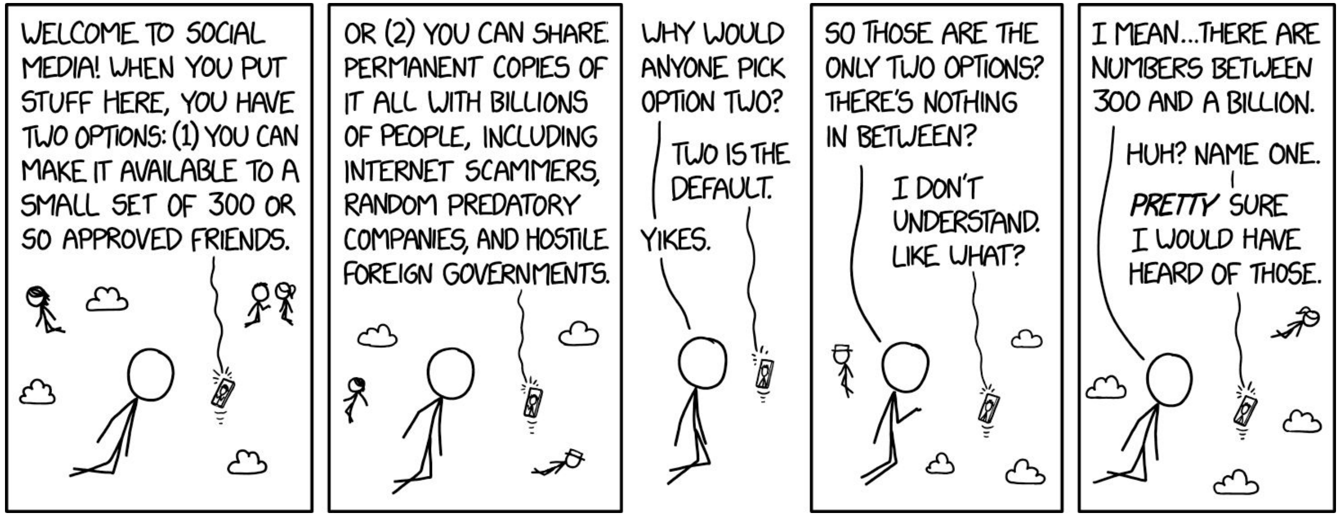
Διάλεξη #21 - Privacy, Policy and More

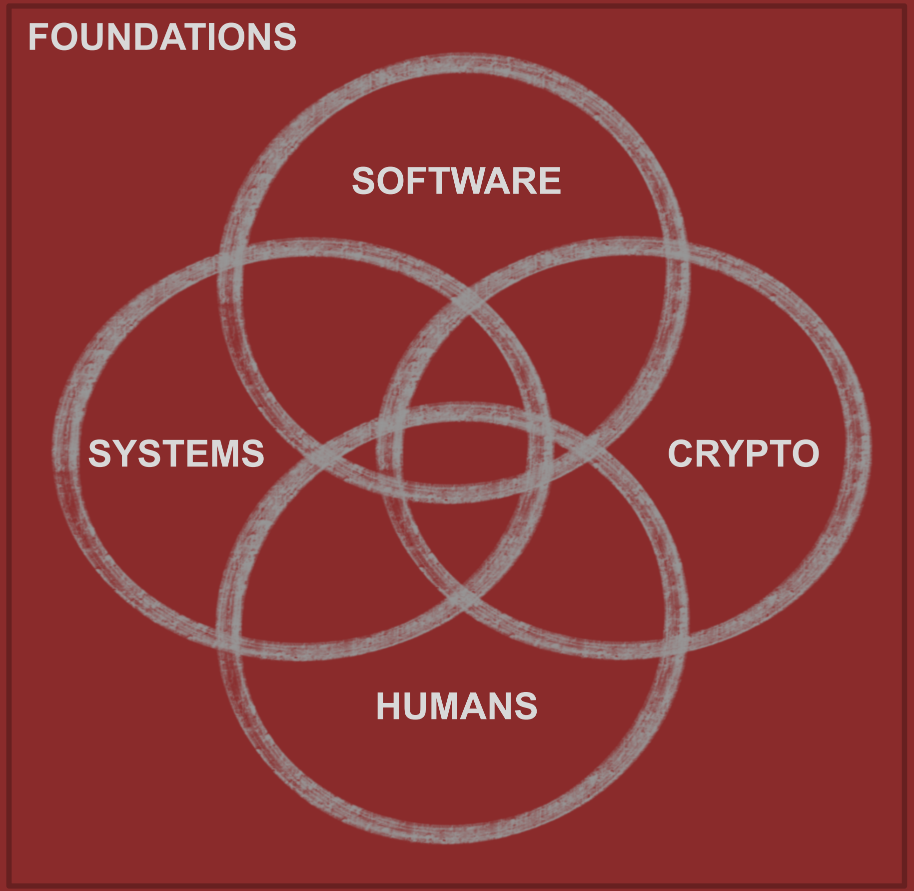
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Εισαγωγή στην Ασφάλεια

Θανάσης Αυγερινός



Huge thank you to [David Brumley](#) from Carnegie Mellon University for the guidance and content input while developing this class!



Ανακοινώσεις / Διευκρινίσεις

- Summer time!

Την προηγούμενη φορά

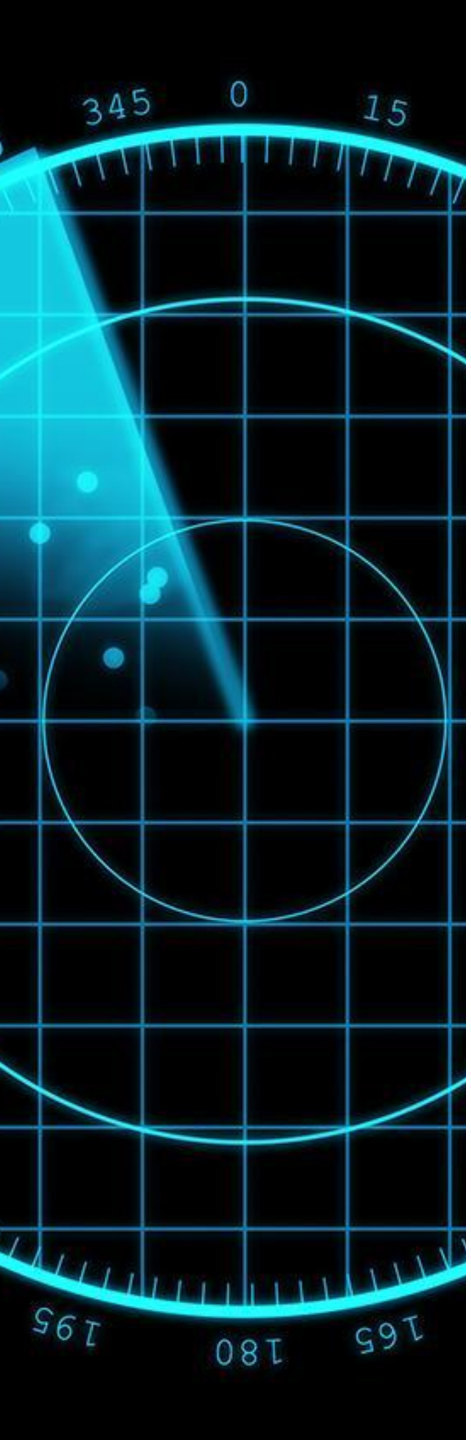
- Networks 101
- Scanning
- Firewalls

Σήμερα

- Base rate fallacy
- A few ideas
  - Anonymity
  - TPMs
  - Verification
- What we saw this year

Detection Theory

Lies, Damn Lies, and Statistics



Detection theory or **signal detection theory** is a means to measure the ability to differentiate between *information-bearing* patterns and *random* patterns that distract from the information (called noise). In the field of electronics, the separation of such patterns from a disguising background is referred to as ***signal recovery.***

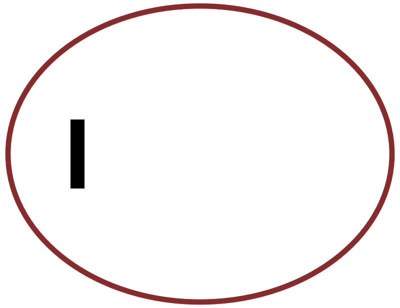
Ω

Let Ω be the set of all possible events.
For example:

- Audit records produced on a host
- Network packets seen

Ω

Example: IDS Received 1,000,000 packets.
20 of them corresponded to an intrusion
The *intrusion rate* Pr[I] is:
Pr[I] = 20/1,000,000 = .00002

