

Composition, Verification, and Differential Privacy

Justin Hsu

University of Wisconsin–Madison

Lightning recap

Definition (Dwork, McSherry, Nissim, Smith (2006))

An algorithm is (ϵ, δ) -differentially private if, for every two adjacent inputs, the output distributions μ_1, μ_2 satisfy:

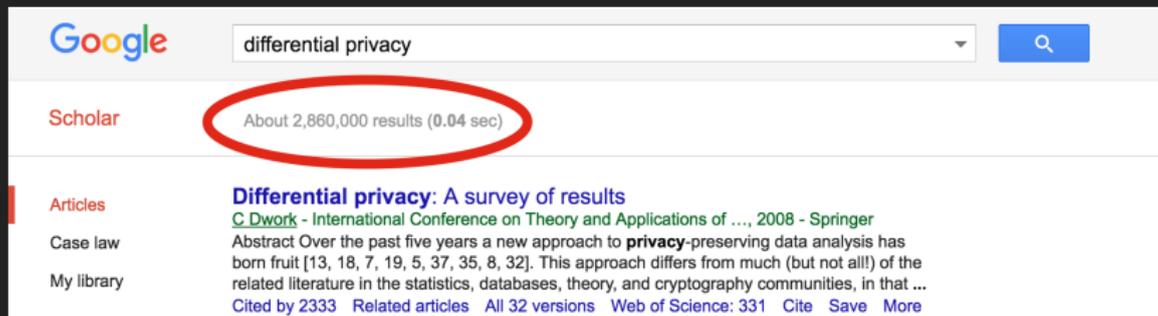
$$\text{for all sets of outputs } S, \Pr_{\mu_1}[S] \leq e^\epsilon \cdot \Pr_{\mu_2}[S] + \delta$$

Intuitively

Output can't depend too much on any **single** individual's data

Tremendous impact

Tremendous impact



The image shows a Google Scholar search interface. At the top left is the Google logo. To its right is a search bar containing the text "differential privacy". Further right is a blue search button with a magnifying glass icon. Below the search bar, the word "Scholar" is displayed in red. To its right, the search results are summarized as "About 2,860,000 results (0.04 sec)", which is circled in red. Below this, there are three categories listed on the left: "Articles", "Case law", and "My library". Under "Articles", there is a blue link for "Differential privacy: A survey of results" by C. Dwork, followed by a green link for "International Conference on Theory and Applications of ...". Below the article title is an abstract snippet starting with "Abstract Over the past five years a new approach to **privacy**-preserving data analysis has born fruit [13, 18, 7, 19, 5, 37, 35, 8, 32]. This approach differs from much (but not all!) of the related literature in the statistics, databases, theory, and cryptography communities, in that ...". At the bottom of the article snippet are several blue links: "Cited by 2333", "Related articles", "All 32 versions", "Web of Science: 331", "Cite", "Save", and "More".

Google

differential privacy

Scholar

About 2,860,000 results (0.04 sec)

Articles

Differential privacy: A survey of results

[C. Dwork](#) - International Conference on Theory and Applications of ..., 2008 - Springer

