use Gaussian mechanism: $F(x) = f(x) + \mathcal{N}(\delta^2)$, $\delta^2 = \frac{\Delta f^2 \alpha}{2\bar{\varepsilon}}$, Δf – L_2 sensetivity • $G(x) = (F_1(x), F_2(x))$ satisfies $(\alpha, \bar{\varepsilon}_1 + \bar{\varepsilon}_2)$-RDP, so lower privacy cost with RDP compose

⑮ **Zero-concentrated differential privacy** $D_\alpha(M(x) \| M(x')) \le \rho\alpha$, all $\alpha \in (1, \infty)$ • stronger than RDP, cuz all α • Conversion to (ε,δ)-DP: δ > 0 and $\varepsilon = \rho + 2\sqrt{\rho\log(\frac{1}{\delta})}$ • $F(x) = f(x) + \mathcal{N}(\delta^2)$, $\delta^2 = \frac{\Delta f^2}{2\rho}$; • $G(x) = (F_1(x), F_2(x))$: $(\rho_1 + \rho_2)$-zCDP

4.

① **Settings** • Players • $P = \{P_1, P_2 \dots P_n\}$ wants to compute $f(x_1 \dots x_n) = (y_1, y_2 \dots y_n)$ They can communicate • Assumption: ideal world ÷ Ƭ – completely trusted, incorruptible party – each P_i sends x_i to Ƭ – Ƭ computes $f(x_1 \dots x_n)$ and return result [in real world there is no Ƭ, but there is a protocol π, which is secure if emulates the ideal execution] so we simulate adversary • $P_r[A(\pi, E) = x] \cong P_r[S(f, E) = x]$ ← in ideal world • Types of adversaries: 1. Semi-honest (follows π, but sniffing) 2. Malicious (deviate from π) ② **Yao's Garbled circuits** (GC) protocol 1. Idea: lookup table 2. Oblivious transfer • R generates Sk, Pk and take random key p_k' • R sends $\begin{cases}(p_k, p_k') - \text{for } \sigma = 0 \\ (p_k', p_k) - \text{for } \sigma = 1\end{cases}$ • S returns $(Enc_{p_k}(m_0), Enc_{p_k'}(m_1))$

• R decrypts e_σ using Sk.



	A	B	C	AB
0	k_A^0	k_B^0	k_C^0	k_{AB}^1
1	k_A^1	k_B^1	k_C^1	k_{AB}^1

A and B: $Enc_{k_A^0, k_B^0}(k_{AB}^0)$, 0 1 (k_{AB}^0), 1 1 (k_{AB}^1), 0 0 (k_{AB}^0)

AB or C: $Enc_{k_{AB}^1, k_C^0}(1)$, $Enc_{k_{AB}^0, k_C^0}(0)$, $Enc_{k_{AB}^1, k_C^1}(1)$, $Enc_{k_{AB}^0, k_C^1}(1)$

• A sends to B: 1. circuit and the its key • B via OT ask keys for its input • B decrypt and sends output to A • Security: Semi-honest model (OT secure in semi-honest) • lose security if we evaluate a circuit more than once with diff. outputs

③ **BGW** • Idea: arithmetic functions with more than two inputs ⇒ arithmetic circuits • we can represent any polynomial with a combination of addition and multiplication gates • we operate on int. • idea: – secret S, that we split into n shares – only t+1 is enough to reconstruct s • Shamir's secret sharing I. – n players, threshold t ∈ [1, n[– select finite field $\mathbb{Z}_q$, q > n – secret is $s \in \mathbb{Z}_q$ – select n distinct elements $P_1, \dots P_n \in \mathbb{Z}_q \setminus \{0\}$ – sample random polynomial p with degree t and p(0) = s – compute the share of player i: $s_i = p(P_i)$ – reconstruct. Lagrange Polynomial Interpolation with any t+1, and evaluate p(0) = s II. • with multiplication degree of polynomial $p_c \le 2t$ ↳ degree reduction step • $p_c(0) = \sum_{i=1}^{2t+1} \lambda_i c_i$ • each player holds one of c_i each player generates and distributes shares c_{ij} of their c_i • each computes $C_j = \sum_{i=1}^{2t+1} \lambda_i c_{ij}$ • C_j is a share of c in a secret sharing with t • BGW: for every wire in the arithmetic circuit the players holds shares $[V_w]$ of the value on that wire; semi-honest model; 2t+1 ≤ n; adversary can corrupt up to t parties.

Zero-knowledge proofs • GMW compiler: run π and prove in zero-knowledge that every msg is the result of running π honestly • can transform any SMC π from semi-honest to malicious model.

6.

① **Background** • Logistic class more informative, cuz measures how far the prediction is from true label • SGD: sample batch in each round to estimate the gradient → Sample mini-batch B from training ex.. For each i∈B compute loss ℓ_{t_i} • Compute average gradient $g_t = \frac{1}{|B|}\sum \nabla_{h_t}\ell_{t_i}$ • Update weights $w_{t+1} = w - \eta g_t$

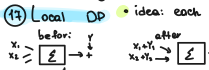
② **Attacks on ML:** 1. Membership inference 2. Shadow models (→ features or stat. properties; member or not) 3. Model Inversion (→ features or stat. properties; face ex) 4. Property Inference (reconstruct properties from trainig data; SNP+gen. info/text from LLM) 5. Training data extraction 6. Model extraction