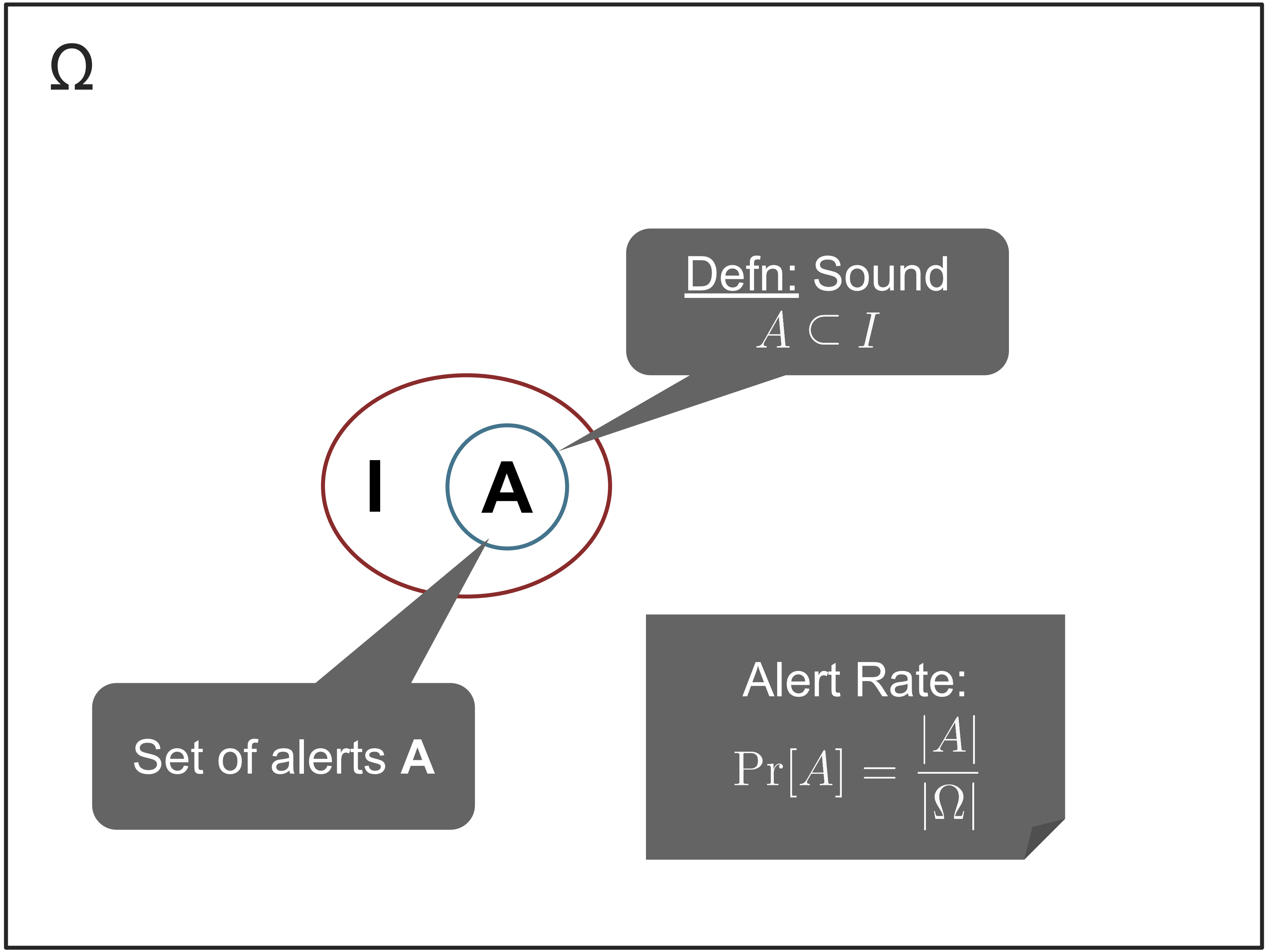


Set of intrusion events I

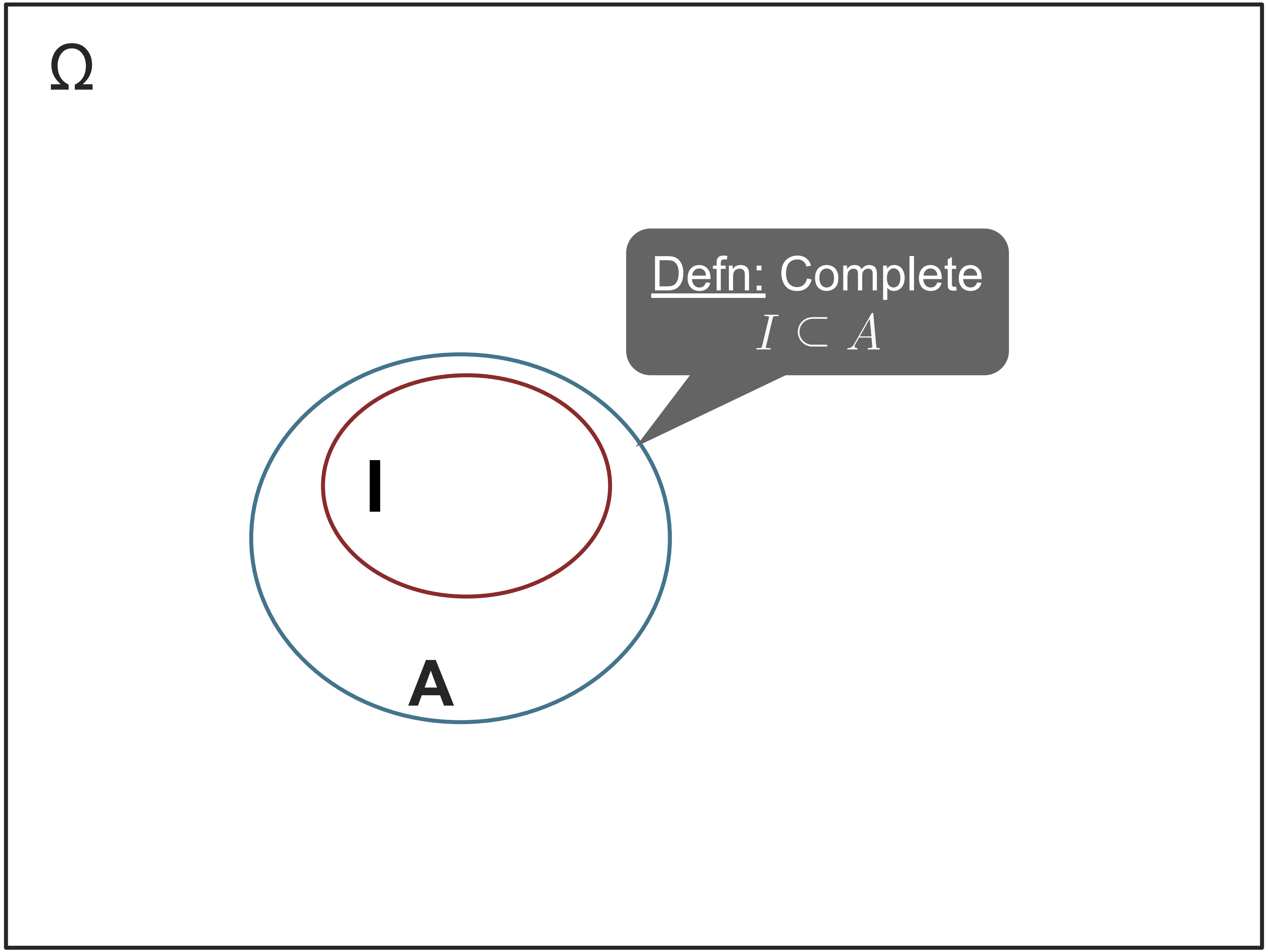
Intrusion Rate:

$$\Pr[I] = \frac{|I|}{|\Omega|}$$

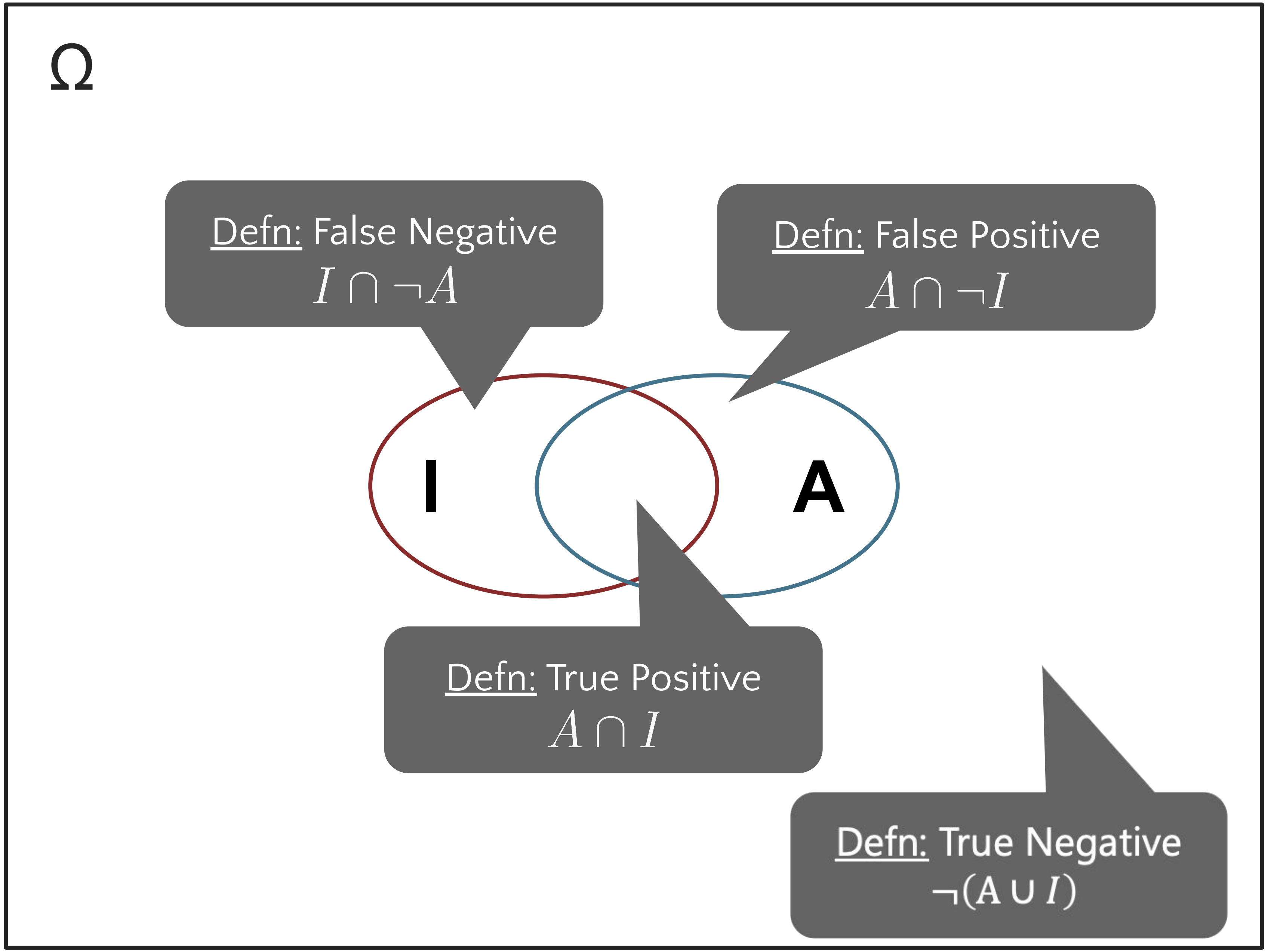
Legend:
Ω = Events
I = Intrusion
A = Alarm



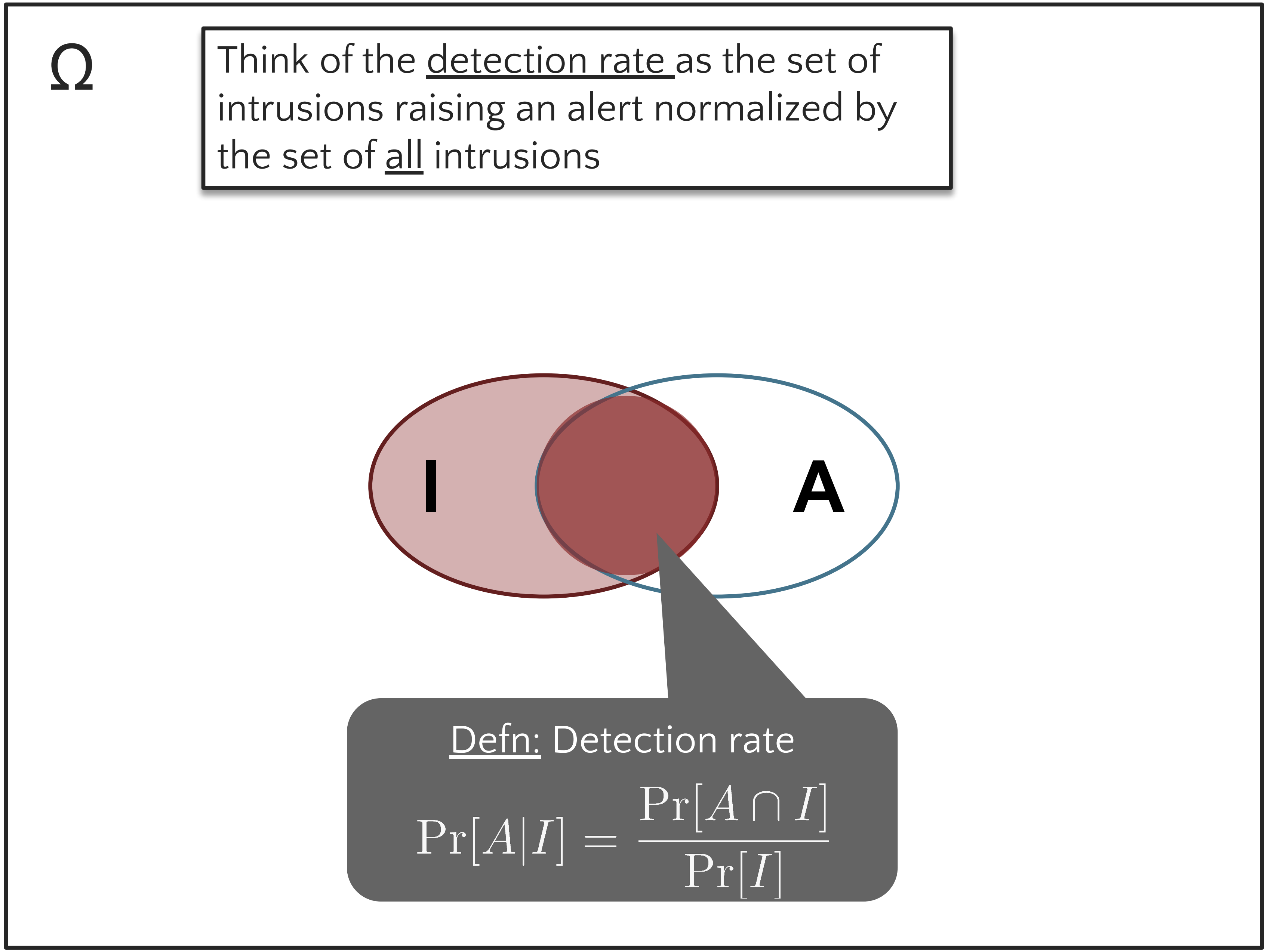
Legend:
Ω = Events
I = Intrusion
A = Alarm



Legend:
Ω = Events
I = Intrusion
A = Alarm



Legend:
Ω = Events
I = Intrusion
A = Alarm



$$\Pr[A|I] = \frac{\Pr[A \cap I]}{\Pr[I]}$$

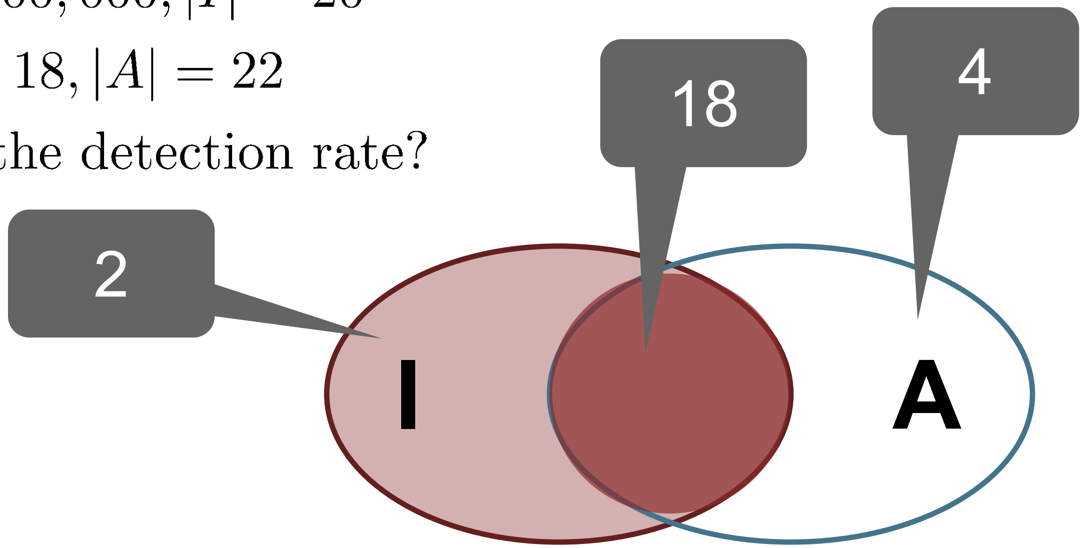
Ω

Suppose:

$|\Omega| = 1,000,000, |I| = 20$

$|I \cap A| = 18, |A| = 22$

What is the detection rate?