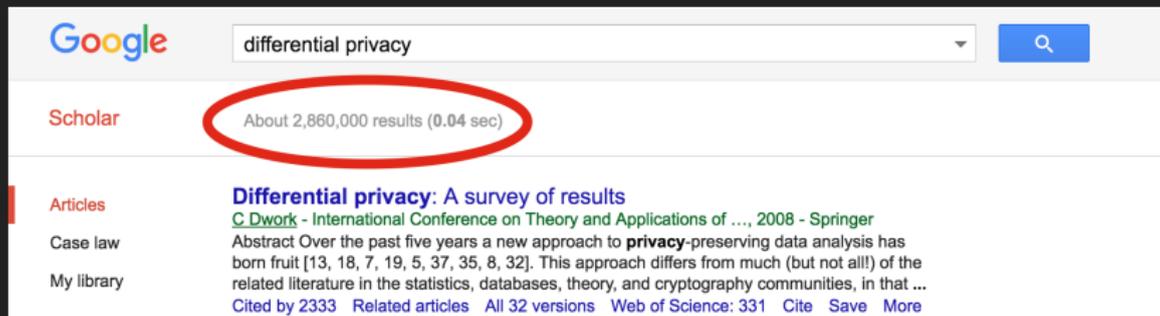
Case law

Abstract Over the past five years a new approach to **privacy**-preserving data analysis has born fruit [13, 18, 7, 19, 5, 37, 35, 8, 32]. This approach differs from much (but not all!) of the related literature in the statistics, databases, theory, and cryptography communities, in that ...

My library

[Cited by 2333](#) [Related articles](#) [All 32 versions](#) [Web of Science: 331](#) [Cite](#) [Save](#) [More](#)

Tremendous impact



The screenshot shows a Google Scholar search interface. The search bar contains the text "differential privacy". Below the search bar, the results are displayed. The first result is highlighted with a red oval and shows "About 2,860,000 results (0.04 sec)". Below this, there is a section for "Articles" with the title "Differential privacy: A survey of results" and a link to a paper by C. Dwork. The abstract of the paper is visible, discussing a new approach to privacy-preserving data analysis.

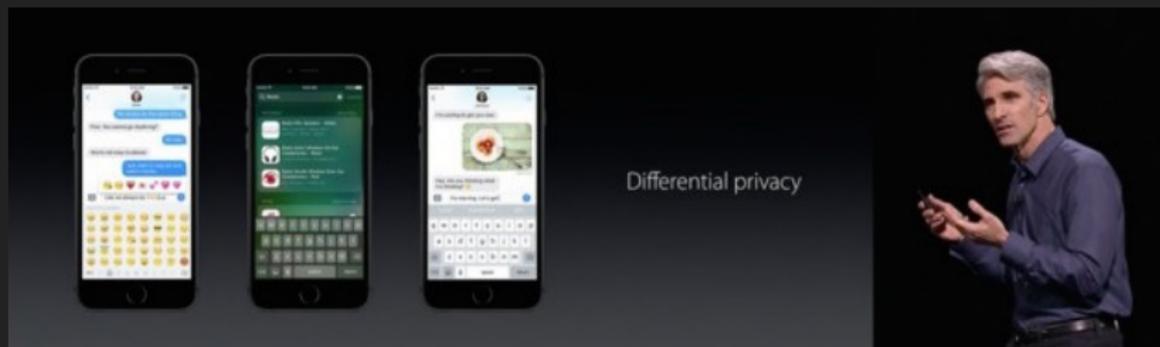
Google differential privacy

Scholar About 2,860,000 results (0.04 sec)

Articles **Differential privacy: A survey of results**
[C. Dwork](#) - International Conference on Theory and Applications of ..., 2008 - Springer

Case law Abstract Over the past five years a new approach to **privacy**-preserving data analysis has born fruit [13, 18, 7, 19, 5, 37, 35, 8, 32]. This approach differs from much (but not all!) of the related literature in the statistics, databases, theory, and cryptography communities, in that ...

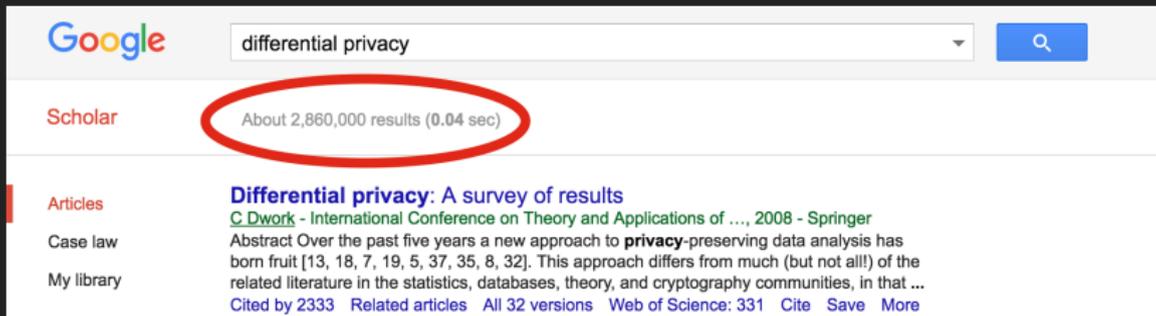
My library Cited by 2333 Related articles All 32 versions Web of Science: 331 Cite Save More



