Browser Randomization against Fingerprinting: A quantitative information flow approach

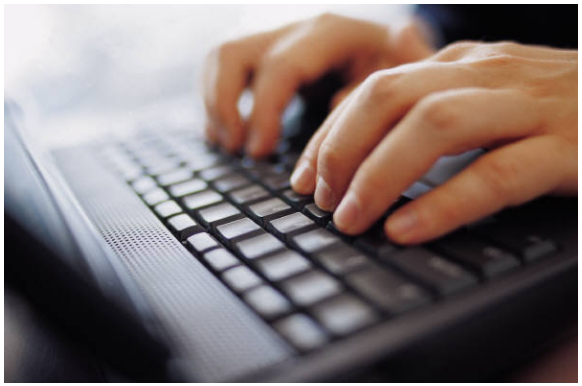
Frederic Besson, **Nataliia Bielova** and Thomas Jensen

published at NordSec 2014

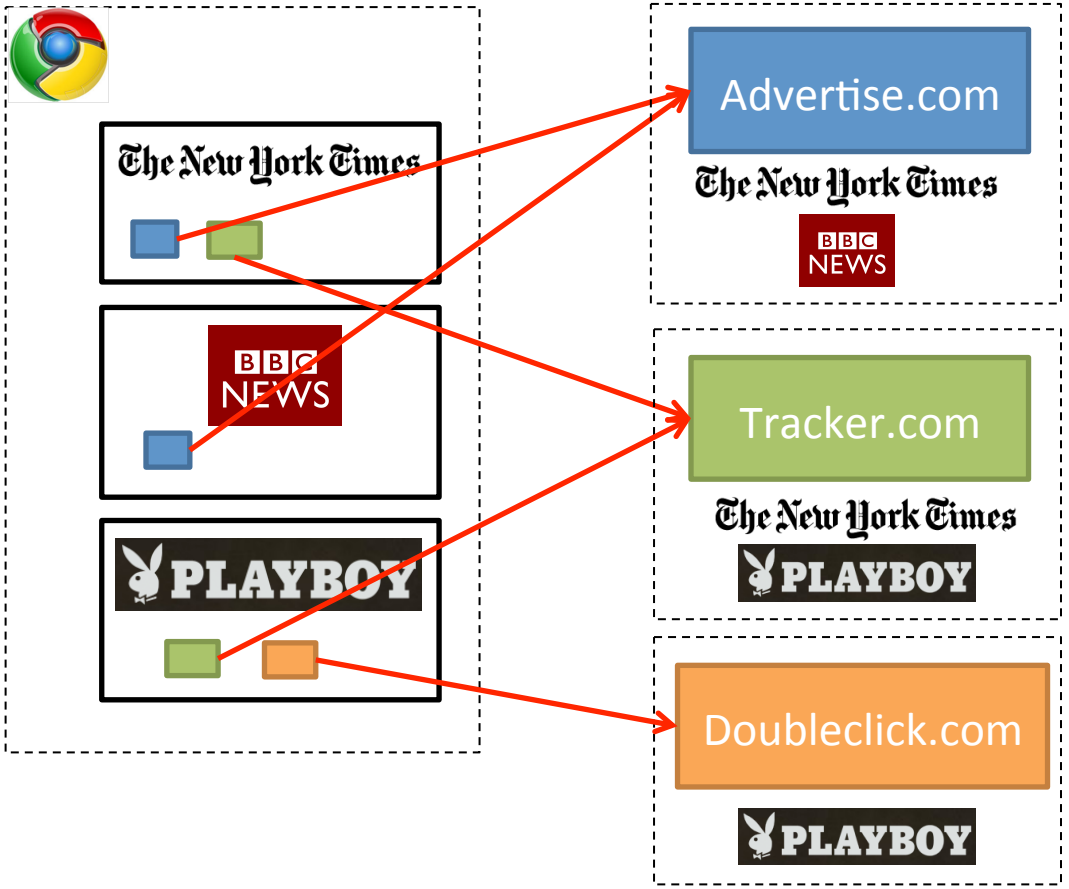
Quantitative Information Flow Day (PRINCESS workshop)

16 December 2014

Web Tracking



Bigger browsing profiles
= increased value for trackers
= reduced privacy for users



(Hypothetical tracking relationships only.)