Quiz Question

A. 10%
B. 20%
C. 50%
D. 90%

$$\Pr[A|I] = \frac{\Pr[A \cap I]}{\Pr[I]}$$

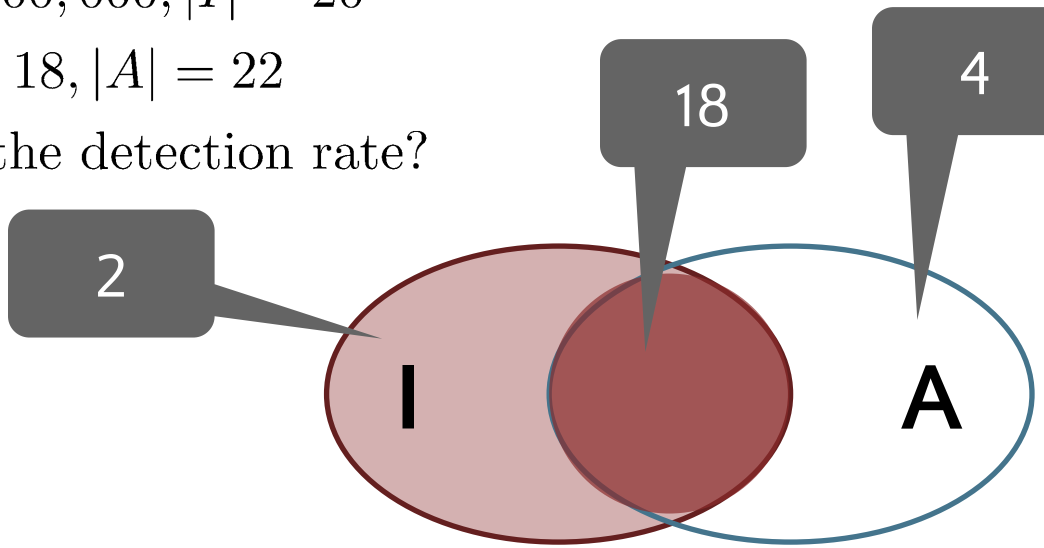
Ω

Suppose:

$|\Omega| = 1,000,000, |I| = 20$

$|I \cap A| = 18, |A| = 22$

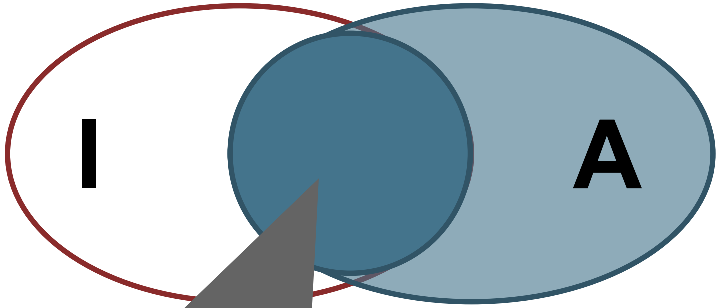
What is the detection rate?



$$Pr[A|I] = \frac{Pr[A \cap I]}{Pr[I]} = 18/20 = .90 = 90$$

Ω

Think of the Bayesian detection rate as the set of *intrusions raising an alert* normalized by the *set of all alerts* (vs detection rate, which normalizes on intrusions)



Defn: *Bayesian* detection rate

$$\Pr[I|A] = \frac{\Pr[A \cap I]}{\Pr[A]}$$

Crux of IDS usefulness!

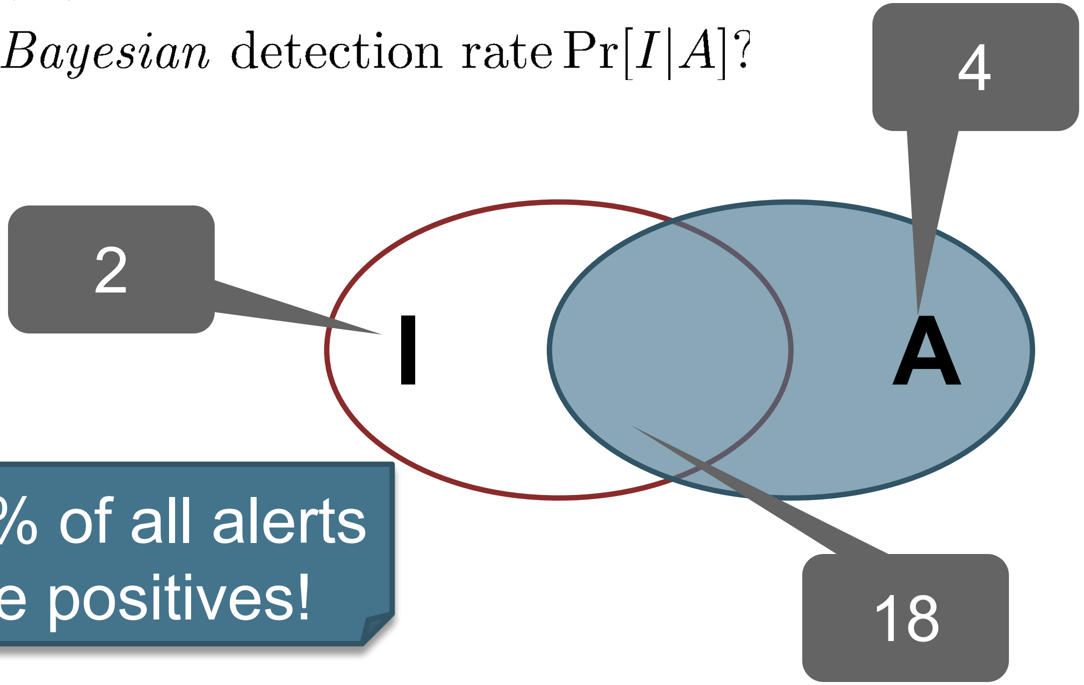
Ω

Suppose:

$|\Omega| = 1,000,000, |I| = 20$

$|I \cap A| = 18, |A| = 22$

What is the *Bayesian* detection rate $\Pr[I|A]$?



About 18% of all alerts are false positives!

$$Pr[I|A] = \frac{Pr[A \cap I]}{Pr[A]} = 18/22 = .81 \approx 82\%$$

Challenge

We're often given the detection rate and can estimate the intrusion rate, and want to calculate the Bayesian detection rate

- 99% accurate medical test
- 99% accurate IDS
- 99% accurate test for deception
- …

Calculating Bayesian Detection Rate

Fact:

$$\Pr[A] = \Pr[I] * \Pr[A|I] + \Pr[\neg I] * \Pr[A|\neg I$$

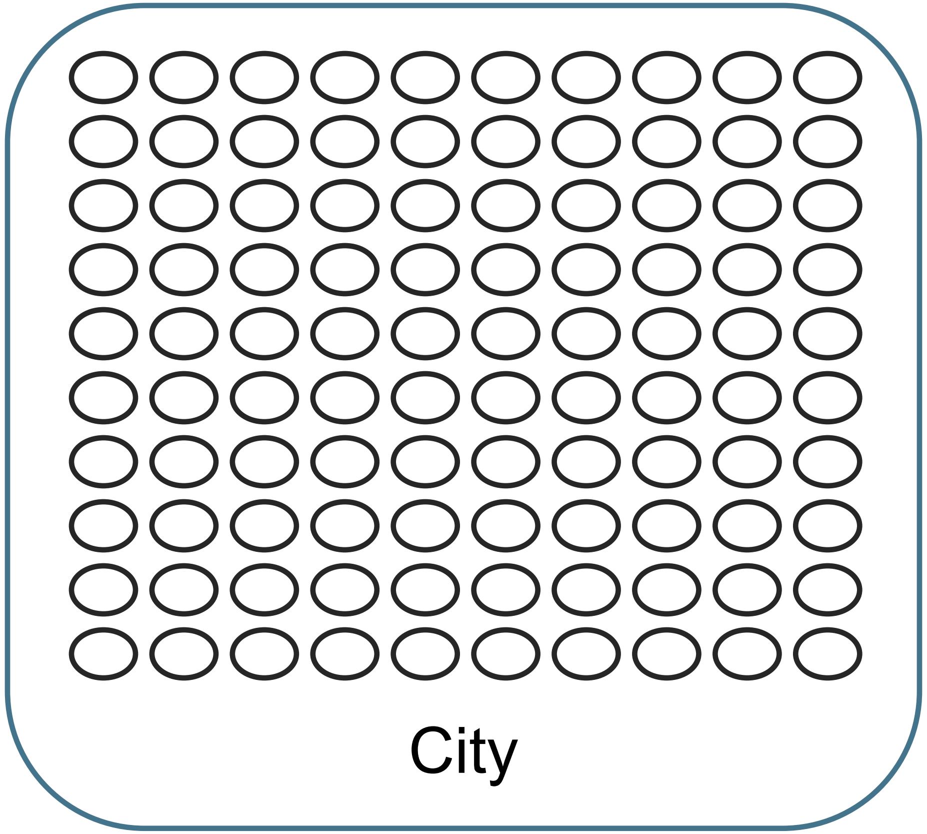
So to calculate the Bayesian detection rate:

$$\Pr[I|A] = \frac{\Pr[A \cap I]}{\Pr[A]}$$

One way is to compute this when Pr[A] but the base rate Pr[I] is:

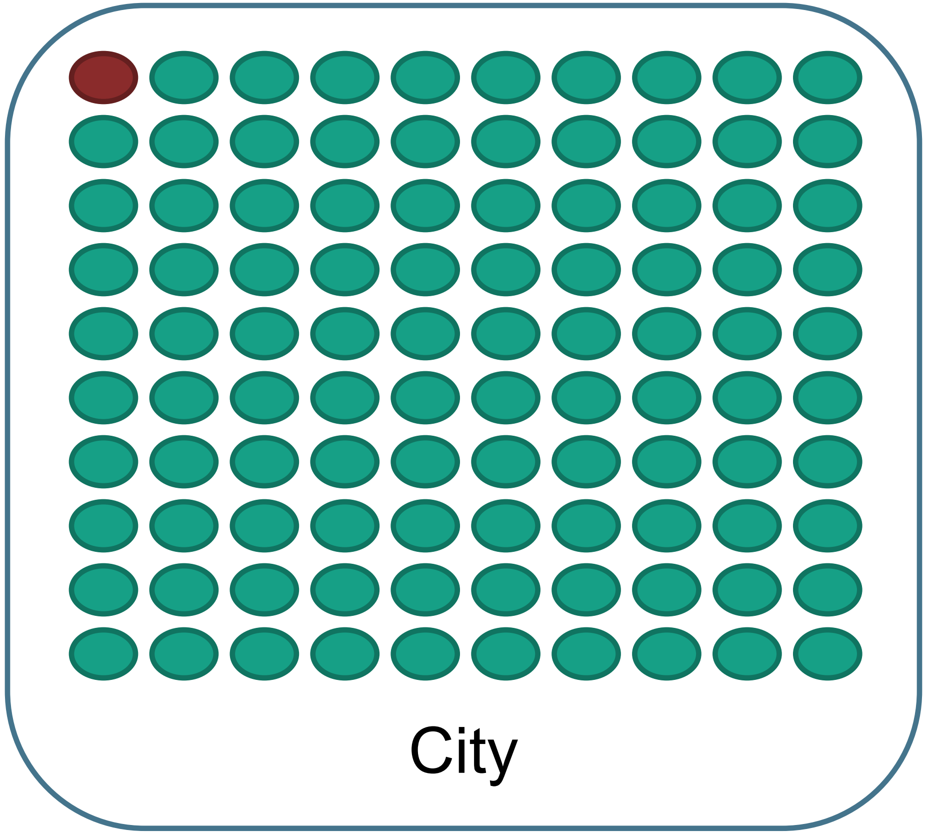
$$\Pr[I|A] = \frac{\Pr[A \cap I]}{\Pr[I] * \Pr[A|I] + \Pr[\neg I] * \Pr[A|\neg I]}$$

Example



- 100 people in the city
- 1 is a terrorist
  - Thus, the *base rate* of terrorists is 1/100
- Suppose we have a new terrorist facial recognition system that is 99% accurate
  - 99/100 times when someone is a terrorist there is an alarm
  - For every 100 good guys, the alarm only goes off once
- An alarm went off; is the suspect really a terrorist?