③ **DP-SGD** • add DP noise to computation of gradient • Comp. avg gradient $g_t = \frac{1}{|B|}\sum_i \nabla_{h_t}L_{t_i} + \mathcal{N}(\delta^2)$, $\delta^2 = \frac{2S^2\ln(\frac{1.25}{\delta})}{\varepsilon^2}$ • Clipping: scaling vector: element wise division by L_2 norm results in L_2 norm (sensetivity) of 1 • Clipping threshold C: median of the norms of the unclipped gradient • Privacy accounting: 1. strong composition: $\delta^2 = \frac{2\ln(\frac{1.25}{\delta})}{\varepsilon^2}$, each step is (qε, qδ)-private, $q = \frac{B}{N}$, for T iterations, the training is $(q\varepsilon\sqrt{T\log(\frac{1}{\delta})}, Tq\delta)$-private 2. moments accountant: better method to analyze privacy cost, tighter bound $(q\varepsilon\sqrt{T}, \delta)$-private [same privacy can train longer] • Algorithm: – Sample mini-batch B – For each i∈B: ℓ_{t_i} – For each i∈B: $g_t(x_i) = \nabla_{h_t}L_{t_i}$ – Clip gradient: $\bar{g}_t(x) = \frac{g_t(x_i)}{\max(1, \frac{\|g_t(x)\|_2}{C})}$ Add noise, avg. gradients: $\tilde{g}_t = \frac{1}{|B|}\sum_i(\bar{g}_t(x_i) + \mathcal{N}(0, \delta^2C^2))$ – Update weights: $w_{t+1} = w - \eta\tilde{g}_t$ – Output w_{tn}, compute (ε,δ) • Issues: fixed choice of optimizer, params to tune, batch size and selection influence privacy guarantee

④ **PATE** (Private Aggreg. of teacher ensembles) • Independent of ML algo and of optimizers, loss f, batch • Main phases: 1. teacher training (data in n disjoint set, each used for train $f_i(x)$ without privacy) 2. aggregation (Label count $n_j(x)$ for class j is # of teacher models that predict class, label with highest count is agg. output; add Lap. noise: $f(x) = \arg\max_j\{n_j(x) + Lap(\frac{1}{\gamma})\}$ a single agg. is (2γ, 0)-DP 3. knowledge transfer (to get around limit number of users queries within; learns how to mimic agg. output of teacher ensemble; control privacy by limiting how many data records labeled.)

⑯ **Sparse vector techique** • Idea: perform a stream of queries with sensetivity 1; release only the identity of the first query which passes a test • Above threshold algorithm Inputs: $q_{1 \dots n}$, D, T, ε • $\hat{T} = T + Lap(\frac{2}{\varepsilon})$ • for i, q in queries: $v_i = Lap(\frac{4}{\varepsilon})$ if $q(D) + v_i \ge \hat{T}$: return i • return None • Application: auto-Clip (helps to find right bound) • It is possible to return multiple values

⑰ **Local DP** • idea: each part adds noise



• Randomized response • with $p = \frac{1}{2} + \alpha$ report true, with $\frac{1}{2} - \alpha$ report $y_i = 1 - x_i$ (flipped v • $\hat{x}_i = \frac{y_i - 1/2 + \alpha}{2\alpha}$, so amount $M(x) = \sum \hat{x}_i$ – unbiased estimator • Randomized response satisfies ε-DP if: $\alpha = \frac{e^{\varepsilon} - 1}{e^{\varepsilon} + 1}$ • accuracy lower than for central DP • Unary encoding • use one-hot encoding • $P_r[B'[i] = 1] = \begin{cases} p & \text{if } B[i] = 1 \\ q & \text{if } B[i] = 0\end{cases}$ • this M satisfies ε-DP, $\varepsilon = \ln\left(\frac{p(1-q)}{(1-p)q}\right)$ • then aggregator for each position: $A[i] = \frac{\sum_j B_j'[i] - nq}{p - q}$

• Example of this M. Google RAPPOR • hash the client value v onto the Bloom filter B • $B_i' = \begin{cases} 1 & \text{with } p = 1/2 f \\ 0 & \text{with } p = 1/2 f \\ B_i & \text{with } 1 - f\end{cases}$ • $P(S_i = 1) = \begin{cases} q & B_i' = 1 \\ p & B_i' = 0\end{cases}$ • Send S to server • Example: Chrome browser homepages

⑱ **Local DP at Apple:** Count mean sketch • Idea: encode the data element (instead of one-hot encoding) (collection data everytime) • difference to RAPPOR: much larger ε, no consideration of longitudinal • Attack against: Pool inference attack

7.

① **Federated SGD:** – Clients $\{c_1, \dots, c_n\}$, input $\{(x_1^{c_i}, y_1^{c_i}) \dots\}$ – Server chooses loss L, initializes weights w, chooses η – For iteration t ∈ [0, …, n]: • Server samples a random fraction of clients, C_r, sends them weights w • Eeach client $c \in C_r$: compute loss for all samples $L_{t_i}^c = L(h_t; x_i^c; y_i^c)$ • Compute gradient vector $g_t^c = \nabla_{h_t}L_{t_i}^c$ • Send g_t^c to server • Server compute avg. gradient $g_t = \frac{1}{|C_r|}\sum_c g_t^c$ • Server update $w_{t+1} = w - \eta g_t$

② **Attacks** • Adversary models: external, server, clients 1. Honest-but-curious server target membership inference (server records all model updates (gradients) from all C at all epochs, after training membership for each C) 2. Honest-but-curious server gradient inversion (attacker steals training data from C; updates dummy inputs and labels to min distance beetween gradients or add more params so client grad. memorizes training) 3. Malicious user or server: gradient ascent (is single datapoint part of dataset, report inverse gradient, so next iter. someones SGD will abruptly reduce grad. of loss)

③ **Privacy FL** 1. FL with central DP (ex. Gboard) (• protect against honest-but-curios server • algorithm (combination of DPSGD and federated SGD, users clip grad and server avg. grads and applies Gaussian noise) 2. FL with local DP • each user adds DP to their grad. • threat model: malicious server, users • drawbacks: need very large # users and amount of noise 3. FL with secure aggregation • malicious server, so use secure multi-party computation to agg. grad. • drawbacks: no DP (membership attacks) 4. FL with distributed DP and secure aggregation 1. Hybrid privacy-preserving FL • each iteration: one epoch of DPSGD (central DP, not local) on each client • gradients shared with threshold homomorphic encryption • assume t honest clients • accuracy is high, cuz secure agg. allows to use less noise 2. Distributed discrete gaussian for FL with secure aggregation • secure agg. is a black box → can plug in any SMC scheme, including ones for malicious server • client discretizes data to reduce communication overhead (fewer bits in discreatizat: reduced accuracy, reduced communication; client apply discrete Gaussian noise, achieves $\frac{1}{2}\varepsilon^2$-concentrated DP) ④ **Synthetic data generation**