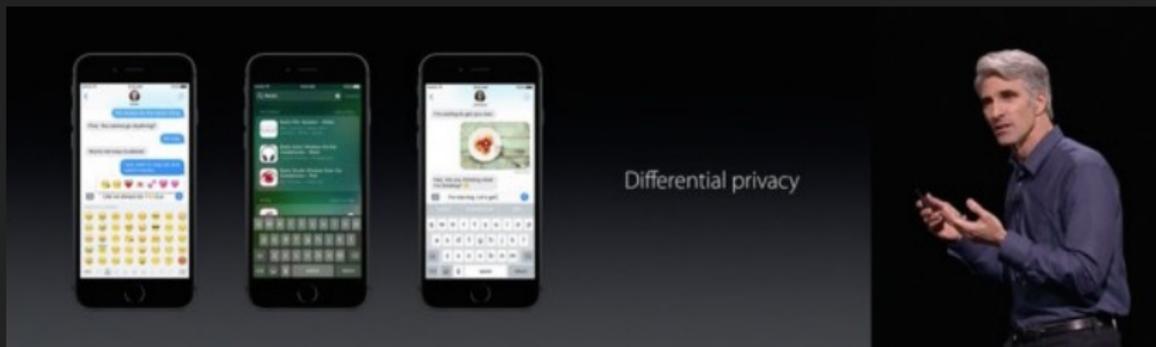
The slide features three smartphones on the left, each displaying a different app interface. The text "Differential privacy" is centered on the slide. On the right, a man in a blue shirt is gesturing with his hands, likely presenting the slide.

Differential privacy

Tremendous impact



A screenshot of a Google Scholar search for "differential privacy". The search bar shows "differential privacy" and a search icon. Below the search bar, the results are displayed. The "Scholar" section shows "About 2,860,000 results (0.04 sec)", which is circled in red. The "Articles" section shows a result titled "Differential privacy: A survey of results" by C. Dwork, from the International Conference on Theory and Applications of ..., 2008 - Springer. The abstract mentions that over the past five years, a new approach to privacy-preserving data analysis has emerged, differing from much of the related literature in the statistics, databases, theory, and cryptography communities. The article is cited by 2333, has 32 versions, and is listed in the Web of Science.



TPDP 2018 - Theory and Practice of Differential Privacy
Toronto, Canada - 15 October 2018 - part of CCS 2018

Why so popular? Elegant definition

Cleanly carve out a slice of privacy

- ▶ Mathematically formalize one kind of privacy
- ▶ “Your data” versus “data about you” (McSherry)

Simple and flexible

- ▶ Can establish property in isolation
- ▶ Achievable via rich variety of techniques

Why so popular? Theoretical features

Protects against worst-case scenarios

- ▶ Strong adversaries
- ▶ Colluding individuals
- ▶ Arbitrary side information

Rule out “blatantly” non-private algorithms

- ▶ Release data record at random: not private!

Above all, one reason...

Above all, one reason...