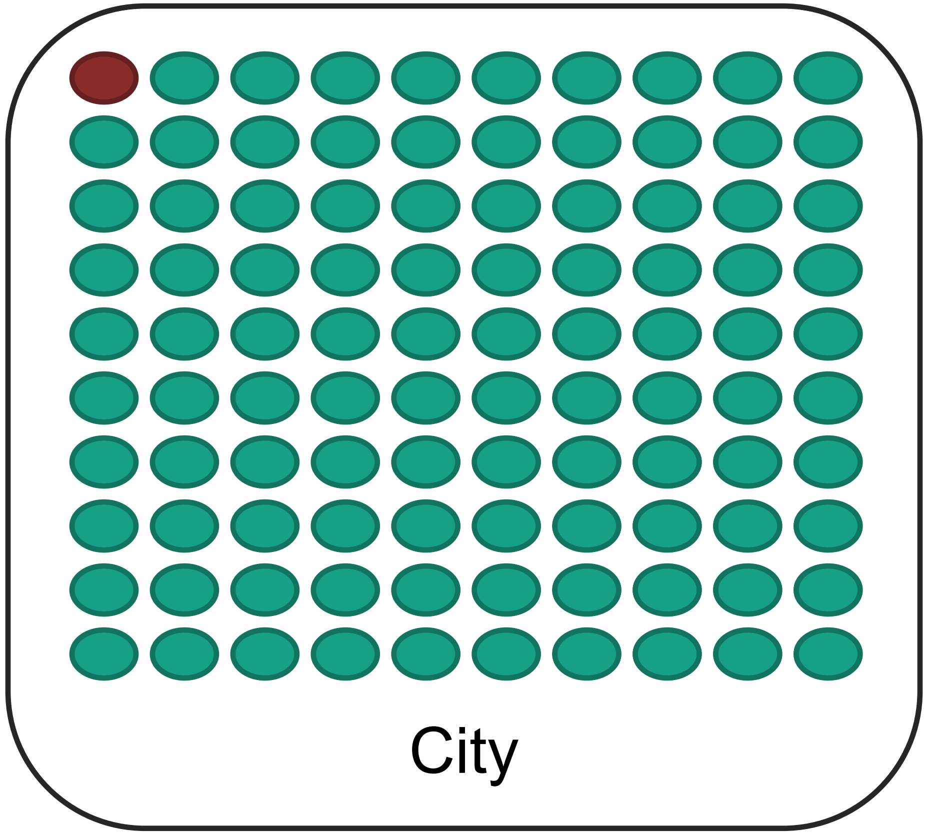
Example



Answer: The facial recognition system is 99% accurate. That means there is only a 1% chance the guy is not the terrorist.

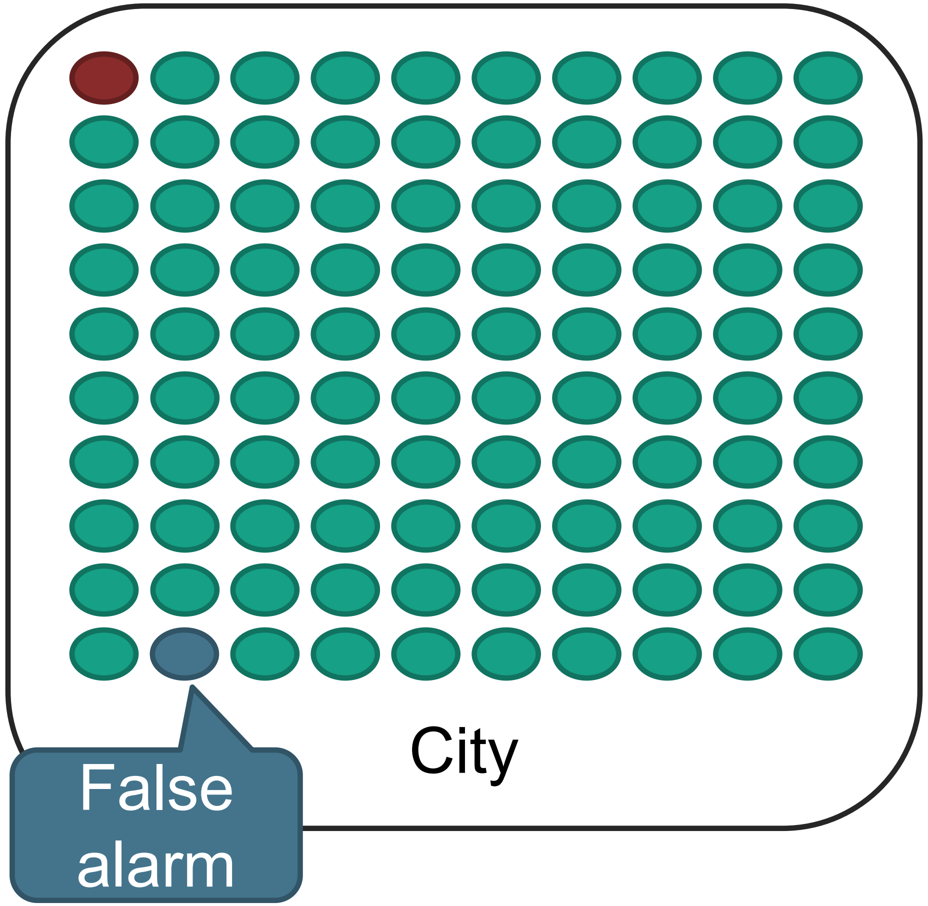
wrong!

Formalization



- 1 is a terrorist, and we have their picture
  - Thus, the *base rate* of terrorists is 1/100
    P[T] = 0.01
- 99/100 times when someone is a terrorist there is an alarm
  P[A|T] = .99
- For every 100 good guys, the alarm only goes off once
  P[A | not T] = .01
- Want to know **P[T|A]**

__Intuition__: Given 99 good guys, we have 99*.01 ≈ 1 false alarm



- 1 is a terrorist, and we have their picture
  - Thus, the ***base rate*** of terrorists is 1/100
    P[T] = 0.01
- 99/100 times when someone is a terrorist there is an alarm
  P[A|T] = .99
- For every 100 good guys, the alarm only goes off once
  P[A | not T] = .01
- Want to know **P[T|A]**

$$\text{Have: } \Pr[T] = 0.01$$

$$\Pr[A|T] = .99, \Pr[A|\neg T] = .01$$

$$\text{Want to calculate: } \Pr[T|A] = \frac{\Pr[T \cap A]}{\Pr[A]}$$

Unknown

Unknown

Mathematically..

$$\Pr[A \cap I] = \Pr[A|I] * \Pr[I]$$
$$\Pr[A] = \Pr[I] * \Pr[A|I] + \Pr[\neg I] * \Pr[A|\neg I]$$

$$\text{Have: } \Pr[T] = 0.01$$
$$\Pr[A|T] = .99, \Pr[A|\neg T] = .01$$

$$\text{Want to calculate: } \Pr[T|A] = \frac{\Pr[T \cap A]}{\Pr[A]}$$

$$= \frac{\Pr[T \cap A]}{\Pr[T]*\Pr[A|T]+\Pr[\neg T]+\Pr[A|\neg T]}$$

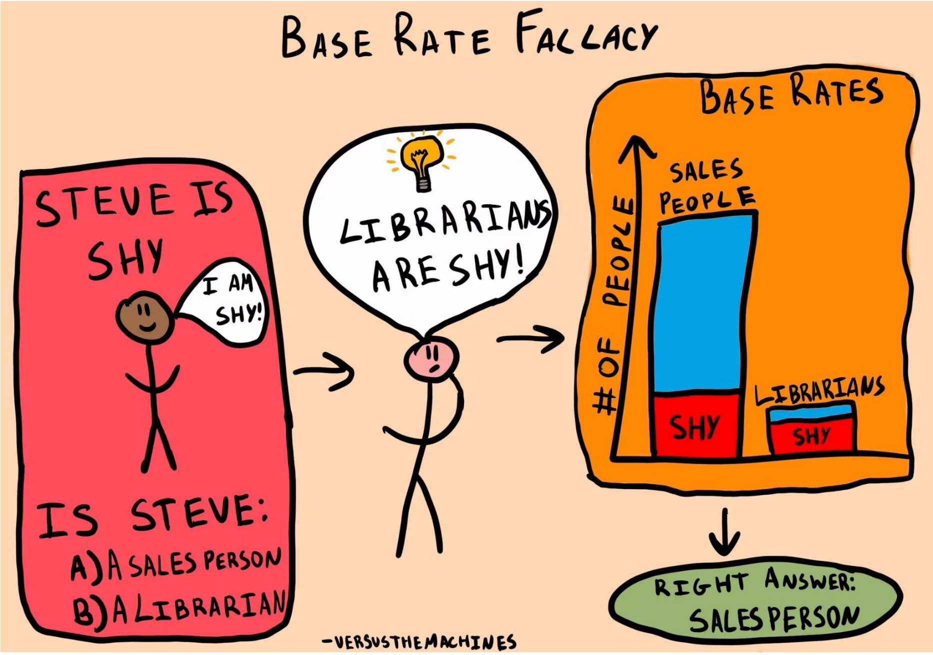
$$= \frac{\Pr[A|T]*\Pr[T]}{\Pr[T]*\Pr[A|T]+\Pr[\neg T]+\Pr[A|\neg T]} = \frac{.99 * .01}{.01 * .99 + .99 \quad * .01}$$

= 50%

99% accuracy + this specific dataset = wrong predictions 50% of the time!

Base Rate Fallacy

- Base rate fallacy = focusing purely on "accuracy" (or similar) and ignoring the base rate
  - Even very high accuracy + very low base rate = potentially very high false positive rate
- Implications for anomaly detection:
  - *Rare* anomalies *very* hard to detect without high false positives



Let's Test Ourselves

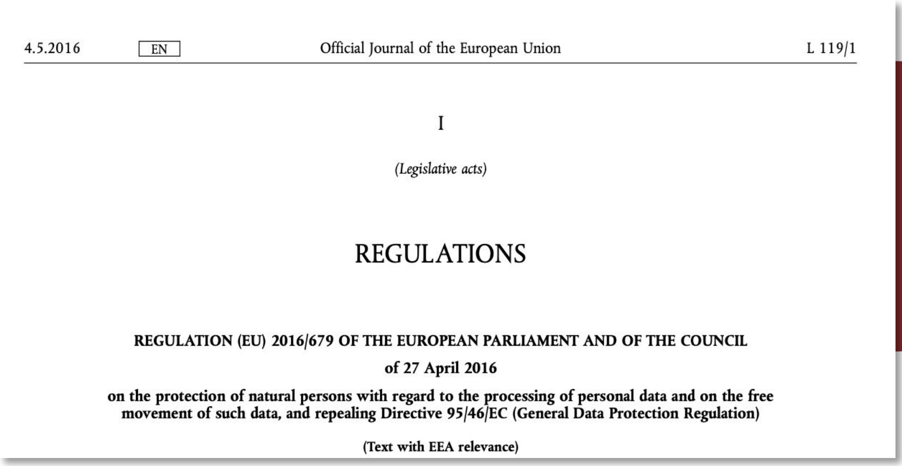
https://www.omnicalculator.com/statistics/false-positive-paradox

Network Security is a Large Field

- **Availability**: Can Alice reach Bob?
- **Reliability**: Do all Alice’s messages reach Bob?
- **Mediation**: Can Alice limit access for Bob?
- **Detection:** Can Alice determine when Bob does something bad?
- **Response:** Can Alice determine what Bob has done?
- **Privacy**: What can Eve learn observing Alice’s (even encrypted) packets?

