Accountable Algorithms

Edward W. Felten Joshua Kroll

Center for Information Technology Policy
Dept. of Computer Science
Princeton University

Fairness, Accountability and Transparency in ML Workshop, NIPS 2014
December 12, 2014

Accountability != Transparency

Example: Tax Audit

Example: Tax Audit

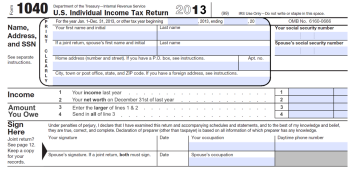
subject

Example: Tax Audit

subject

agency

Example: Tax Audit



1040 U.S. Individual Income Tax Return 2013

Name, Address, and SSN

Income

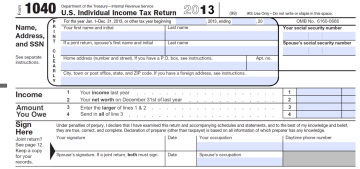
Amount You Owe

Sign Here

subject

agency

Example: Tax Audit

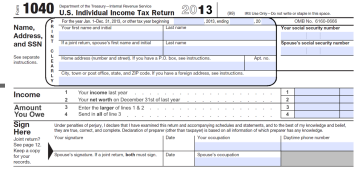


“We’re auditing you.”

subject

agency

Accountable Tax Audit



Form 1040 U.S. Individual Income Tax Return 2013

Name, Address, and SSN

Income

Amount You Owe

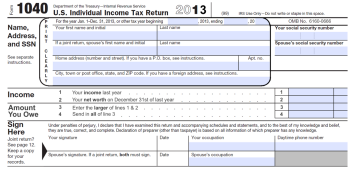
Sign Here

“We’re auditing you.”

subject

agency

Accountable Tax Audit



Form 1040 U.S. Individual Income Tax Return 2013

Name, Address, and SSN

Income

Amount You Owe

Sign Here

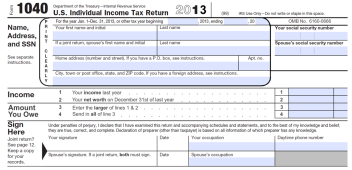
“We’re auditing you.”

proof

subject

agency

Accountable Tax Audit



subject

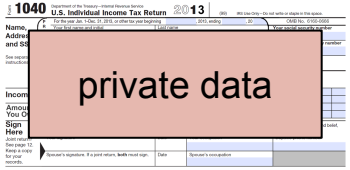
“We’re auditing you.”

proof

agency

overseer

Accountable Tax Audit



"We're auditing you."

proof

subject

agency

overseer

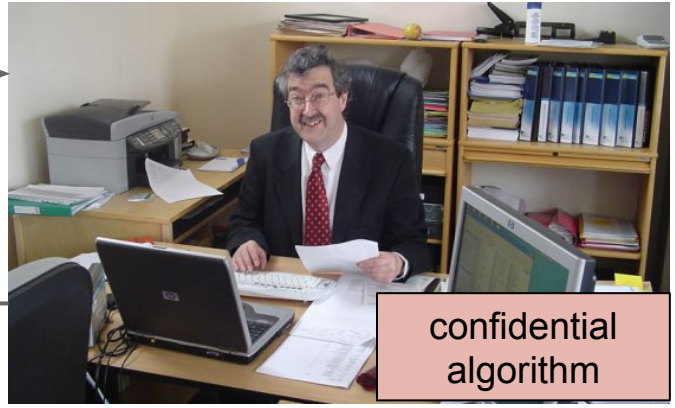
Accountable Algorithms

subject

private data

Decision

proof



agency

overseer

Accountable Algorithms: Examples

- Tax audit
- School admission lottery
- Eligibility for housing, credit, ...
- Scoring / predictive decisionmaking
- Search at airport security
- Immigration lottery
- Voting

Guarantees

Subject verifies that:

- Agency committed to algorithm in advance
- Agency used same algorithm for everyone
- Correct subject-specific data was used
- Randomness was fairly generated

Overseer determines whether:

- Algorithm is lawful, or fair, or good policy

Role of the Overseer

Overseer: court, executive, or legislature

Can rule on algorithm's suitability, after the fact

Don't have a specification

Model as an oracle, can consult only rarely

Confidentiality + Accountability

Key idea:

Advanced crypto lets us “square the circle” and keep information confidential while proving it has certain properties.

Goal: Build a working prototype.

The Protocol: Setting

- Parties: *Agency*, *Subjects*, *Overseer*
- *Agency*'s *policy* is divided into a public *function* f and a secret *parameter* y.
  - Decision for subject i is $z_i = f(y, x_i)$ for each subject's data x_i
- *Agency* writes signed protocol output into a *log* (visible to all parties) without interaction by any *subject*.