Courtesy of Franziska Roesner

Web Tracking and Price discrimination

Are you REALLY getting the best deal? Research reveals online customers are victims of 'price discrimination' when booking their holidays

- It often pays to sign up as a 'member' to receive the best deals
- Expedia 'steer' users to hotels and prices dependent on browser history
- Hotel prices can vary dependent on what device you're using
- This kind of 'price discrimination' is not illegal - it pays to be clued up

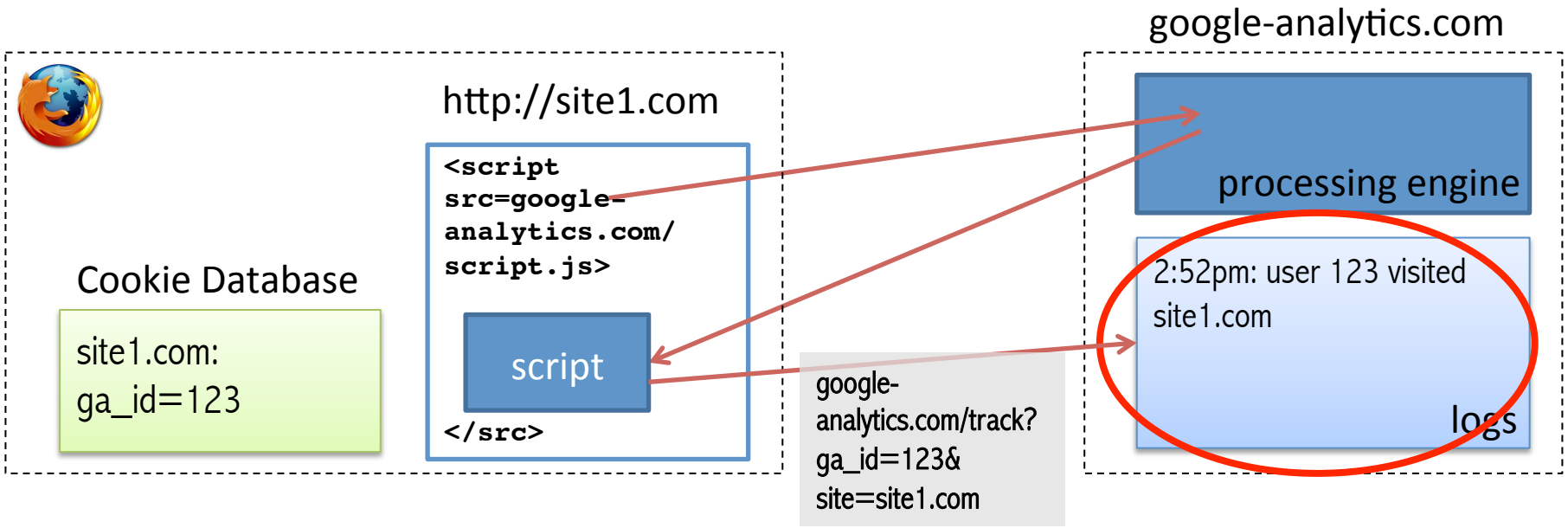
By JOHN HUTCHINSON FOR MAILONLINE

PUBLISHED: 13:08 GMT, 29 October 2014 | **UPDATED:** 13:11 GMT, 29 October 2014

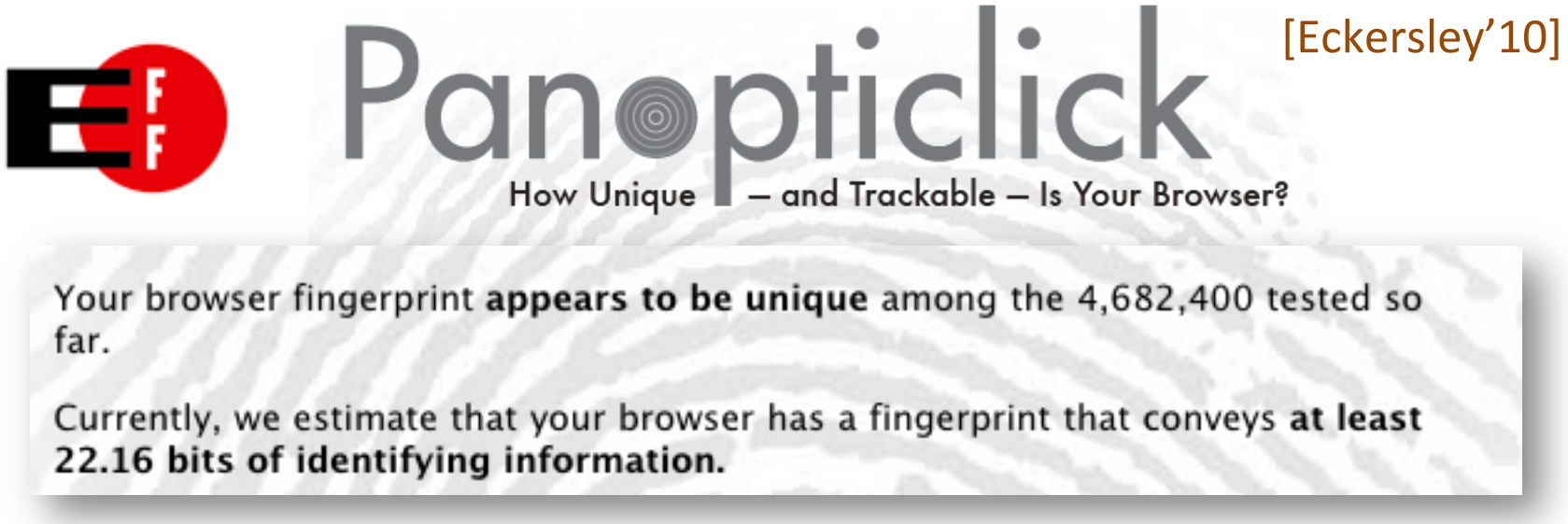
From www.dailymail.co.uk

Tracking by storing identity

Cookies are used to track repeated visits to a site.



Tracking by creating identity



[Eckersley'10]