Privacy

Privacy



4.5.2016 EN Official Journal of the European Union L 119/1

I

(Legislative acts)

REGULATIONS

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

(Text with EEA relevance)

GDPR considers your IP PII, thus is regulated w/ providers (your work has more latitude)

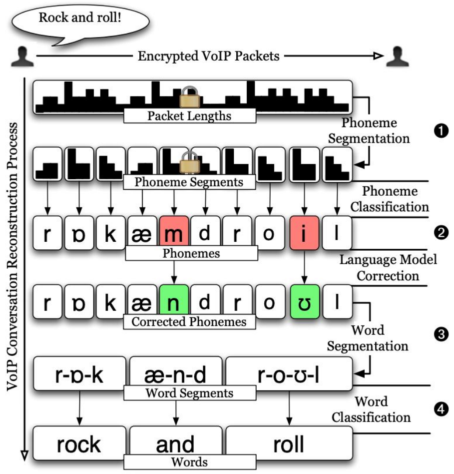


Figure 2. Overall architecture of our approach for reconstructing transcripts of VoIP conversations from sequences of encrypted packet sizes.

Even encrypted traffic can leak information, such as length.

Phonotactic Reconstruction of Encrypted VoIP Conversations, White et al.

"Privacy is a value so complex, so entangled in competing and contradictory dimensions, so engorged with various and distinct meanings, that I sometimes despair whether it can be usefully addressed at all."

Robert C. Post, Three Concepts of Privacy,
89 Geo. L.J. 2087 (2001).

Some Conceptions of Privacy

- Personhood / personality
- Intimacy
- Secrecy
- Right to be let alone
- Limited access to the self
- Control over information

Contextual Privacy

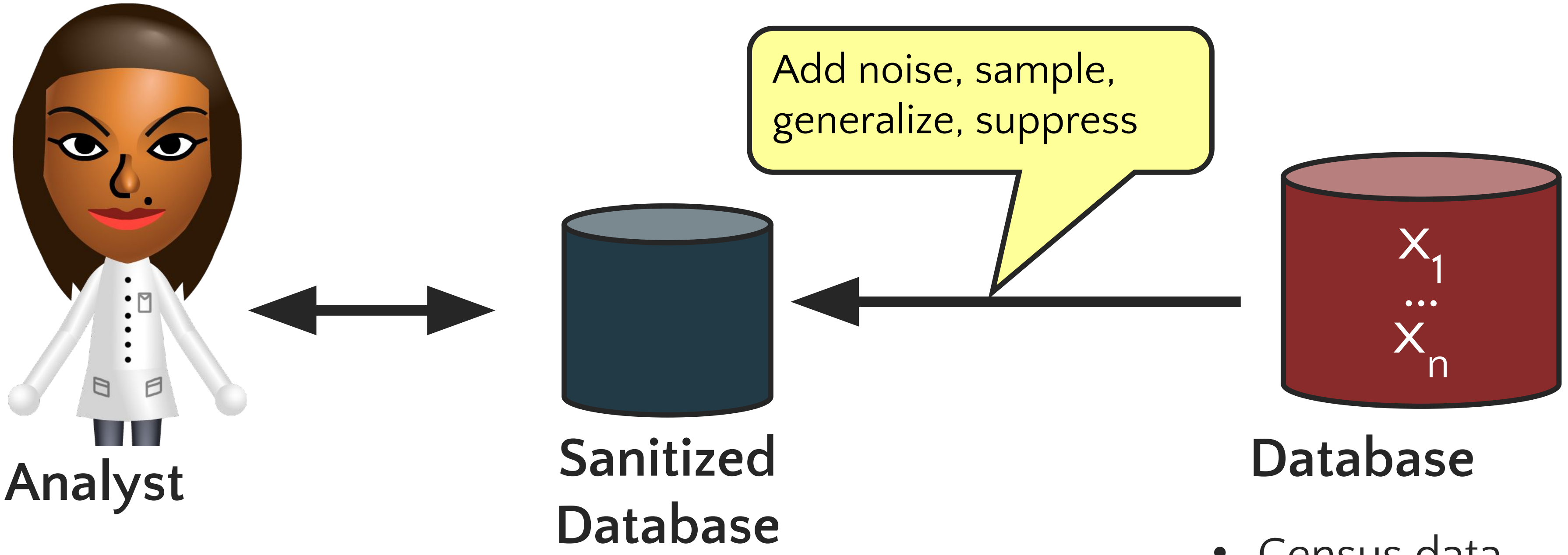
- Privacy is not one-size-fits-all, it varies by context (Nissenbaum)
  - data subject, sender, recipient, information type, and transmission principle
- Do you mind telling everybody in class:
  - The last TV show you watched?
  - What you most recently discussed with your doctor?
- This is reflected in professional guidelines and norms (e.g., Hippocratic Oath)

Information vs Decisional Privacy

- *Information(al) privacy* concerns the collection, use, and disclosure of personal information
- *Decisional privacy* concerns the freedom to make decisions about one's body and family (e.g., Roe v. Wade)

Early Attempts in Data Privacy

Goal: Privacy-Preserving Data Disclosure



- Census data
- Health data
- Network data
- …

Anonymizing Data Is Hard!

AOL Proudly Releases Massive Amounts of Private Data

Michael Arrington @arrington?lang=en / 12 years ago Comment

SECTIONS The New York Times

TECHNOLOGY

A Face Is Exposed for AOL Searcher No. 4417749

By MICHAEL BARBARO and TOM ZELLER Jr. AUG. 9, 2006

<u>User # 4417749</u>
numb fingers
60 single men
dog that urinates on everything
landscapers in Lilburn, Ga
homes sold in shadow lake subdivision gwinnett county georgia
[various first names] Arnold

Thelma
Arnold

Anonymizing Data Is Hard!

AOL Proudly Releases Massive Amounts of Private Data

Michael Arrington @arrington?lang=en / 12 years ago Comment

<u>User # 2178</u>

foods to avoid when breast feeding

<u>User # 3505202</u>

depression and medical leave

<u>User # 7268042</u>

fear that spouse contemplating cheating

Anonymizing Data is Hard!

> **AOL Proudly Releases Massive Amounts of Private Data**
>
> **Michael Arrington** @arrington?lang=en / 12 years ago Comment

<u>Consequences</u>

- Researcher and his supervisor fired
- CTO resigned
- Class-action lawsuit against AOL
- Companies less willing to share data

Anonymizing Data Is Hard!

100,480,507 movie ratings
Created by 480,189 Netflix subscribers
From 12/99–12/05

“all customer identifying information has been removed; … only a small sample was included (less than one tenth of our complete dataset) and that data was subject to perturbation”

Does ~~the average Netflix~~ any subscriber care about the privacy of his/her movie viewing history?

Anonymization Mechanism

Netflix DB

Row = Individual

Column = Attribute (e.g., movie)

	Gladiator	Titanic	Heidi
Bob	5	2	1
Alice	3	2.5	2
Charlie	1.5	2	

Delete name ID

Add Noise

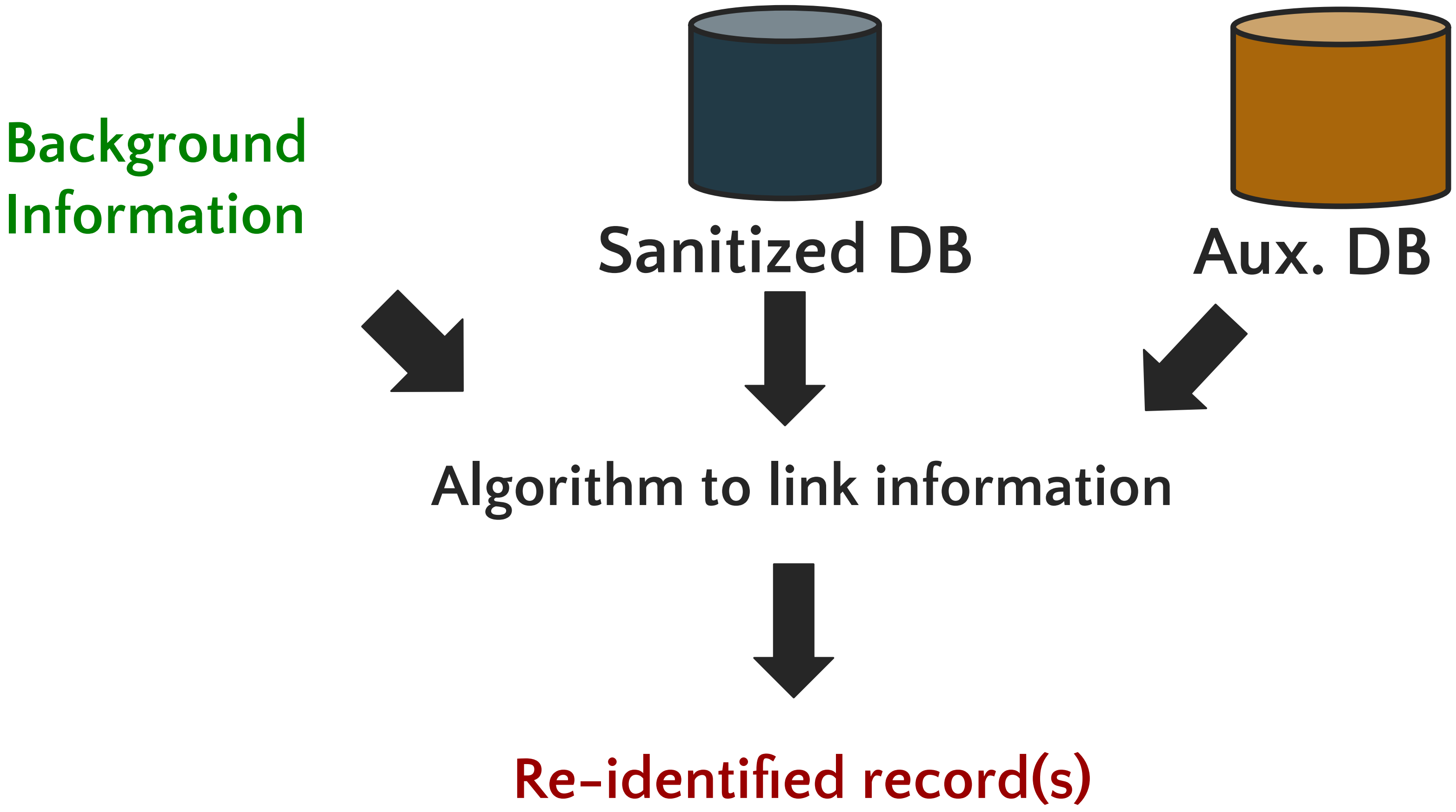
“Sanitized” Netflix DB

	Gladiator	Titanic	Heidi
? r_1	4	1	0
r_2	2	1.5	1
r_3	0.5	1	

De-anonymization Attacks Still Possible

- Isolation Attacks
  - Recover individual's record from sanitized DB
    - E.g., find user's ratings in sanitized Netflix movie DB
- Information Amplification Attacks
  - Find more information about individual in sanitized DB
    - E.g., find more ratings for specific movie for user in Netflix DB

Non-Interactive Linking



Netflix-IMDb Empirical Attack

[Narayanan et al. 2008]

Anonymized Netflix DB

	Gladiator	Titanic	Heidi
r_1	4	1	0
r_2	2	1.5	1
r_3	0.5	1	1

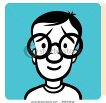
Publicly available IMDb ratings (noisy)

	Titanic	Heidi
Bob	2	0

Used as auxiliary information

Weighted Scoring Algorithm

Isolation Attack!