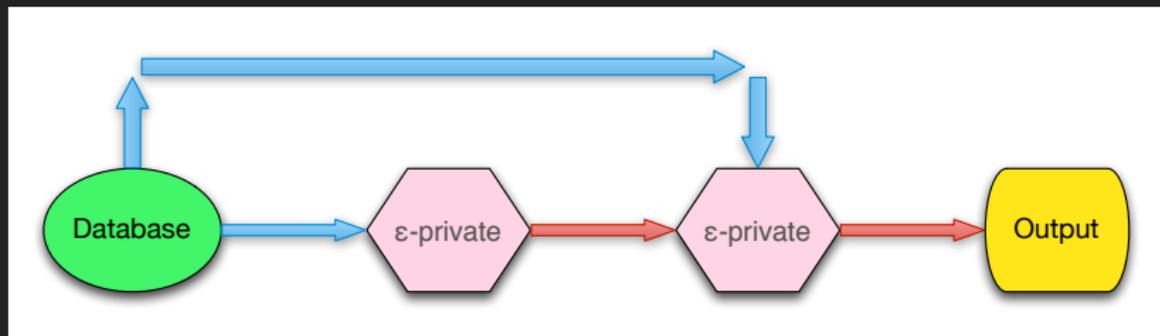
Composition!

Today

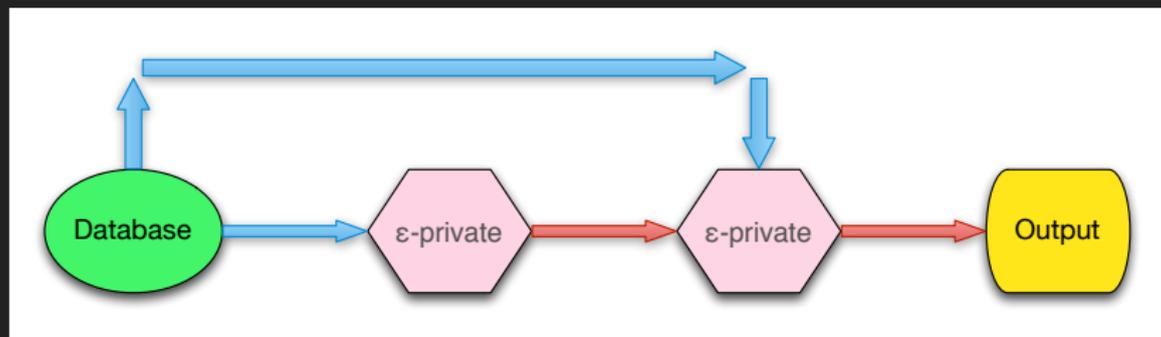
1. Review and motivate composition properties
2. Case study: formal verification for privacy
3. Case study: advanced composition

A Quick Review: Composition and Privacy

Sequential composition



Sequential composition



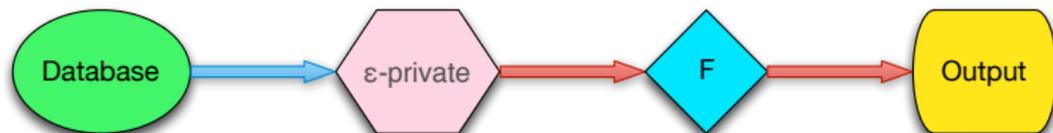
Theorem

Consider randomized algorithms $M : D \rightarrow \text{Distr}(R)$ and $M' : R \times D \rightarrow \text{Distr}(R')$. If M is (ϵ, δ) -private and for every $r \in R$, $M'(r, -)$ is (ϵ', δ') -private, then the composition

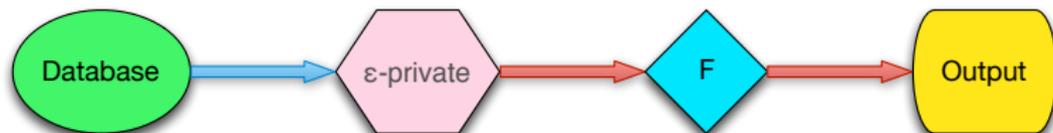
$$r \sim M(d); \text{out} \sim M'(r, d); \text{return}(\text{out})$$

is $(\epsilon + \epsilon', \delta + \delta')$ -private.