- **Idea: distinguish** users **by browser fingerprints**:
  - HTTP headers
  - Browser and OS features: language, plugins, fonts, screen, …

The most identifying features (via JavaScript and Flash)

How can we protect users?

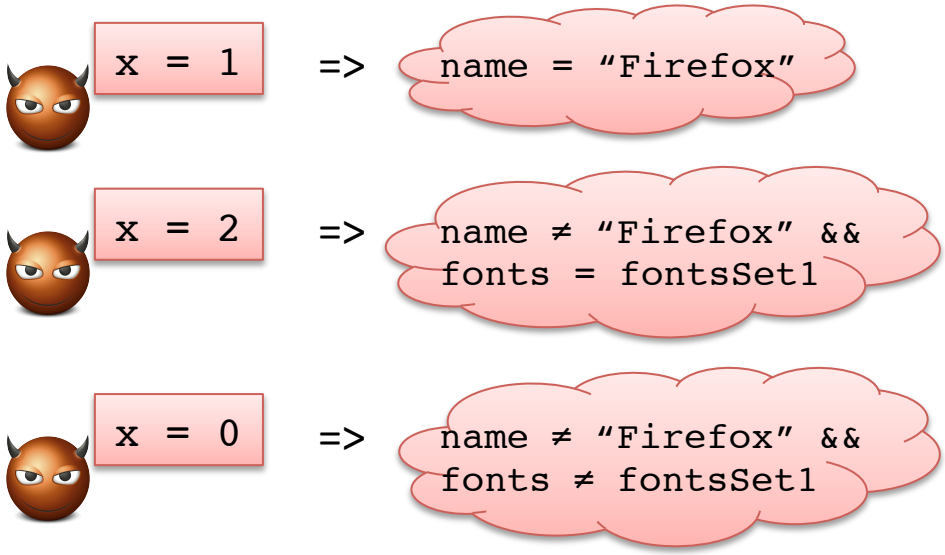
- Storing identity: well-known and getting addressed
  - Third-party cookies blocking
  - Non-interference for JavaScript
  - EU e-Privacy directive

  [Austin, Flanagan 12]
  [De Groef et al. 12]
  [Hedin, Sabelfeld 12]

- Creating identity: not addressed
  - IP address tracking
  - Web browser fingerprinting 

What does tracker learn?

```
var x = 0;
if (name == "Firefox") {
    x = 1;
}
else {
    if (fonts == fontsSet1) {
        x = 2;
    }
}
output x;
```



Depending on user's browser, **different executions** of the same script **leak different quantity** of information!