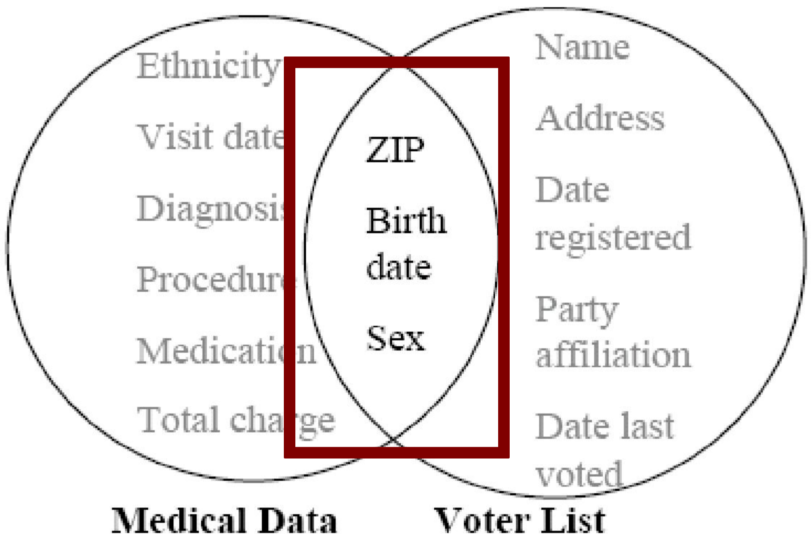
r_1	4	1	0

De-Anonymizing Netflix: Results

- 99% of records can be uniquely IDed with 8 ratings (2 may be wrong) and dates +/- 14 days
- 68% with 2 ratings and dates +/- 3 days

Re-identification by Linking

Linking two sets of data on shared attributes may uniquely identify some individuals



87 % of US population uniquely identifiable by 5-digit ZIP, gender, DOB

Publicly-Released “Anonymized” Datasets Broken

- Useful for research purposes
  - Improving recommendation systems
  - Social sciences
- Contain personal information
- Removing identifiers insufficient
- Adding noise may still be insufficient

Differential Privacy

Classical Intuition for Privacy

"If the release of statistics S makes it possible to determine the value [of private information] more accurately than is possible without access to S, a disclosure has taken place."
[Dalenius 1977]

- Privacy means that anything that can be learned about a respondent from the statistical database can be learned without access to the database

(Similar to semantic security of encryption)

Impossibility Result [Dwork, Naor 2006]

Result: In any "reasonable" setting, even a sanitized database combined with auxiliary information can lead to a privacy breach

Example

- Imagine a database that contains the heights of Lithuanian men and allows users to calculate the average height.
- If the average height of a Lithuanian man in this database is publicly known, and you also know that Andrew Carnegie is eight inches shorter than this average, you can deduce Andrew Carnegie's height without his personal data being in the database.
- This deduction can be made purely based on the average information from the database and the additional fact you know about Carnegie.

Differential privacy

[Dwork, McSherry, Nissim, Smith 2006]

Differential privacy formalizes privacy in statistical and machine learning analysis.

- In simple terms, it says the probability of a particular outcome of an algorithm (e.g., query), does not change much whether or not any individuals data is included.
- ”Not change much” is formalized by a ϵ value, where smaller ϵ values correspond to greater privacy protection

Differential Privacy

Database 1: Has Andrew's info

Name	F1	F2
Andrew		
Bob		
Charlie		
David		

→ Data analysis algorithm w/ randomization → Result 1

Database 2: Andrew's info missing

Name	F1	F2
Bob		
Charlie		
David		

→ Data analysis algorithm w/ randomization → Result 2

Intuition

If Result 1 = Result 2, then the presence of Andrew's info didn't reveal any new information

Privacy Laws and Regulation

US vs EU Approach

US

- Mostly sector-specific laws, with relatively minimal protections - often referred to as "patchwork quilt"
- No explicit constitutional right to privacy
- Tension between federal and state laws
- Many self-regulatory programs

EU

- General Data Protection Regulation (GDPR) is EU-wide, comprehensive privacy law
- Privacy as fundamental human right
- Before GDPR: Privacy commissions in each country (some countries have national and state commissions)

EU General Data Protection Regulation (GDPR)

- Law passed by EU Parliament
- Single set of rules governing data privacy across EU
- Fairly strict legalization of FIPs
  - Can contest “algorithmic” decisions
  - Companies may require a Data Protection Officer
  - Notification of data breaches
  - Higher sanctions
  - Pseudonymization
  - Right to erasure (right to be forgotten)
  - Data portability
  - Records of processing activities must be kept
- Administrative penalties up to 4% of global revenue
- Controller vs processor

Security Regulation

In a nutshell

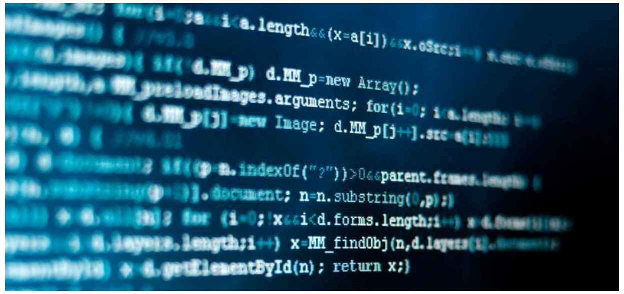
- Computer Fraud and Abuse Act (1986)
  - Prohibits accessing a computer without authorization, or in excess of authorization
  - Criminalized distributing malware, DoS attacks,
  - Many amendments
- Digital Millennium Copyright Act (DMCA, 1998):
  - Anti-circumvention provisions criminalize circumventing a copyright protection device.
- Sector-specific laws and privacy-focused laws
  - May require protecting systems, have penalties for failure to do so

Export Overview

- Two primary set of regulations in the US:
  - EAR: Export Administration Regulations
  - International Traffic in Arms Regulations (ITAR)
- Export includes physical export, but also simply letting a foreign national (even in the US) know about something export controlled

<u>Consider this</u>: if you are a US-based company with 1 employee in Canada, and 99 employees in the US, you must segregate possible export-controlled information from that 1 employee

Crypto Wars: Policy Question



?
==

Dual use technologies are those with both commercial and military applications. Crypto is an example.

Should the export of cryptography be regulated?

A. Yes

B. In some cases

C. No

Crypto and Export

Encryption products, especially those that are strong or use advanced cryptographic techniques, may be considered as defense articles under ITAR if they are specifically designed or modified for military applications

- Military encryption
- Encryption hardware. (It's unclear whether the AES instructions in every x64 chip, for example, are export controlled. Depends who you ask.)
- Satellite encryption systems, VoIP encryption, and some other random examples.

Anonymity

Communication anonymity

- Two approaches covered here
  - Mixes (a.k.a. mixnets)
    - Originally proposed by David Chaum at UC Berkeley around 1980 for untraceable email
  - Proxies
    - Generally used by web browsing anonymizing services

Motivation

Alice wants to send Bob a letter, but doesn't want people to know she sent Bob a letter.

Bob

Alice

Chaum Mix (1981)

Bob

Alice

Minnie (Mix)

Bob

Chaum Mix (1981)

(Envelopes are sealed using the recipient's public key)

Minnie (Mix)

Bob

Alice

Chaum Mix (1981)

(Envelopes are sealed using the recipient's public key)

Minnie (Mix)

Alice

Bob

Chaum Mix (1981)

(Envelopes are sealed using the recipient's public key)

Minnie (Mix)

Bob

Alice

Chaum Mix (1981)

(Envelopes are sealed using the recipient's public key)

Minnie (Mix)

Alice

Bob

Chaum Mix (1981)

(Envelopes are sealed using the recipient's public key)

Minnie (Mix)

Bob

Alice

Chaum Mix

- No one but Minnie knows the author of the original letter
- However, an observer could easily guess it is Alice by observing that Alice sent something to Minnie shortly before Minnie sent it to Bob

Chaum mix w/ multiple participants

Alice

Bob

Carol

Minnie (Mix)