Example: post processing



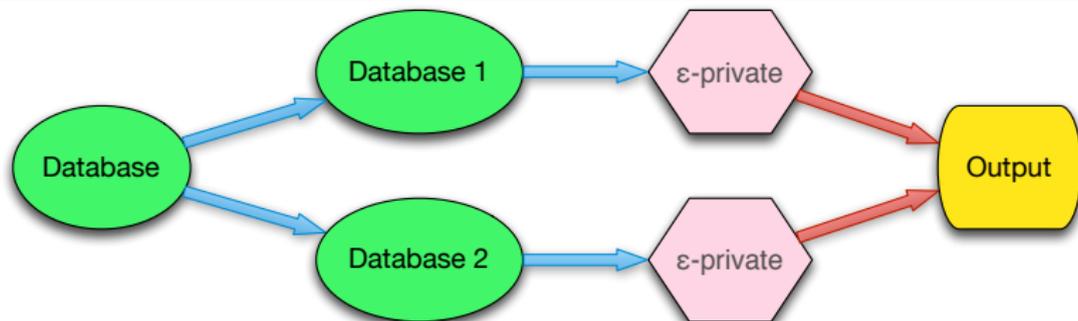
Example: post processing



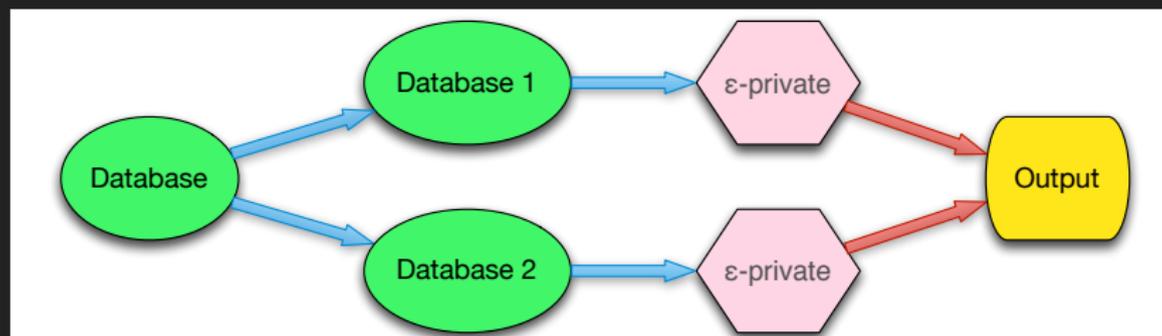
Privacy is preserved

- ▶ F is $(0, 0)$ -private: doesn't use private data
- ▶ Result is still (ϵ, δ) -private

Parallel composition



Parallel composition



Theorem

Consider randomized algorithms $M_1 : D \rightarrow \text{Distr}(R_1)$ and $M_2 : D \rightarrow \text{Distr}(R_2)$. If M_1 and M_2 are both (ϵ, δ) -private, then the parallel composition

$$(d_1, d_2) \leftarrow \text{split}(d); r_1 \sim M_1(d_1); r_2 \sim M_2(d_2); \text{return}(r_1, r_2)$$

is (ϵ, δ) -private.

Example: local differential privacy

Each individual adds noise

- ▶ Split data among individuals
- ▶ Each individual computation achieves privacy

Central computation aggregates noisy data

- ▶ Post-processing

Group privacy

Bound output distance when multiple inputs differ

- ▶ Inputs databases differ in one individual: $(\epsilon, 0)$ -privacy
- ▶ Inputs databases differ in k individuals: $(k\epsilon, 0)$ -privacy

Cast privacy as Lipschitz continuity

- ▶ Composes well
- ▶ Not so clean for (ϵ, δ) -privacy...