[Besson, Bielova, Jensen CSF'2013]

Hybrid Info Flow Monitoring

- Dynamic environment: $env: Var \rightarrow Val$
- Static constant propagation: $env: Var \rightarrow Val \cup \{\top\}$

```
var x = 1;          env(x) = 1
var y = fonts;      K(y): fonts = fontsSet

if (name == "Firefox") {
    x = 1;          env(x) = 1    K'(x): tt
}
else {
    if (y != fontsSet) {
        x = 2;
    }               env(x) = 1
}
output x;           K(x)
```

Values are the same after both branches

Dynamic		Static
$env(x) = 1$	=	$env(x) = 1$

New knowledge in x after branching

$(name = \text{“Firefox”} \Rightarrow K'(x)) \wedge$

$(name \neq \text{“Firefox”} \Rightarrow K'(x))$

Knowledge in x from non-executed branch computed by static analysis

How to enforce anonymity?

- Our hybrid monitor evaluates how much a tracker learns for a concrete user

Challenge:

Which mechanism can **provably guarantee** that **every user is protected** from being tracked?

```
p(name) =
     Firefox  0.45
     Chrome   0.45
     Opera    0.10
```

- 10 users: Opera user will be uniquely identified
- Halting the program (or suppressing output x=A) will still make the user uniquely identified
- The other users may help to hide the unique user…

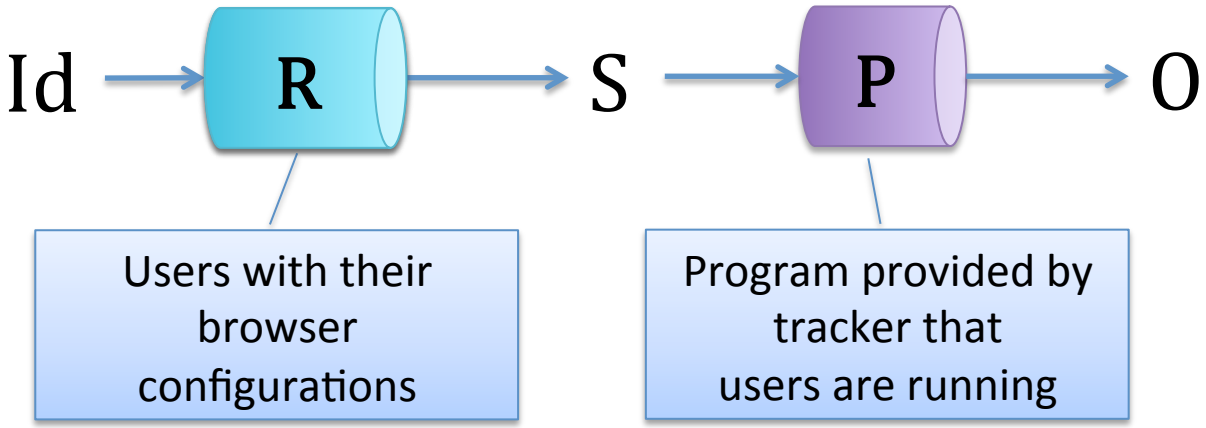
Our solution

- Users continuously switch between configurations

- In theory…
  - How many configurations needed?
  - How often they need to switch?