• one column: distribution of data, histograms, #. of bins = # of unique values; add Lap. noise, parallel composition; sample points • multi-column: n-way marginal distribution; ok for small data; for big: low accuracy (noise overwhelms the signal)

1. PrivBayes (mitigate low cell counts) • represent correlations in Bayesian Network or Markov random fields (better acc) (derive low-dim tables; generate DP version of it, sample) • DP Bayesian Network: input d attributes $A_1 \dots A_d$, k > 0 • initialize empty Bayesian network N • consider all possible (k+1) attribute combination $A_{i1} \dots A_{ik}, A_j$ and evaluate mutual info $I(A_{i1} \times \dots \times A_{ik}; A_j)$ • Add DP noise to mutual info values and select combination with largest noisy mutual info • Add selected $A_{i1}, \dots A_{ik} \to A_j$ into N • Repeat with A_j & N

2. PATEGAN (GAN – generative adversarial networks) • random noise → generator → generator; original training data → discriminator → fake / real • add PATE into GANs: 1. training teacher-discriminator ensemble 2. training generator and student-discriminator (add noise on aggregation)

⑤ **Implementation challenges** • both implementation determine value ranges for columns from raw data → break DP • Win approach of NIST (MST): 1. measure 1,2,3-way marginals using gaussian mechanism 2. find synthetic dataset that has those marginals, using graphical model 3. generate data samples and measure perfomance (lower score – better) • future?: other data types, image\time data; with rare events.

8.

① **Cencorship phases:** 1. fingerprinting (destinations [IP, ports, domains], content, flow properties [packet lengths, time], protocol semantics 2. direct censorship (user-side [preinstalled software], publisher-side censorship, degrade performance, block destinations, corrupt routing info, corrupt flow content [modify html] corrupt protocol semantics) ② **Censorship resistance systems** (CRS) • phases: 1. Communication establishment (obtain credentials, CRS-specific info) • mechanisms – high churn access (freq. credentials change) – rate-limited access (proof of work) – active probing resistance (obfuscate aliveness or service) – true-based access (social graph) 2. Conversation • client connects to CRS (cencor should not be able to block, fingerprint link) • CRS server connects to publisher • mechanism: 1. focus on link user-CRS • mimicry (transform traffic so „looks" as whitelisted, content mimicry → high traffic volumes – or censor; • imitate protocol, introduce randomness → censors typically permit unknown traffic; flow mimicry) • tunnelling • covert channel (hide content), • traffic manipulation • destination obfuscation 2. publisher-CRS link: 1. content redundancy (store content on multiple servers) 2. distributed content storage (chunks)

③ **Tor** (send data streams anonymously) • communication: • directory servers • IP of directory authorities (DA) hardcoded into Tor client • every hour DAs publish consensus on all available Tor relays (each compiles list of all known relays + flags bandwidth, submits status-vote, collects votes, combines params and sends signature to other DAs • majority of DA should agree on the data → consensus • some tor relays not published by DAs: Tor bridges (Tor project maintains list of bridges and gives small list)

④ **Scramblesuit/OBFS4**

Tor | VPN
SOCKS
*
TCP
IP

• pseudo-random payload: traffic indistinguishable from randomness → no useful fingerprints for DPI • polymorphic: flow characteristic change • shared secret: server only speak protocol if client proves knowledge of secret

⑤ **Domain fronting/MEEK** • domain frontig: using different domain names for diff. protocols in the stack • not free • Google, Azure capable of this

⑥ **Snowflake:** • idea: proxy traffic through tmp proxies over WebRTC • volunteers run snowflake proxies (acts Tor bridge), broker manages and gives proxy addresses to clients – use domain fronting → high cost of blocking – only communication with broker is domain-fronted → lower cost compared to meek

⑦ **Conjure** • idea: instead of hiding Tor traffic integrate in into network core • IPS deploys Conjure station: client connect to phantom proxy: unused address in IPS address range • station injects packets as if there was true proxy available • hard to block: censor need to block all websites served by this ISP • PF_Ring: interface that allows conjure station to tap ISP traffic

9.

① **Privacy design strategies:** 1. minimize (processing of personal data) 2. separate (isolate data) 3. abstract (how data was processed) 4. hide (protect personal data) 5. inform (subject about processing their data) 6. control (subject can control data) 7. enforce (policy commitment) 8. demonstrate (document, report results)

② **DevPrivOPS** 1. Plan (select privacy design strategies, threat modeling: apply LINDDUN framework risk analysis) 2. code (itegrate libs that align with design objectives, document → TILT/TIRA), add logging, monitoring 3. build (flag outdated versions, create distinct versions for diff. groups) 4. test 5. Release 6. deploy (issue: foreign jurisdictions, use canary releases and A/B test to mitigate risk of unauthorized data processing) 7. operate (penetration testing, observation and cross-verification against prev. assumption) 8. monitor (logging, auto-reports)

③ **TILT:** Transparency info language and toolkit • standardized machine-readable format for transparency info in accordance with EU data protection law • data subject get info they need (choice of interface + of filtering) • data controllers collect and process info efficiently. • Ex. GDPR obligations → categorization → detailed analysis → formal language def → machine-readable format → uniformly represented transparency info.

④ **TIRA** • extends OpenAPI specification to allow developers to annotate transparency info.

5.

① **Privacy Metrics** 1. Entropy (high → more uncertainty $Priv = H(X) = -\sum p(x)\log_2 p(x)$ (adversary estimate; ex. users; guess) 2. Expected estimation err $Priv_{AEE} = \sum p(x|y) d(x, x^*)$ (observations; true) 3. Amount of Leaked informatio $Priv_{ALI} = |S|$ 4. Relative entropy (number of bits of prob. info revea to adversary) $Priv_{RE} = D_{KL}(X^* \| X)$ 5. Adv. sucess rate

② **Utility metrics** (how useful private is vs non-private) 1. General (Statistics, distance (Wasserstein, Kullback–Leibler, tot-var), confidence intervals, 2. Task-specific (error, classification (Acc, Precision, Recall, F1, AU

③ **Perfomance** (Many steps) • Time • Communication • Storage

1.