Why You Might Care About Composition

Make definitions easier to use

Easier to prove property

- ▶ Privacy proofs are often straightforward
- ▶ Don't need to unfold definition each time

More people can prove privacy

- ▶ Don't need years of PhD training

Increase re-usability

Dramatically increases impact

- ▶ One useful algorithm can enable many others
- ▶ Repurpose for new, unforeseen applications

Increase re-usability

Dramatically increases impact

- ▶ One useful algorithm can enable many others
- ▶ Repurpose for new, unforeseen applications

Key algorithms used everywhere

- ▶ Laplace, Gaussian, Exponential mechanisms
- ▶ Sparse vector technique
- ▶ Private counters
- ▶ Subsampling
- ▶ ...

Build larger algorithms

Scale up private algorithms

- ▶ Construct complex private algorithms out of simple pieces
- ▶ Composition ensures result is still correct

Enables common toolboxes

- ▶ PINQ framework (McSherry)
- ▶ PSI project (see Salil's talk)

Sign of a “good” definition

Not just about generalizing

- ▶ More general: must **assume less** about the pieces
- ▶ More specific: must **prove more** about the whole

Sweet spot between specific and general

- ▶ One way of probing robustness of definitions

Case Study: Verifying Privacy

Recap: verification setting

Dynamic

- ▶ Monitor program as it executes on particular input
- ▶ Raise error if it violates differential privacy

Static

- ▶ Take program (maybe written in special language)
- ▶ Check differential privacy on **all** inputs

Composition is crucial

Simplify verification task

- ▶ Trust a (small) collection of primitives
- ▶ Verify components separately

Enable automation

- ▶ Generally: enables faster/simpler verification
- ▶ So simple, a computer can do it

Privacy-integrated queries (PINQ)

C# library for private queries

- ▶ Proposed by Frank McSherry (2006)
- ▶ First verification technique for privacy

Dynamic analysis

- ▶ User writes PINQ query in C#
- ▶ Runtime monitors privacy budget as query runs

The Fuzz family of languages

History

- ▶ Reed and Pierce (2010), many subsequent extensions
- ▶ Programming language and custom type system

Main concept: function sensitivity

- ▶ Equip each type with a **metric**
- ▶ Types can express Lipschitz continuity

The Fuzz family of languages

History

- ▶ Reed and Pierce (2010), many subsequent extensions
- ▶ Programming language and custom type system

Main concept: function sensitivity

- ▶ Equip each type with a **metric**
- ▶ Types can express Lipschitz continuity

Example

$!_k \sigma \multimap \tau$ is type of a **k -sensitive function** from σ to τ

The Fuzz family of languages

Strengths

- ▶ Static analysis: don't need to run program
- ▶ Typechecking/privacy checking can be automated
- ▶ Can express sequential and parallel composition
- ▶ Captures kind of **group privacy** (e.g., $(\epsilon, 0)$ -privacy)

Weaknesses

- ▶ Can't verify programs where proof isn't from composition
- ▶ Have to use a custom programming language

The Fuzz family of languages

Recent developments: extending to (ϵ, δ) -privacy

- ▶ Idea: cast (ϵ, δ) -privacy as sensitivity property
- ▶ For inputs that are two apart, output distributions are (ϵ, δ) -related via some **intermediate** distribution
- ▶ So-called **path metric** construction
- ▶ Incorporate (ϵ, δ) -privacy into Fuzz framework

Privacy as an approximate coupling

History

- ▶ Arose from work on verifying cryptographic protocols via game-based techniques, comparing pairs of hybrids
- ▶ Target more familiar, imperative programming language

Main concept: prove privacy by constructing a coupling

- ▶ Consider program run on two adjacent inputs
- ▶ Approximately couple sampling instructions
- ▶ Establish relation between coupled outputs

Privacy as an approximate coupling

Strengths

- ▶ Static analysis: don't need to run program
- ▶ Can verify examples beyond composition
- ▶ Sparse vector, propose-test-release, ...
- ▶ No issue handling (ϵ, δ) -privacy

Weaknesses

- ▶ Checks proof automatically, but doesn't build proof
- ▶ Human expert must provide proof, manual process

Privacy as an approximate coupling

Recent developments: automate proof construction

- ▶ Encode proof requirement as a logical constraint
- ▶ Use techniques from program synthesis to find valid proofs
- ▶ Automatically verify sophisticated algorithms
- ▶ Sparse vector, report-noisy-max, between thresholds, ...