The Protocol

- Protocol proceeds in 4 epochs:
  - Commit
  - Randomness
  - Decision
  - Oversight (optional)

The Protocol

agency

The Protocol: Normal Operation

log

agency

signed protocol output

subject

Decision

The Protocol: Commit

log

agency

ZK-SNARK Public Parameters
Decision Algorithm
Commit(**Agency input**)
Commit(**Privacy randomness**)

The Protocol: Randomness

log

Random Process (e.g. Beacon, Ceremony, Protocol) output

agency

The Protocol: Decision

log

agency

Commit(**Decision**)
Commit(**subject-specific input**)
Proof - **ZK-SNARK**

subject

Decision

z - **Decision**
r2 - **Commitment Randomness**

The Protocol: Decision

log

agency

Commit(**Decision**)
Commit(**subject-specific input**)
Proof - **ZK-SNARK**

subject

Decision

z - **Decision**
r2 - **Commitment Randomness**

The Protocol: Oversight (optional)

subject

proof

overseer

log

Major Tools

1. Cryptographic Commitments
2. ZK-SNARKs

Cryptographic Commitments

A pair of functions

- $C_x \leftarrow$ **Commit**(x, r) [r: random]
- b $\leftarrow$ **Verify**(x, r, C_x)

Such that

- C_x leaks negligible information about x
- C_x is *computationally binding*

ZK-SNARKs

Zero-knowledge succinct non-interactive arguments of knowledge

Given arithmetic circuit with bilinear gates **C**(**x**,**a**), taking *input* **x** and *witness* **a**

Can prove knowledge of **a** such that **C**(**x**,**a**) = 1. [BCTV14]

Proof is:

- non-interactive
- zero knowledge w.r.t. **a**
- succinct:
  - small constant space (~300 bytes)
  - small, constant time to check (~ 5 ms)

[BCTV14] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, "Succinct Noninteractive Arguments for a von Neumann Architecture". USENIX Security 2014.

ZK-SNARKs

SNARKs exist in a *preprocessing model*

- First, derive from the circuit **C**:
  - a public *proving key* pk, and
  - a public *verification key* vk.
- From these, anyone can generate and verify proofs.
- Must trust the party who generates (pk, vk).

How We Use ZK-SNARKs

- **x**: public information (value)
- **a**: secret information (witness)
- **C:** agency circuit
- $\mathbf{C_A}(\mathbf{x},\mathbf{a})$ verifies **x**, **a**, and result “match up”
  - prover knows how to unwrap commitments, and
  - committed-to values lead to asserted decision
- Zero-knowledge w.r.t. secret information
  - don’t reveal committed-to values
  - can later reveal selectively to appropriate parties

Extensions

1. Add fairness predicates to $\mathbf{C_A}$. Can prove any predicate on agency's secret policy, as part of the zk-SNARK.

2. Less intrusive disputes: structure commitment to subject's private data, so subject can raise dispute without revealing all of subject's private data

Implementation

- Building a prototype, based on [BCTV14] *libsnark* code for generic ZK-SNARKs.

- Implemented generic $\mathbf{C_A}$;
  experimenting with example applications.

Example Applications

Classifier

Scoring algorithm

Lottery for immigration, school admission, etc.

Perspective

Key challenge:
interface between computer science and real policymaking

Computer Science: want complete, precise spec to enforce
Policymaking: compromise-driven,
ex post evaluation of specific policies

What Computer Science Can Do

enforce procedural fairness

- policy chosen in advance
- same policy for everyone
- fair randomness

enforce “easy parts” of substantive fairness

enable well-informed oversight

Fairness vs. Accountability

Fairness is the ideal.

Accountability is more achievable.

Accountable Algorithms

Edward W. Felten Joshua Kroll

Center for Information Technology Policy
Dept. of Computer Science
Princeton University

Fairness, Accountability and Transparency in ML Workshop, NIPS 2014
December 12, 2014

The Protocol (in handy chart form)

	Agency Writes to Log	Agency Knows Secretly	Agency sends confidentially to each subject
Commit	Ψ - **ZK-SNARK Parameters** f - **Decision Algorithm** Commit(y - **Agency input**) Commit(ϱ - **Privacy randomness**)	y - **Agency input** ϱ - **Privacy Randomness** r1 - **Commitment Randomness**	
Randomness	b - **Beacon output**		
Decision	Commit(z - **Decision**) Commit(x- **subject-specific input**) Π - **ZK-SNARK**	z = f(x, y, b $\oplus$ ϱ) - **Decision** x - **Subject input** r2 - **Commitment Randomness**	z - **Decision** r2 - **Commitment Randomness**