	Properties	Threats
a) HARD:	1. Unlinkability	Linkability
	2. Anonymity and Pseudonymity	Identifiability
	3. Plausible deniability	Non-repudiation
	4. Undetectability	Detectability
	5. Confidentiality	Disclosure of info
б) Soft	1. Content awareness content	Unawarenes
	2. Policy and consent compliance	policy and consent Noncompliance

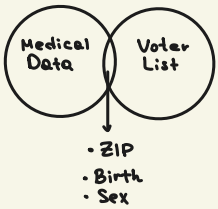
4 Example: NYC TAXI Data • fare and trip data)
5. Example: Netflix (ranking competition)
6. Example: Apple Quicktype
7. Example: Genomics (SNPs)
8. Example: Sugarbeet auction

① Introduction

1. Privacy types:
   1. Location 2. State of Body and Mind 3. Behavior + Action 4. Social life 5. Media
2. Privacy properties

	Properties	Threats
a) HARD:	1. Unlinkability	Linkability
	2. Anonymity and Pseudonymity	Identifiability
	3. Plausible deniability	Non-repudiation
	4. Undetectability	Detectability
	5. Confidentiality	Disclosure of information
b) Soft	1. Content awareness	content Unawareness
	2. Policy and consent compliance	policy and consent Noncompliance

3. Example: Medical Records (Group Insurance Commission)
   - Voter list was bought in addition (← Adversary has side info)



4. Example: NYC TAXI Data
   - fare data and trip data
5. Example: Netflix (ranking competition)
6. Example: Apple Quicktype
7. Example: Genomics (SNPs)
8. Example: Sugarbeet auction

② k-anonymity

1. Aim: publish a lot of data without harming the privacy
2. Approache: 1. Naive Anonymization
   - remove all identifiers
   - linked attacks (example: medical records)
3. k-anonymity:
   - 3 types of attributes:
     - Identifiers (PII)
     - Quasi-identifiers (can be identifying in combination)

     def Let $T(A_1, \dots, A_n)$ be a table. a quasi-id. of T is a set of attributes $\{A_i, \dots, A_j\} \subseteq \{A_1, \dots, A_n\}$ whose release must be controlled.
     - Sensitive attributes
   - Idea: create anonymity sets, so that each individual connot be distinguished from at least k-1 others. (its about quasi-identifiers)
   - def k-anonymity: Let $RT(A_1, \dots A_n)$ be a released table and QI_{RT} be the quasi-id. RT is said to satisfy k-anonymity if and only if for each $QI \subset QI_{RT}$ each sequence of values in $RT[QI_{RT}]$ appears with at least k occurrences in $RT[QI_{RT}]$
   - Approaches to make data k-anonymous:
     1. Generalization
     2. Suppression
   - Find optimal k-anonymized table with generalization and suppression is NP-Hard.
4. Attacks against k-anonymity
   1. Homogeneity attack
   2. Background knowledge attack
5. ℓ-diversity
   - A QI-block is ℓ-diverse if it contains at least ℓ well-represented values for the sensitive attribute S. A table is ℓ-diverse if every QI-block is ℓ-diverse.
   - Types:
     1. Probabilistic

- Freq. of the most frequent value in an equivalence class is bounded by $\frac{1}{e}$

2. Entropy

- entropy of the distribution of sensitive values in each equivalence class is at least $\log_2(e)$
- $H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$, X - freq. of sensetive values in each eq. class (level of uncertainty)

3. Recursive (c, e) - diversity

- $r_i < c(r_e + r_{e+1} + \dots + r_m)$, r_i - freq of the i-th most freq. value
- idea: the most freq. value r_1 does not appear too frequently.

- Checking for l-diversity straightforward, but creating NP-complete.
- Attacks:

1. Skewness
   - Overall distribution of sensitive values
2. Semantic Similarity
3. Probabilistic Inference

⑥ m-invariance (for multy releases)

1. every eq. class E must have at least m rows.
2. the values for sensitive attributes s must all be different
3. set of distinct sensitive values in each eq. class must be the same.

- methods:
  - generalize quasi-identifiers
  - counterfeit tuples
- attacks the same

⑦ t-closeness

- idea: distribution of sensitive attvibute values within each eq. class should be „close" to the distribution of senset. attrib. in the entive table.
- methods:
  1. for numeric attributes: normalize the distance by range $\frac{i-j}{\text{range}}$
  2. for categorical: define distance based on a hierarchy
- attacks: - background knowledge

③ Differential Privacy

① Setting:

- database D
- query S with the response Rs
- mechanism M: S ⟶ Rs
  - ↳ 1. give statistically useful responses (utility)
  - 2. without leaking sensetive info (privacy)

② Naive mechanism

a) always report correct and try to limit query types

↳ differencing attack (aggregating over large groups)

b) add noise

↳ 1. independent noise ⟶ attack with repeated queries (median filter)

2. too little noise ⟶ Dinur-Nissim reconstruction attack

- Such mechanisms are called blatantly non-private,
- adding $o(\sqrt{n})$ noise is blatantly non-private.

③ Differential privacy

- D, A: D → T
- Adversary get access to T and details of A
- adversary should learn nothing new about an individual after seeing A and T, regardless any other background knowledge
- def A mechanism M: X ⟶ Y is ε-diff. private (ε-DP) if for any two neighboring D_1 and D_2 (the same except one row):

$$\forall T \subseteq Y: Pr[M(D_1) \in T] \leq e^{\varepsilon} Pr[M(D_2) \in T]$$

- smaller ε => more privacy; for small ε: $e^{\varepsilon} \approx 1+\varepsilon$

- Properties:
  1. Safety against post-processing
  2. Sequential composition ($\varepsilon_1+\varepsilon_2 : k\cdot\varepsilon$)
  3. Parallel composition ($X=X_1 \cup X_2 \ldots$; $M(X_1), M(X_2)$ - ε-diff.)
  4. Group privacy: $Pr[M(D_1) \in T] \leq e^{k\cdot\varepsilon} Pr[M(D_2) \in T]$, D_1 and D_2 differ in k rows

④ Mechanisms for DP:

1. Laplace mechanism:

$A(S) + \eta$, η from $Lap(\mu, b)$, $f(x) = \frac{1}{2b}\exp\left(\frac{-|x-\mu|}{b}\right)$, $\sigma^2 = 2b^2$

def Sensetivity $S(q)$ is the smallest number so that for $\forall D_1, D_2$:

$$|q(D_1) - q(D_2)| \leq S(q)$$

def ℓ_1-sensitivity: $\Delta_1^q = \max_{D_1, D_2 : d(D_1, D_2) \leq 1} |q(D_1) - q(D_2)|$

- so $b = \frac{S}{\varepsilon}$ => ε-diff. privacy
- -Counting query: $S(q) = 1$
- - Sum query and Average unbounded
- Mechanism: Clipping

2. Exponential mechanism

- D, R - set of outputs from which we want to select best element
- **def** Scoring function: $u: D \times R \to \mathbb{R}$ with sensetivity Δu

$$\Delta u = \max_{R, D, D'} |u(D, R) - u(D', R)|$$

- Exp. mechanism:
  - given input d, set of output R, scoring function u

samples output $r \in R$ with $Pr \sim \exp\left(\frac{\varepsilon \cdot u(d, r)}{2\Delta u}\right)$

⑤ "Unit" of Privacy:

- One persone
- Person-day

⑥ Some examples:

1. U.S. Broadband coverage dataset (unit - one device, $\varepsilon = 0.2$)
2. Post-secondary employment outcomes (unit - person, $\varepsilon = 3$)

⑦ Approximate differential privacy

$$\forall T \subseteq Y: Pr(M(D_1) \in T) \leq e^{\varepsilon} Pr(M(D_2) \in T) + \delta' \quad \begin{cases} 1 & \text{1-}\varepsilon\text{- DP} \\ \delta & \text{- no privacy} \end{cases}$$

- Typical choice: $\delta' < \frac{1}{n^2}$, $|D| = n$
- Properties: same as DP (Seq. composition, Post-processing, parallel composition)
- Mechanism

1. Gaussian mechanism

- for a query function $f(x)$: $G(x) = f(x) + \mathcal{N}(\sigma^2)$

- Gaussian M satisfies (ε,δ)-diff. privacy if $\delta^2 = \frac{2S^2(\ln\frac{1.25}{\delta})}{\varepsilon^2}$, where
  - S is the sensitivity of $f(x)$
  - $\mathcal{N}(\delta^2)$ - gaussian distribution

⑧ Vector-valued functions

- $f: D \to \mathbb{R}^k$
- examples: ML, histograms
- sensitivity Δ_1^q: use maximum magnitude

$$\Delta_1^q = \max_{D_1,D_2:\, d(D_1,D_2)\le 1} |q(D_1) - q(D_2)|, \quad |\ | - \text{chosen norm}$$

- Magnitude of a vector:
  1. L_1 norm: $\|V\|_1 = \Sigma|V_i|$ ← Laplace mechanism
  2. L_2 norm: $\|V\|_2 = \sqrt{\Sigma V_i^2}$ ← Gaussian
- L_2 sensetivity generally smaller => more privacy

⑨ Examples:

1. Linkedin's audience engagements api (user-month-analyst)
2. Urban mobility data (unit: whether an individual made a trip from A to B during week W)

⑩ Advanced composition:

1. k-fold adaptive composition
   - seq. of mechanism
   - each m_i is chosen adaptively, based on the outputs of m_{i-1}
   - examples: loops, recursive function
- Advanced composition theorem:
  - if each m_i ε-DP then k-fold adaptive composition satisfies (ε',δ')-DP for $\delta' \ge 0$, $\varepsilon' = 2\cdot\varepsilon\sqrt{2\cdot k\cdot\log(\frac{1}{\delta'})}$
  - if m_i (ε,δ)-DP, then $(\varepsilon', k\delta+\delta')$-DP, $\delta' \ge 0$ and $\varepsilon' = 2\cdot\varepsilon\sqrt{2\cdot k\cdot\log(\frac{1}{\delta'})}$
  - Advanced better than Sequential with bigger k

⑪ Local sensitivity

- $LS(f,x) = \max_{x':\, d(x,x')} |f(x) - f(x')|$ - max. difference on an actual dataset
- cannot be used to achieve DP

⑫ Propose-test-release

- analyst proposes an uper bound on local sensitivity

- framework test whether

1. $A(f,x,k) = \max_{y: d(x,y)\le k} LS(f,y)$
2. $D(f,x,b) = \operatorname{argmin}_k A(f,x,k) > b$
3. Test: $D(f,x,b) + Lap(\frac{1}{\varepsilon}) < \frac{\log(\frac{2}{\delta})}{2\varepsilon}$
4. $f(x) + Lap(\frac{b}{\varepsilon})$

- PTR satisfies (ε,δ)-DP

(13) Smooth Sensitivity

- use an approximation of local sensetivity to calibrate noise
- considers local sensetivity of all datasets in the neighborhood of the x
- $\beta = \frac{\varepsilon}{2\log(2/\delta)}$; $S = \max e^{-\beta k} A(f,x,k)$, release $f(x) + Lap(\frac{2S}{\varepsilon})$
- larger than local

(14) Renyl dyfferential privacy (RDP)

- $D_\alpha(P\|Q) = \frac{1}{\alpha-1} \ln E_{x\sim Q}\left(\frac{P(x)}{Q(x)}^\alpha\right)$, $\alpha=\infty \Rightarrow$ DP
- Randomized mechanism M satisfies $(\alpha,\bar{\varepsilon})$-RDP if for $\forall x,x'$

$$D_\alpha(M(x)\|M(x')) \le \bar{\varepsilon}$$

- Any M which is $(\alpha,\bar{\varepsilon})$-RDP also is (ε,δ)-DP for any δ, $\varepsilon = \bar{\varepsilon} + \frac{\ln(\frac{1}{\delta})}{\alpha-1}$
- To achive this we can use Gaussian mechanism:

$$F(x) = f(x) + \mathcal{N}(\delta^2), \quad \delta^2 = \frac{\Delta f^2 \alpha}{2\bar{\varepsilon}}, \ \Delta f\text{-}L_2 \text{ sensetivity}$$