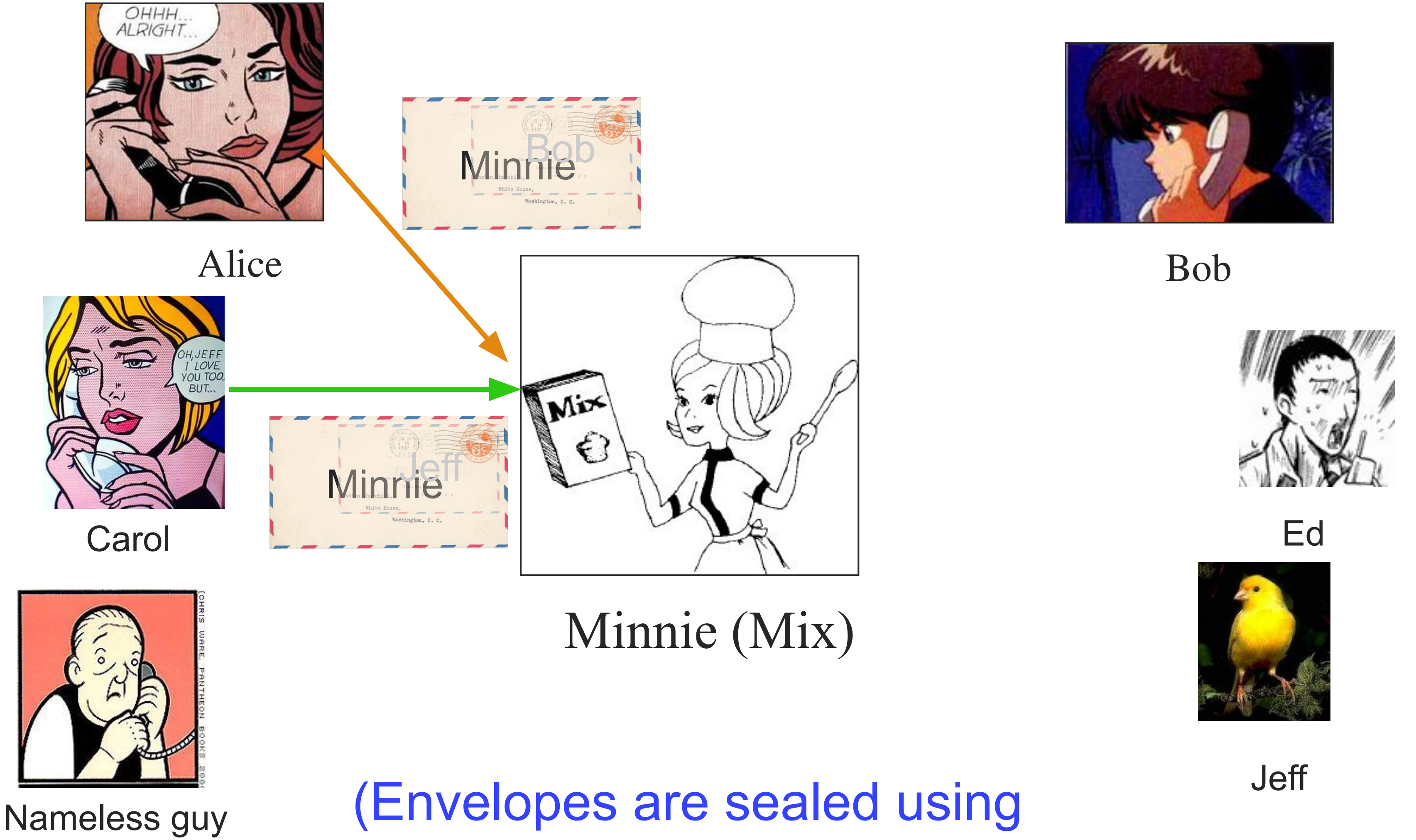
Ed

Nameless guy

Jeff

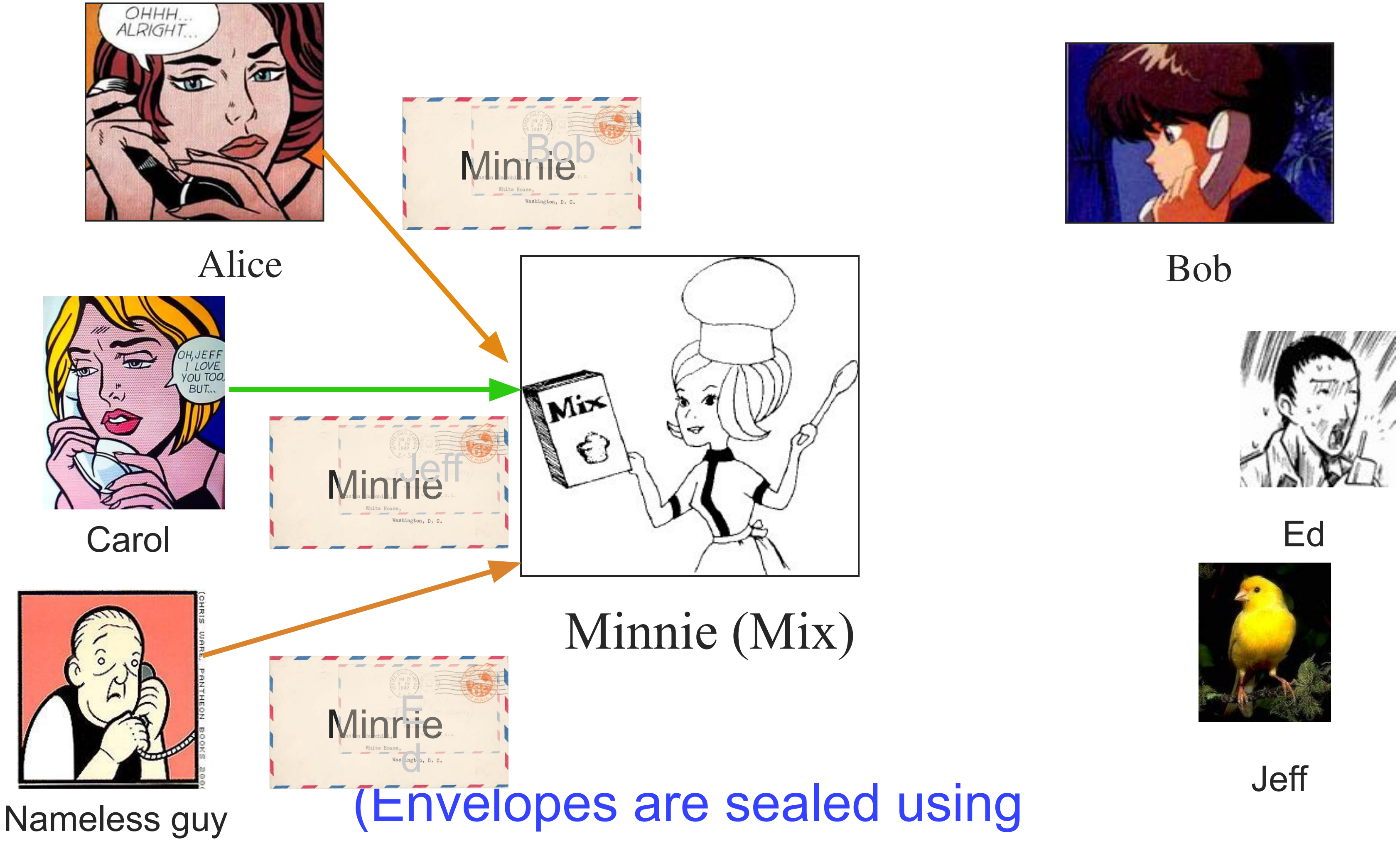
(Envelopes are sealed using the recipient's public key)

Chaum mix w/ multiple participants



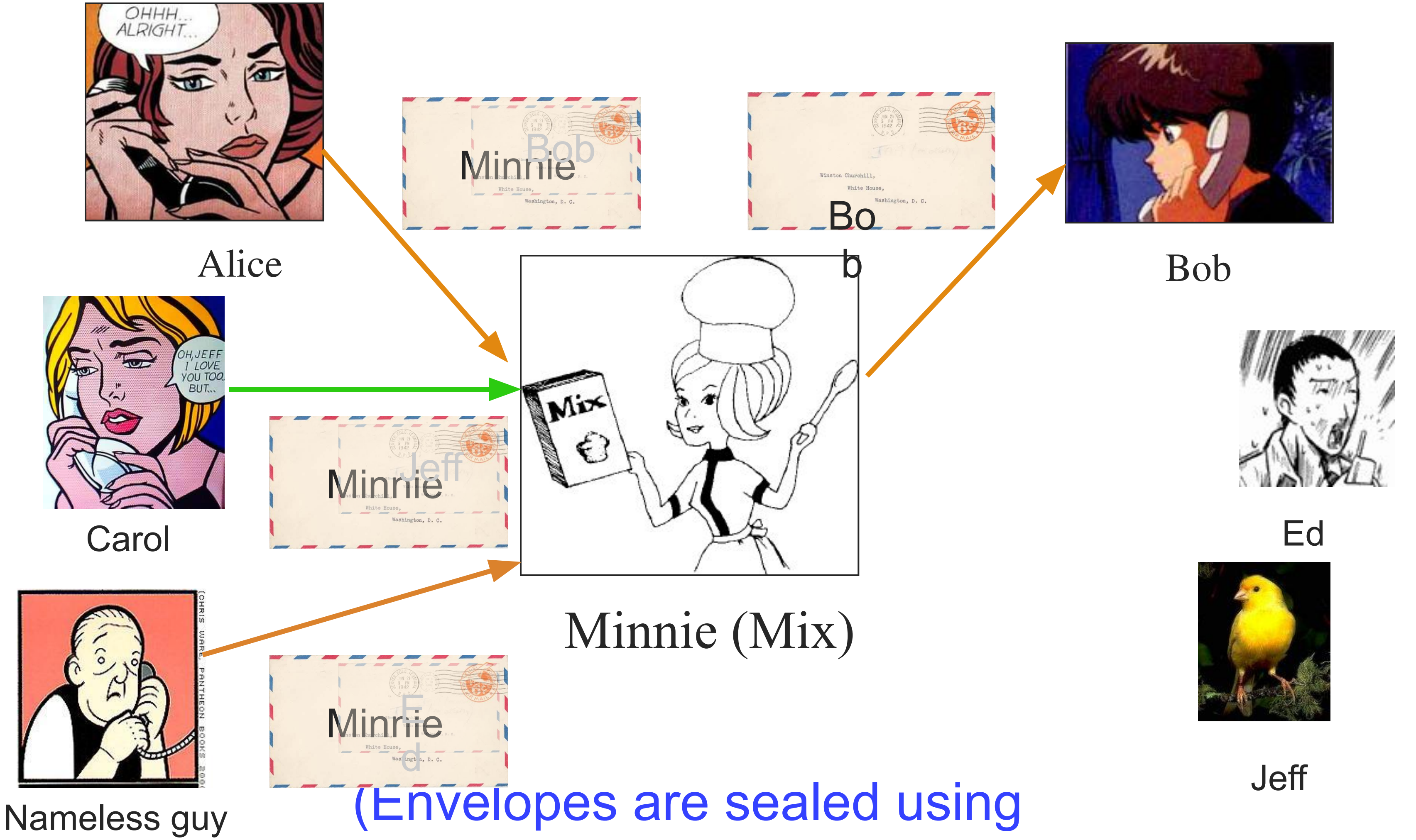
(Envelopes are sealed using the recipient's public key)

Chaum mix w/ multiple participants



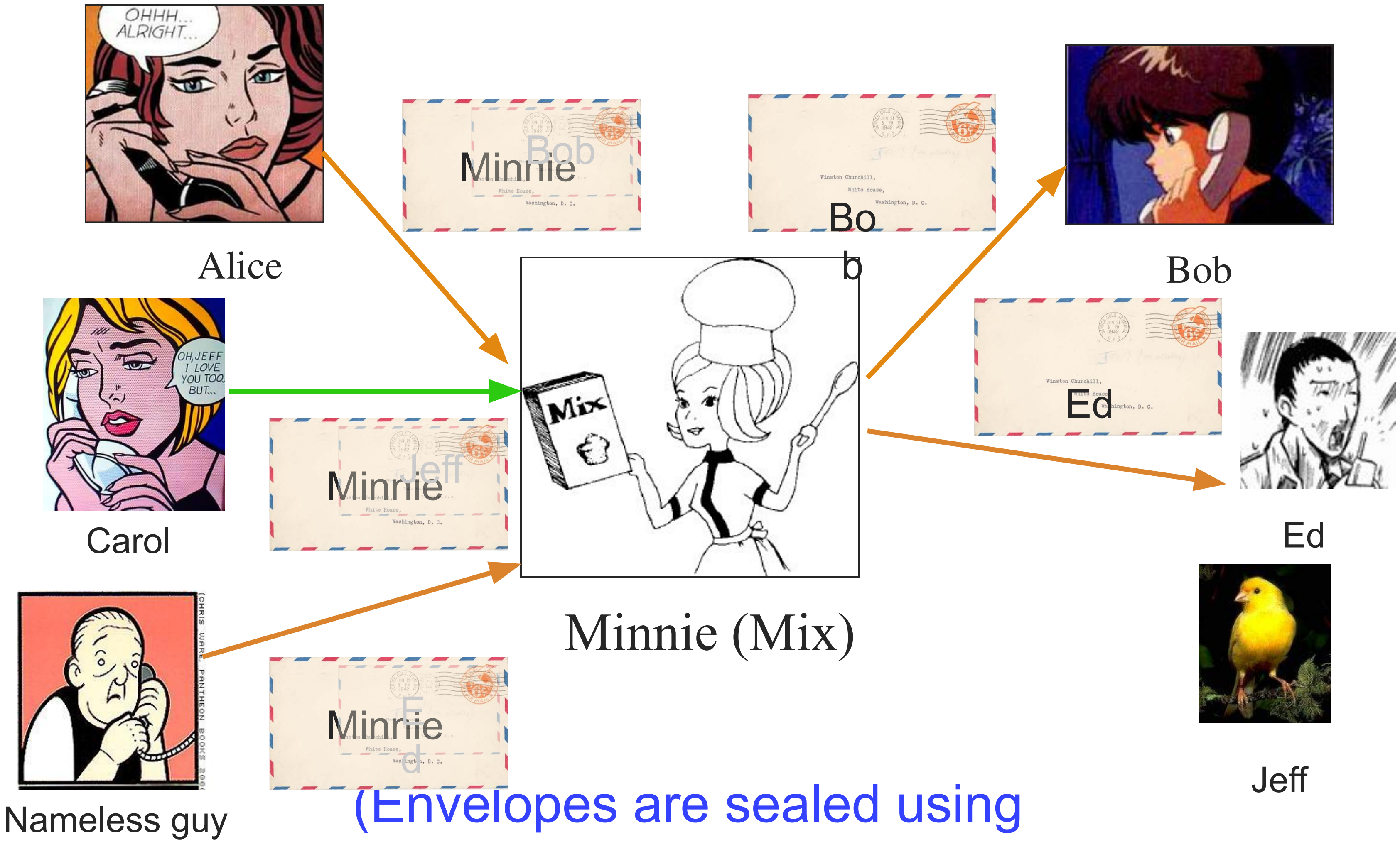
(Envelopes are sealed using the recipient's public key)

Chaum mix w/ multiple participants



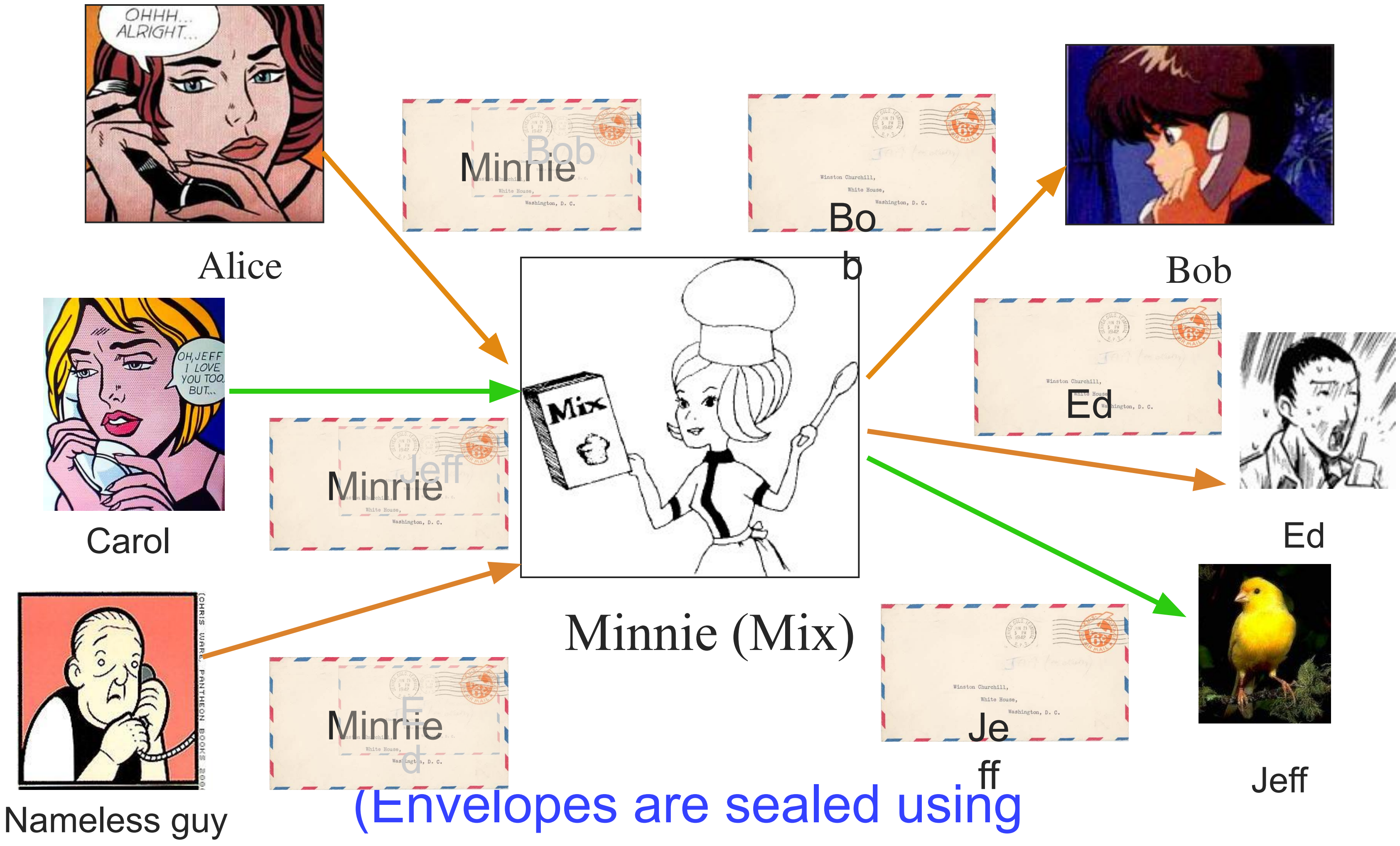
(Envelopes are sealed using the recipient's public key)