Brilliant collaborators



Case Study:

Advanced Composition

Recap: advanced composition

Sequentially compose k mechanisms

- ▶ Each (ε, δ) -private
- ▶ Basic analysis: result is $(k\varepsilon, k\delta)$ -private

Recap: advanced composition

Sequentially compose k mechanisms

- ▶ Each (ε, δ) -private
- ▶ Basic analysis: result is $(k\varepsilon, k\delta)$ -private

Better analysis

- ▶ Proposed by Dwork, Rothblum, and Vadhan (2010)
- ▶ For any δ' , result is $(\varepsilon', k\varepsilon + \delta')$ -private for

$$\varepsilon' = \varepsilon \sqrt{2k \ln(1/\delta')} + k\varepsilon(e^\varepsilon - 1)$$

Extremely useful, but seems a bit off...

Intuitively

- ▶ Slow growth of ϵ by increasing δ a bit more
- ▶ Privacy loss is “usually” much less than $k\epsilon$

Composition is not so clean

- ▶ Best bounds if applied to a block of k mechanisms
- ▶ Weaker if repeatedly applied pairwise

Improving the definitions: RDP and zCDP

History

- ▶ “Concentrated DP”: Dwork and Rothblum (2016)
- ▶ “Zero-Concentrated DP”: Bun and Steinke (2016)
- ▶ “Rényi DP”: Mironov (2017)
- ▶ Bound **Rényi divergence** between output distributions
- ▶ Refinement of (ϵ, δ) -privacy

Cleaner composition

Theorem (Mironov (2017))

Consider randomized algorithms $M : D \rightarrow \text{Distr}(R)$ and $M' : R \times D \rightarrow \text{Distr}(R')$. If M is (α, ϵ) -RDP and for every $r \in R$, $M'(r, -)$ is (α, ϵ') -RDP, then the composition

$$r \sim M(d); \text{out} \sim M'(r, d); \text{return}(\text{out})$$

is $(\alpha, \epsilon + \epsilon')$ -RDP.

Benefits

- ▶ Composing pairwise or k -wise: same bounds
- ▶ Closure under post-processing
- ▶ Improved formulation of advanced composition

Simplify reasoning

Enable formal verification

- ▶ Extensions of techniques for imperative languages
- ▶ Also works for programs in functional languages
- ▶ Opens the way to automated proofs

Wrapping Up

Success of privacy is a success of composition

Key factor behind high interest

- ▶ Make proofs easy enough for all
- ▶ The world has only so many TCS researchers
- ▶ Trivial to adapt privacy to new applications
- ▶ Ancillary benefit: enable computer verification

Composition matters!

Often not easy, but...

- ▶ Difference between a theoretically interesting definition, and a practically usable one
- ▶ Worth extra work and trouble to achieve

Compare to situation in cryptography

- ▶ Immense need for this technology, but poor composition
- ▶ Implementation still tricky, subtle errors
- ▶ “Don’t roll your own cryptography”

Trend towards “formal engineering”

Security is too hard for humans

- ▶ Want formal guarantees from our systems
- ▶ Rule out classes of attacks (subject to assumptions...)
- ▶ Principled construction of safe software

Compositional definitions are critical to this vision

- ▶ Needed to reason about large systems
- ▶ Only way to manage complexity

As I once heard from a famous systems researcher...

As I once heard from a famous systems researcher...

Without modularity,
there is no civilization.

As I once heard from a famous systems researcher...

Without modularity,
there is no civilization.

(Or at least, the going is pretty tough.)

Composition, Verification, and Differential Privacy

Justin Hsu

University of Wisconsin–Madison