- $G(x) = (F_1(x), F_2(x))$ satisfies $(\alpha, \bar{\varepsilon}_1 + \bar{\varepsilon}_2)$-RDP, so

lower privacy cost with RDP compose

(15) Zero-concentrated differential privacy

- $D_\alpha(M(x)\|M(x')) \le \rho\alpha$, all $\alpha \in (1,\infty)$
- stronger than RDP, cuz all α
- Conversion to (ε,δ)-DP: $\delta > 0$ and $\varepsilon = \rho + 2\sqrt{\rho \log(\frac{1}{\delta})}$
- $F(x) = f(x) + \mathcal{N}(\delta^2)$, $\delta^2 = \frac{\Delta f^2}{2\rho}$;
- $G(x) = (F_1(x), F_2(x))$: $(\rho_1+\rho_2)$-zCDP

(16) Sparse vector technique

- Idea: perform a stream of queries with sensetivity 1; release only the identity of the first query which passes a test
- Above threshold algorithm

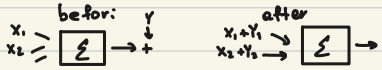
Inputs: $q_{1\ldots n}$, D, T, ε

- $\hat{T} = T + Lap(\frac{2}{\varepsilon})$
- for i, q in queries:
  $v_i = Lap(\frac{4}{\varepsilon})$
  if $q(D) + v_i \geq \hat{T}$: return i
- return None

- Application: auto-clip (helps to find right bound)
- It is possible to return multiple values

(17) Local DP

- idea: each part adds noise



- Randomized response
  - with $p = \frac{1}{2} + \alpha$ report true, with $\frac{1}{2} - \alpha$ report $y_i = 1 - x_i$ (flipped value)
  - $x_i = \frac{y_i - 1/2 + \alpha}{2\alpha}$, so amount $M(x) = \sum \hat{x}_i$ - unbiased estimator
  - Randomized response satisfies ε-DP if: $\alpha = \frac{e^{\varepsilon} - 1}{e^{\varepsilon} + 1}$
  - accuracy lower than for central DP
- Unary encoding
  - use one-hot encoding
  - $Pr[B'[i] = 1] = \begin{cases} p & \text{if } B[i] = 1 \\ q & \text{if } B[i] = 0 \end{cases}$
  - this M satisfies ε-DP, $\varepsilon = \ln\left(\frac{p(1-q)}{(1-p)q}\right)$
  - then aggregator for each position:
    $$A[i] = \frac{\sum_j B'_j[i] - nq}{p - q}$$
- Example of this M: Google RAPPOR
  - hash the client value v onto the Bloom filter B
  - $B'_i = \begin{cases} 1 & \text{with } p = 1/2 f \\ 0 & \text{with } p = 1/2 f \\ B_i & \text{with } 1 - f \end{cases}$
  - $P(S_i = 1) = \begin{cases} q & B_i = 1 \\ p & B_i = 0 \end{cases}$
  - Send S to server
- Example: Chrome browser homepages

(18) Local DP at Apple: Count mean sketch

- Idea: encode the data element (instead of one-hot encoding)

collection data over time

- difference to RAPPOR: much larger ε, no consideration of longitudinal privacy.
- Attack against: Pool inference attack

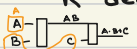
④ Secure multi-party computation

① Settings

- Players: $P=\{P_1, P_2 \dots P_n\}$ wants to compute $f(x_1 \dots x_n) = (y_1, y_2 \dots y_n)$
  They can communicate
- Assumption: ideal world:
  - Υ - completely trusted, incorruptible party
  - each P_i sends x_i to Υ
  - Υ computes $f(x_1 \dots x_n)$ and return result

  [in real world there is no Υ, but there is a protocol π, which is secure if emulates the ideal execution] so we simulate adversary
  - $Pr[A(\pi, E) = x] \cong Pr[S(f, E) = x]$ ← in ideal world
- Types of adversaries:
  1. Semi-honest (follows π, but snitting)
  2. Malicious (deviate from π)

② Yao's Garbled circuits (GC) protocol

1. Idea: lookup table
2. Oblivious transfer
   - R generates S_k, P_k and take random key p_k'
   - R sends $\begin{cases} (p_0, p_k') - \text{for } \delta=0 \\ (p_k', p_k) - \text{for } \delta=1 \end{cases}$
   - S returns $(Enc_{p_0}(m_0), Enc_{p_1}(m_1))$.
   - R decrypts e_δ using s_k.
- 

	A	B	C	AB
0	k_A^0	k_B^0	k_C^0	k_{AB}^0
1	k_A^1	k_B^1	k_C^1	k_{AB}^1

A and B	AB or C
$Enc_{k_A^0, k_B^0}(k_{AB}^0)$	$Enc_{k_{AB}^1, k_C^1}(1)$
0 1 (k_{AB}^0)	$Enc_{k_{AB}^0, k_C^1}(0)$
1 1 (k_{AB}^1)	$Enc_{k_{AB}^1, k_C^1}(1)$
0 0 (k_{AB}^0)	$Enc_{k_{AB}^0, k_C^1}(1)$

- A sends to B: 1. circuit and the its key
- B via OT ask keys for its input
- B decrypt and sends output to A

- Security: Semi-honest model (OT secure in semi-honest)
- We lose security if we evaluate a circuit more than once with diff. outputs

③ BGW

- Idea: arithmetic functions with more than two inputs => arithmetic circuits
- we can represent any polynomial with a combination of addition and multiplication gates
- we operate on int.
- idea: - secret S, that we split into n shares
  - only $t+1$ (called threshold) is enough to reconstruct s

- **Shamir's secret sharing**

I. - n players, threshold $t \in [1, n[$
- select finite field Z_q, $q > n$
- secret is $s \in Z_q$
- select n distinct elements $P_1, \dots P_n \in Z_q \setminus \{0\}$
- sample random polynomial p with degree t and $p(0) = s$
- compute the share of player i: $s_i = p(P_i)$
- reconstruct: Lagrange Polynomial Interpolation with any $t+1$, and evaluate $p(0) = s$