- But in practice…
  - Users don't want change their habits
  - They prefer switch very rarely

Our ingredients



- Find $\mathbf{R}$: a distribution of configurations for all users such that:
  - User privacy is protected (soundness)
  - Usability is maximized

U: users & their current configurations

	{Chrome/32.0.1700.77, Intel Mac OS X 10_8_5, {Arial, Arial Black, …}, {Chrome PDF Viewer, DivX Web Player v.1.4, …}, …}	{Firefox/26.0, Intel Mac OS X 10.8, {}, {Google Talk Plugin, QuickTime Plug-in 7.7.1, …}, …}	…
id_1	1	0	0
id_2	1	0	0
…	…	…	…
id_n	0	1	0

R: randomized configurations

	{Chrome/32.0.1700.77, Intel Mac OS X 10_8_5, {Arial, Arial Black, …}, {Chrome PDF Viewer, DivX Web Player v.1.4, …}, …}	{Firefox/26.0, Intel Mac OS X 10.8, {}, {Google Talk Plugin, QuickTime Plug-in 7.7.1, …}, …}	…
id_1	x_{11}	x_{12}	x_{1m}
id_2	x_{21}	x_{22}	x_{2m}
…	…	…	…
id_n	x_{n1}	x_{n2}	x_{nm}

P: program that users run

- Deterministic programs

```
if (name == "Opera") x = A;
else x = B;
output x;
```

p(o | s)	A	B
{Firefox/26.0, Intel Mac OS X 10.8, …}	0	1
{Chrome/32.0.1700.77, Intel Mac OS X 10_8_5, …}	0	1
{Chrome/32.0.1700.77, Windows7…, …}	0	1
{Opera…}	1	0

- We also support probabilistic programs

Soundness: Vulnerability?

```
if (name == "Opera") x = A;
else x = B;
output x;
```

A priori distribution

p(i) =	Firefox	0.45
	Chrome	0.45
	Opera	0.10

Attacker observes

$$x = A$$

A posteriori distribution

p(i|A)=	Firefox	0
	Chrome	0
	Opera	1

- Probability of guessing the secret given an observation:

$$\max_i p(i|A) = 1$$

Soundness: Vulnerability?

```
if (name == "Opera") x = A;
else x = B;
output x;
```

A priori distribution

p(i) =	Firefox	0.45
	Chrome	0.45
	Opera	0.10

Attacker observes

$$x = B$$

A posteriori distribution

p(i|B)=	Firefox	0.5
	Chrome	0.5
	Opera	0

- Probability of guessing the secret given an observation:

$$\max_i p(i|A) = 1$$
$$\max_i p(i|B) = 0.5$$

- Probability of guessing the secret in one try (aka average posterior vulnerability):

$$P^{aver}(C) = p(A) \max_i p(i|A) + p(B) \max_i p(i|B)$$
$$= 0.1 * 1 + 0.9 * 0.5 = 0.55$$

But the secret is leaked completely when x = A!

Soundness: probability of guessing

(aka worst-case posterior vulnerability [Espinoza, Smith 2013])

```
if (name == "Opera") x = A;
else x = B;
output x;
```

A priori distribution

p(i) =	Firefox	0.45
	Chrome	0.45
	Opera	0.10

Worst-case observation

$x = A$

Worst-case
a posteriori distribution

p(i|A)=	Firefox	0
	Chrome	0
	Opera	1

- Probability of guessing the secret in case of worst observation:

$$P^G(C) = \max_{i,o} p(i|o) = p(\text{Opera} \mid A) = 1$$

Very biased towards the worst output, but provides a strong guarantee

Soundness: Probability of guessing

Definition (Threshold-based privacy)

A channel C is t-private if the probability of guessing channel's input is bounded by t:

$$P^G(C) \leq t$$

- How to achieve t-privacy if $P^G(U \bullet P) > t$?
- Find a randomized user channel R, s.t.:

$$P^G(R \bullet P) \leq t$$

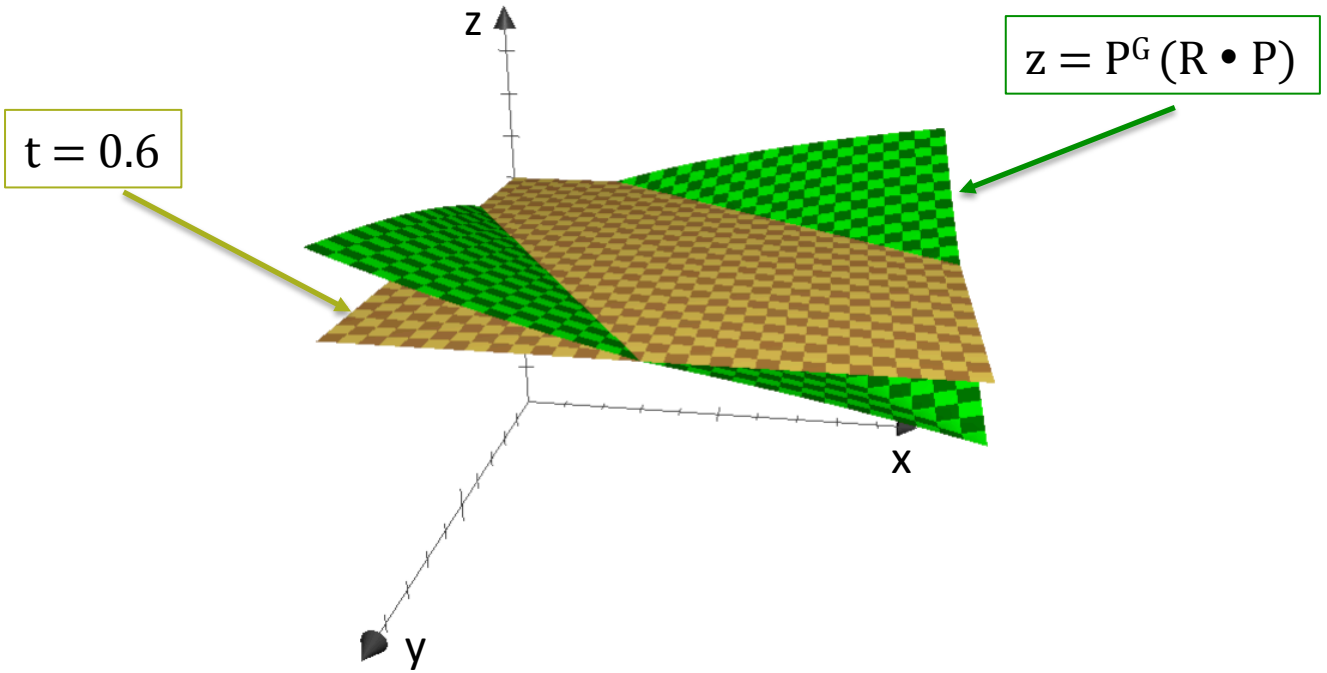
amounts to solving

$$\max_{i \in I, o \in O} \frac{\sum_{s \in S} x_{is} \cdot P[s,o]}{\sum_{j \in I} \sum_{s \in S} x_{js} \cdot P[s,o]} \leq t.$$

Building a sound R channel

R	"Firefox"	"Opera"
id_1	x	1-x
id_2	1-y	y

P	o1	o2	o3
"Firefox"	1/2	1/3	1/6
"Opera"	1/6	1/2	1/3



Usability: requirement for user satisfaction

U	"Firefox"	"Opera"
id_1	1	0
id_2	0	1

R	"Firefox"	"Opera"
id_1	x	1-x
id_2	1-y	y

Usability: (x+y) is maximized

Definition (Usability)

Given a user channel U, the randomized user channel R ensures that the users get their original configuration as much as possible:

$\Sigma_i R[i, Im(U,i)]$ reaches its maximum value

where $Im(U,i) = o$ if and only if $U[i,o] = 1$

Reduction to Linear Programming

U	"Firefox"	"Opera"
id_1	1	0
id_2	0	1

R	"Firefox"	"Opera"
id_1	x	1-x
id_2	1-y	y

Usability:
$\max \Sigma_i R[i, Im(U,i)] = \max (x+y)$

$$\max (x + y) \text{ s.t.}$$
$$0 \leq x \leq 1$$
$$0 \leq y \leq 1$$
$$Ax + By + C \leq t$$