II. • with multiplication degree of polynomial $p_c \leq 2t$
↳ **degree reduction step**
- $p_c(0) = \sum_{i=1}^{2t+1} \lambda_i c_i$
- each player holds one of c_i
  ↳ • each player generates and distributes shares c_{ij} of their c_i
  • each computes $C_j = \sum_{i=1}^{2t+1} \lambda_i c_{ij}$
  • C_j is a share of c in a secret sharing with t

Example
- 5 and 2, Z_{11}
- 3 parties
- $\delta_1(x) = 5 + 8x$, $\delta_2(x) = 2 + 9x$
- S: 2, 10, 7; λ: 0.9, 7
- new shares 0, 2.5
- $\delta_3(x) = 0 + 3x$ for 1, $\delta_4(x) = 2 + 6x$
- $S_5(x) = 5 + 2x$ (3.6.8), (8.3.9); (9.9.9)
- 1: (3.8.3), 2 (6 2.8) 3: (9.9.9)

- BGW: for every wire in the arithmetic circuit the players holds shares $[v_w]$ of the value on that wire; semi-honest model; $2t+1 \leq n$; adversary can corrupt up to t parties.

④ Zero-knowledge proofs

- GMW compiler: run π and prove in zero-knowledge that every msg is the result of running π honestly
- can transform any SMC π from semi-honest to malicious model.

⑤ Privacy Metrics

① Privacy Metrics

1. **Entropy** (high → more uncertainty

$$Priv = H(X) = -\sum p(x) \log_2 p(x)$$

(adversary estimate; ex. users; guess)

2. **Expected estimation error**

$$Priv_{AEE} = \sum p(x|y) d(x, x^*)$$

(observations; true)

3. **Amount of Leaked information**

$$Priv_{ALI} = |S|$$

4. **Relative entropy** (number of bits of prob. info revealed to adversary)

$$Priv_{RE} = D_{KL}(X^* \| X)$$

5. **Adv. Sucess rate**

② Utility metrics (how useful private is vs non-private)

1. General (Statistics, distance (Wasserstein, Kullback, total-var), confidence intervals,
2. Task-specific (error, classification (Acc, Precision, Recall, F_1, AUC)

③ Performance (Many steps)

- Time
- Communication
- Storage

⑥ Privacy for ML

① Background

- Logistic class more informative, CE measures how far the prediction is from true label
- SGD: sample batch in each round to estimate the gradient

→ Sample mini-batch B from training ex.

For each $i \in B$ compute loss ℓ_i

- Compute average gradient $g_t = \frac{1}{|B|} \sum \nabla_{h_t} \ell_{t,i}$
- Update weights $w_{t+1} = w - \eta g_t$

② Attacks on ML:

1. Membership inference
2. Shadow models (→ features or stat. properties; member or not)
3. Model Inversion (→ features or stat. properties; face ex.)
4. Property Inference (reconstruct properties from trainig data; SNP+gen. info/text from LLM)
5. Training data extraction
6. Model extraction

③ DP-SGD

- add DP noise to computation of gradient
- Comp. avg gradient $g_t = \frac{1}{|B|} \sum_i \nabla_{h_t} L_{t,i} + \mathcal{N}(0, \sigma^2)$, $\sigma^2 = \frac{2S^2 \ln\left(\frac{1.25}{\delta}\right)}{\varepsilon^2}$
- Clipping: scaling vector: element wise division by L_2 norm results in L_2 norm (sensetivity) of 1
- Clipping threshold C: median of the norms of the unclipped gradient
- Privacy accounting: 1. strong composition: $\varepsilon^2 = \frac{2\ln\left(\frac{1.25}{\delta}\right)}{\varepsilon^2}$, each step is $(q\varepsilon, q\delta)$-private, $q = \frac{B}{N}$,

for T iterations, the training is $(q\varepsilon\sqrt{T\log\left(\frac{1}{\delta}\right)}, Tq\delta)$-private

2. moments accountant: better method to analyze privacy cost, tighter bound

$(q\varepsilon\sqrt{T}, \delta)$-private; (same privacy, can train longer)

- Algorithm:
  - Sample mini-batch B
  - For each $i \in B$: ℓ_i
  - For each $i \in B$: $g_t(x_i) = \nabla_{h_t} L_{t,i}$
  - Clip gradient: $\bar{g}_t(x) = \frac{g_t(x_i)}{\max\left(1, \frac{\|g_t(x)\|_2}{C}\right)}$
  - Add noise, avg. gradients: $\tilde{g}_t = \frac{1}{|B|} \sum_i \left(\bar{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2)\right)$
  - Update weights: $w_{t+1} = w - \eta \tilde{g}_t$
  - Output w_{t+1}, compute (ε, δ)

- Issues: fixed choice of optimizer, params to tune, batch size and selection influence the privacy guarantee.

④ PATE (Private Aggreg. of teacher ensembles)

- Independent of ML algo and of optimizers, loss f, batch
- Main phases: 1. teacher training (data in n disjoint set, each used for train $f_i(x)$ without privacy)

  2. aggregation (Label count $n_j(x)$ for class j is # of teacher models that predict class, label with highest count is agg. output; add Lap. noise: $f(x) = \text{argmax}_j \{n_j(x) + \text{Lap}(\frac{1}{\gamma})\}$ a single agg. is $(2\gamma, 0)$-DP

  3. knowledge transfer (to get around limit number of users queries within; learns how to mimic agg. output of teacher ensemble; control privacy by limiting how many data records labeled.)

⑦ Federated learning

① Federated SGD:

- Clients $\{c_1, \dots, c_n\}$, input $\{(x_1^{c_i}, y_1^{c_i}) \dots\}$
- Server chooses loss L, initialises weights w, chooses η
- For iteration $t \in [0, \dots \eta]$:
  - Server samples a random fraction of clients, C_r, sends them weights w
  - Each client $c \in C_r$: compute loss for all samples $L_{t_i}^c = L(h_t; x_i^c; y_i^c)$
  - Compute gradient vector $g_t^c = \nabla_{h_t} L_{t_i}^c$
  - Send g_t^c to server
  - Server compute avg gradient $g_t = \frac{1}{|C_r|} \sum_c g_t^c$
  - Server update $w_{t+1} = w - \eta g_t$

② Attacks

- Adversary models: external, server, clients.