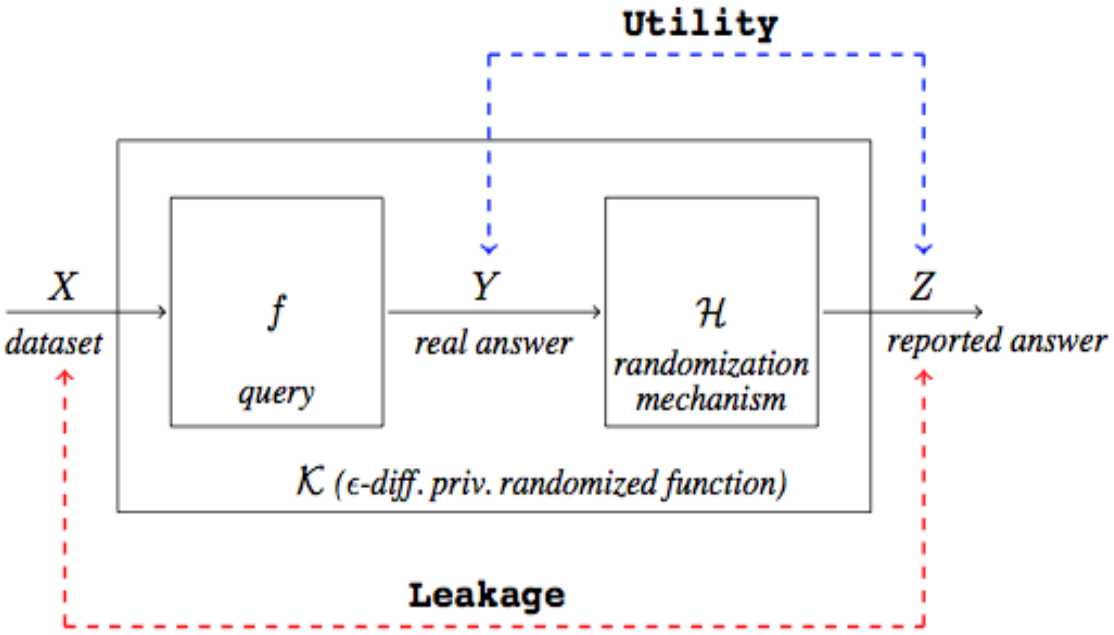
Soundness:
$P^G (R \bullet P) = Ax + By + C \leq t$

Enforcement algorithms

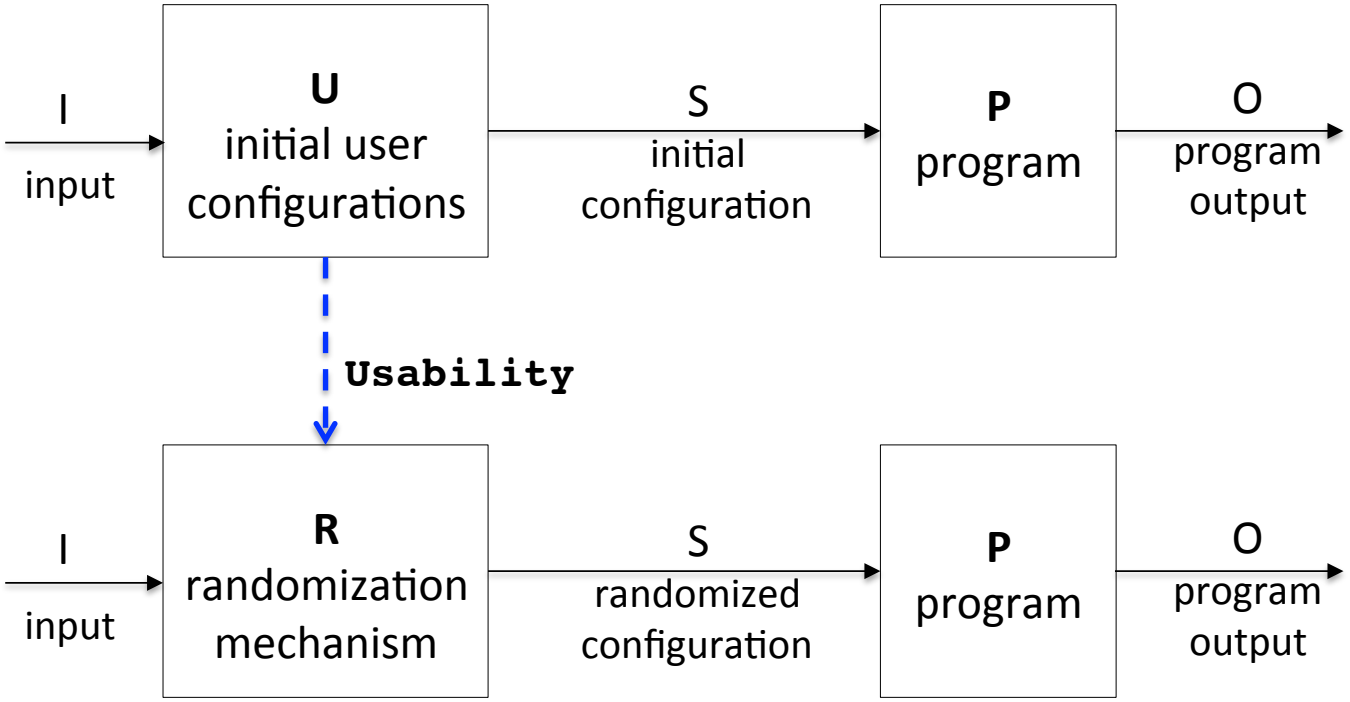
- Protect against one program P
  - Naïve global optimization costly
  - Reduce the number of variables:
    - unite “identical users”
    - exclude “safe users”
- Greedy algorithm