1. Honest-but-curious server: target membership inference (server records all model updates (gradients) from all C at all epochs, after training membership for each C)

2. Honest-but-curious server: gradient inversion (attacker steals training data from C; updates dummy inputs and labels to min distance between gradients or add more params so client grad. memorizes training)

3. Malicious user or server: gradient ascent (is single datapoint part of dataset, report inverse gradient, so next iter. someones SGD will abruptly reduce grad. of loss)

③ Privacy FL

1. FL with central DP (ex. Gboard)

(• protect against honest-but-curios server

• algorithm (combination of DPSGD and federated SGD, users clip grad and server avg. grads and applies Gaussian noise)

2. FL with local DP

- each user adds DP to their grad.
- threat model: malicious server, users
- drawbacks: need very large # users and amount of noise

3. FL with Secure aggregation

- malicious server, so use secure multi-party computation to agg. grad.

- drawbacks: no DP (membership attacks)

4. FL with distributed DP and secure aggregation

1. Hybrid privacy-preserving FL

- each iteration: one epoch of DPSGD (central DP, not local) on each client
- gradients shared with threshold homomorphic encryption
- assume t honest clients
- accuracy is high, cuz secure agg. allows to use less noise

2. Distributed discrete gaussian for FL with secure aggregation

- secure agg. is a black box → can plug in any SMC scheme, including ones for malicious server
- client discretizes data to reduce communication overhead (fewer bits in discreatization: reduced accuracy, reduced communication; client apply discrete Gaussian noise, achieves $\frac{1}{2}\varepsilon^2$-concentrated DP)

④ Synthetic data generation

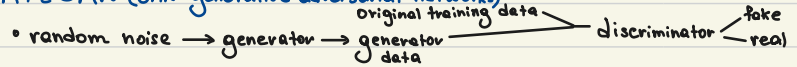
- one column: distribution of data, histograms, #. of bins = # of unique values; add Lap. noise, parallel composition; sample points
- multi-column: n-way marginal distribution; ok for small data; for big: low accuracy (noise overwhelms the signal)

1. PrivBayes (mitigate low cell counts)

- represent correlations in Bayesian Network or Markov random fields (better acc) (derive low-dim tables; generate DP version of it, sample)
- DP Bayesian Network: input d attributes $A_1 \dots A_d, k>0$
  - initialize empty Bayesian network N
  - consider all possible $(k+1)$ attribute combination $A_{i_1} \dots A_{i_k}, A_j$ and evaluate mutual info $I(A_{i_1} \times \dots \times A_{i_k}; A_j)$
  - Add DP noise to mutual info values

  and select combination with largest noisy mutual info.
  - Add selected $A_{i_1}, \dots A_{i_k} \to A_j$ into N
  - Repeat with A_j & N

2. PATEGAN (GAN - generative adversarial networks)



- add PATE into GANs: 1. training teacher-discriminator ensemble

2. training generator and student-discriminator (add noise on aggregation)

⑤ Implementation challenges

- both implementation determine value ranges for columns from raw data → break DP
- Win approach of NIST (NSI): 1. measure 1,2,3-way marginals using gaussian mechanism
  2. find synthetic dataset that has those marginals, using graphical model
  3. generate data samples and measure performance (lower score-better)
- future ?: other data types, image\time data; with rare events.

⑧ Censorship

① Cencorship phases:

1. fingerprinting (destinations [IP, ports, domains], content, flow properties [packet lengths, time], protocol semantics.

2. direct censorship (user-side [preinstalled software], publisher-side censorship, degrade performance, block destinations, corrupt routing info, corrupt flow content [modify html], corrupt protocol semantics)

② Censorship resistance systems (CRS)

- phases: 1. Communication establishment (obtain credentials, CRS-specific info)

- mechanisms – high churn access (freq. credentials change) – rate-limited access (proof of work)

– active probing resistance (obfuscate aliveness or service)

– true-based access (social graph)

2. Conversation

- Client connects to CRS (censor should not be able to block, fingerprith link)
- CRS server connects to publisher

- mechanism:

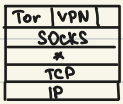
1. focus on link user-CRS: • mimicry (transform traffic so „looks" as whitelisted, content mimicry → high traffic volumes for censor; imitate protocol, introduce randomness → censors typically permit unknown traffic; flow mimicry)
• tunnelling • Covert channel (hide content), traffic manipulation
• destination obfuscation

2. publisher-CRS link: 1. content redundancy (stove content on multiple servers)
2. distributed content storage (chunks)

③ Tor (send data streams anonymously)

- communication: • directory servers • IP of directory authorities (DA) hardcoded into Tor client
- every hour DAs publish consensus on all available Tor relays (each compiles list of all known relays + flags, bandwidth, submits status-vote, collects votes, combines params and sends signature to other DAs
- majority of DA should agree on the data → consensus
- some tor relays not published by DAs: Tor bridges (Tor project maintains list of bridges and gives small list)

④ Scramblesuit/OBFS4*



- pseudo-random payload: traffic indistinguishable from randomness → no useful fingerprints for DPI
- polymorphic: flow characteristic change
- shared secret: server only speak protocol if client proves knowledge of secret

⑤ Domain fronting/MEEK

- domain frontig: using different domain names for diff. protocols in the stack
- not free • Google, Azure capable of this

⑥ Snowflake:

- idea: proxy traffic through tmp proxies over WebRTC
- volunteers run snowflake proxies(acts Tor bridge), broker manages and gives proxy addresses to clients
  - use domain fronting → high cost of blocking
  - only communication with broker is domain-fronted → lower cost compared to meek

⑦ Conjure

- idea: instead of hiding Tor traffic integrate in into network core
- IPS deploys Conjure station: client connect to phantom proxy: unused address in IPS address range • station injects packets as if there was true proxy available
- hard to block: censor need to block all websites served by this ISP
- PF_Ring: interface that allows conjure station to tap ISP traffic