- Protection against any program P.

Is usability related to utility in differential privacy?



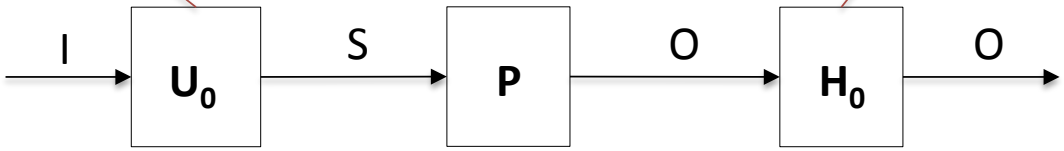
Is usability related to utility in differential privacy?



U_0	s_1	s_2	…
i_1	1	0	0
i_2	1	0	0
…	0	1	0

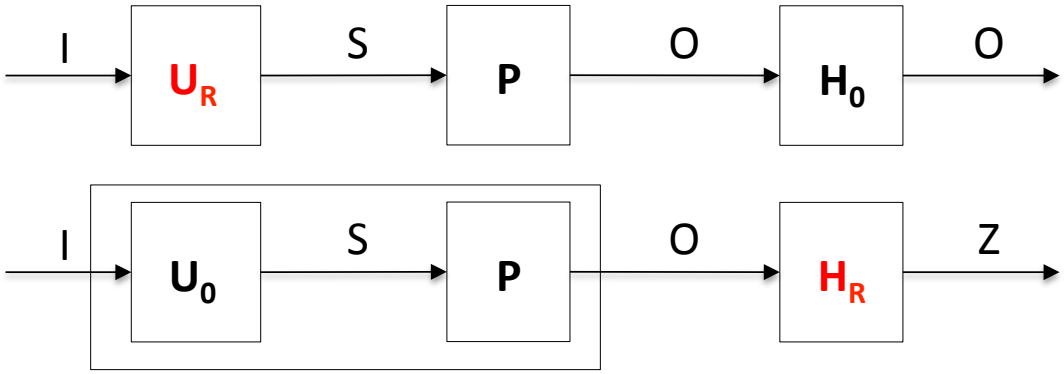
H_0	z_1	z_2	…
o_1	1	0	0
o_2	0	1	0
…	0	0	1

Initial system:



Given an input randomization $\mathbf{U_R}$ and an output randomization $\mathbf{H_R}$ such that $\mathbf{U_R \bullet P \bullet H_0 = U_0 \bullet P \bullet H_R}$ the following holds:

Usability($U_R \bullet P \bullet H_0$) = Usability($U_0 \bullet P \bullet H_R$) = Utility (O, Z) ?



Summary

- **Web tracking** is done by different technologies
  - cookies, other browser storages, fingerprinting
- **Analysis of tracking scripts**
  - By hybrid information flow monitoring
  - Computes a tracker's knowledge
  - Monitor made more precise with static analysis
- **Enforcing browser anonymity**
  - Systematic switching between browser configurations
  - Soundness: t-privacy for every user
  - Usability: users switch to other configurations as rare as possible