Chaum mix w/ multiple participants



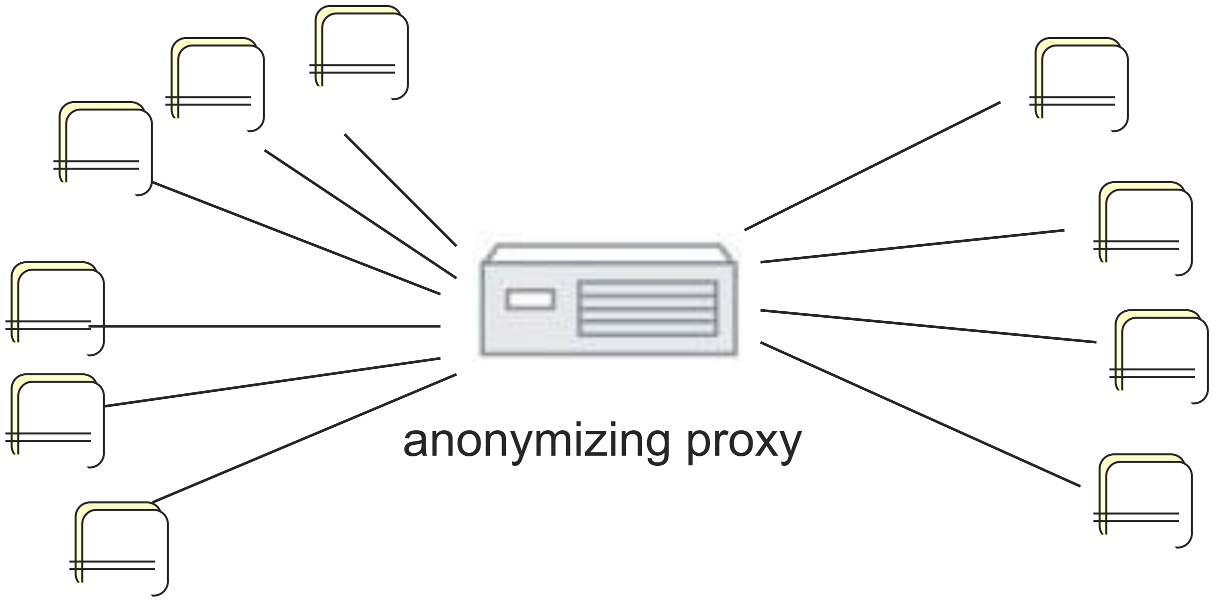
(Envelopes are sealed using the recipient's public key)

Chaum mix w/ multiple participants



(Envelopes are sealed using the recipient's public key)

(Basic) Anonymizing proxy



- Conceptually much simpler solution
- Channels appear to come from proxy, not true originator
  - IPsec can actually implement this!

Chaum mix w/ multiple participants

- Mix (Minnie in the example) reorders messages (e.g., in lexicographic order)
- Only Minnie knows who is talking to whom (but she doesn't know the content of the message)
- Note that Minnie can actually be part of the people talking if she can use another mix herself (e.g., if Alice can perform the functions of a mix)

Tor is a proxy network that uses mixing to try and achieve three goals:

- **Anonymity:** Keep adversaries from learning who is talking to who
- **Privacy:** crypto keeps traffic secret
- **Anti-Censorship:** Allow users to access resources otherwise blocked

How Does Tor Work?

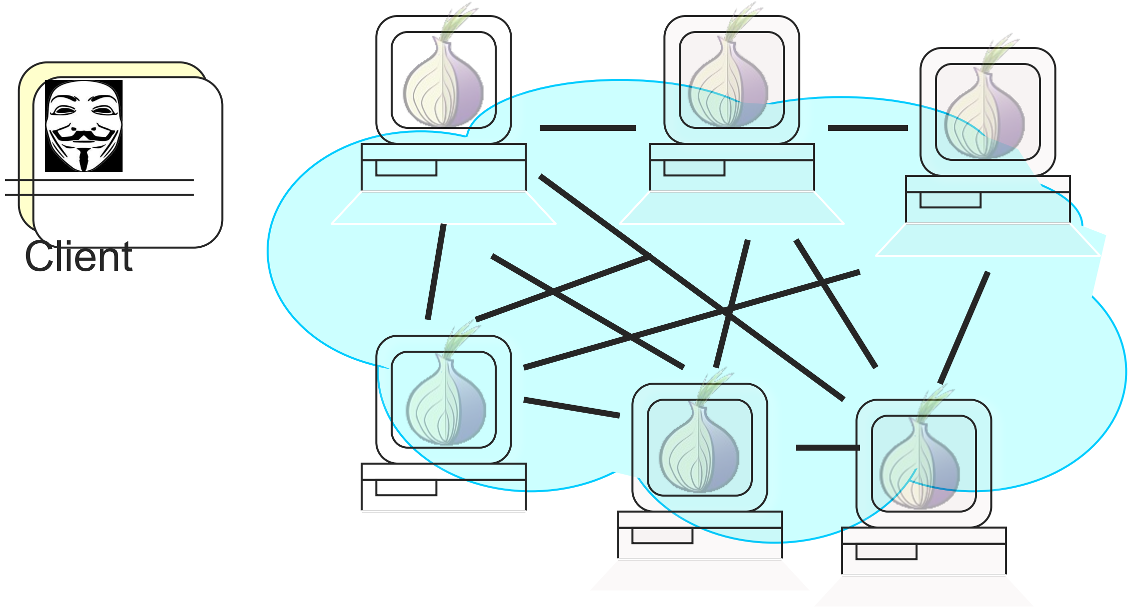
[Dingledine et al., 2004]

Client first gets IP address of possible Tor entry nodes from directory server

How Does Tor Work?

[Dingledine et al., 2004]

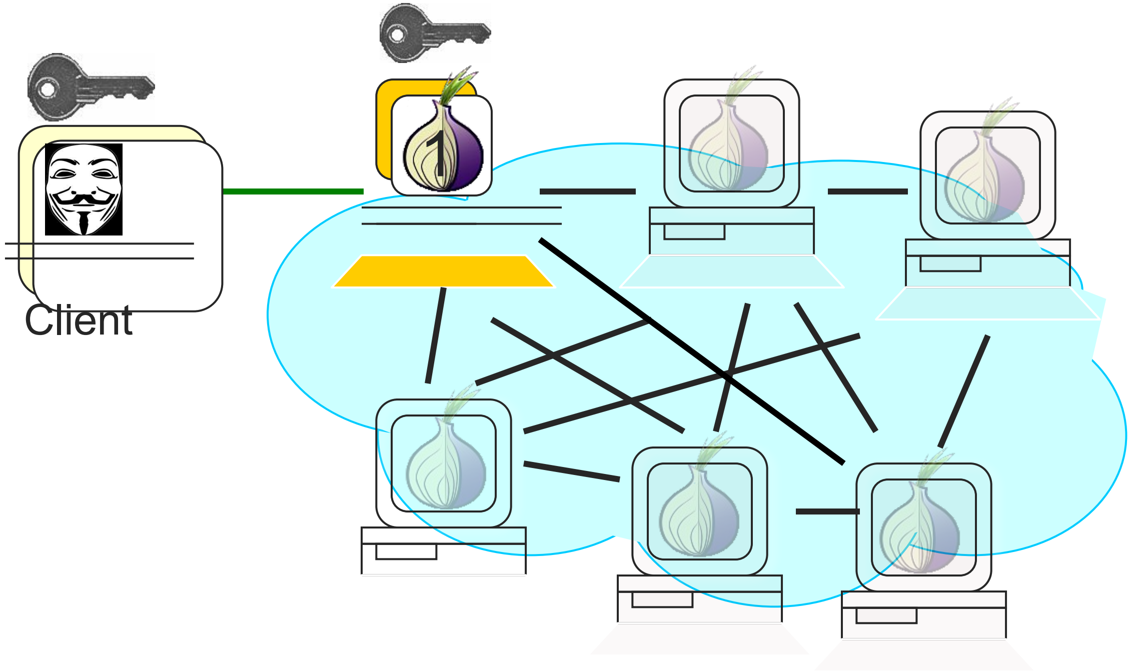
Client first gets IP address of possible Tor entry nodes from directory server



How Does Tor Work?

[Dingledine et al., 2004]

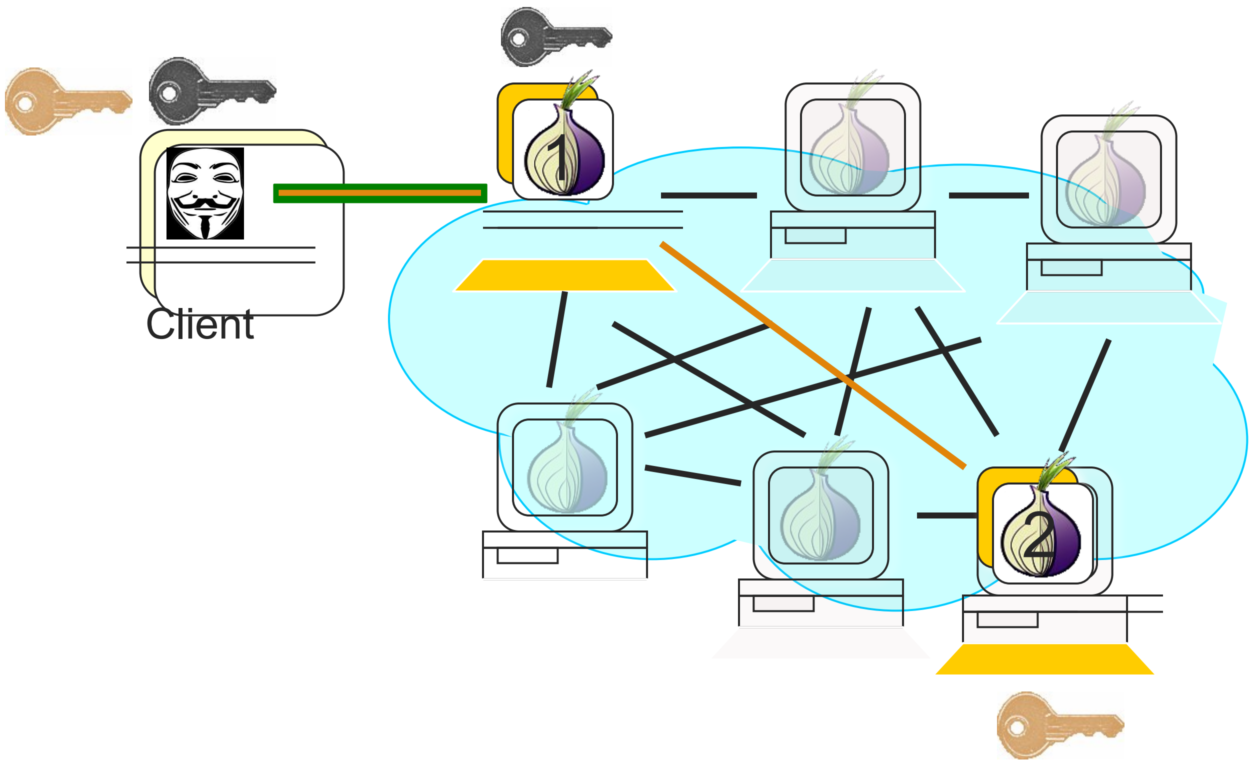
Client proxy establishes session key+circuit w/ Onion Router 1



How Does Tor Work?

[Dingledine et al., 2004]

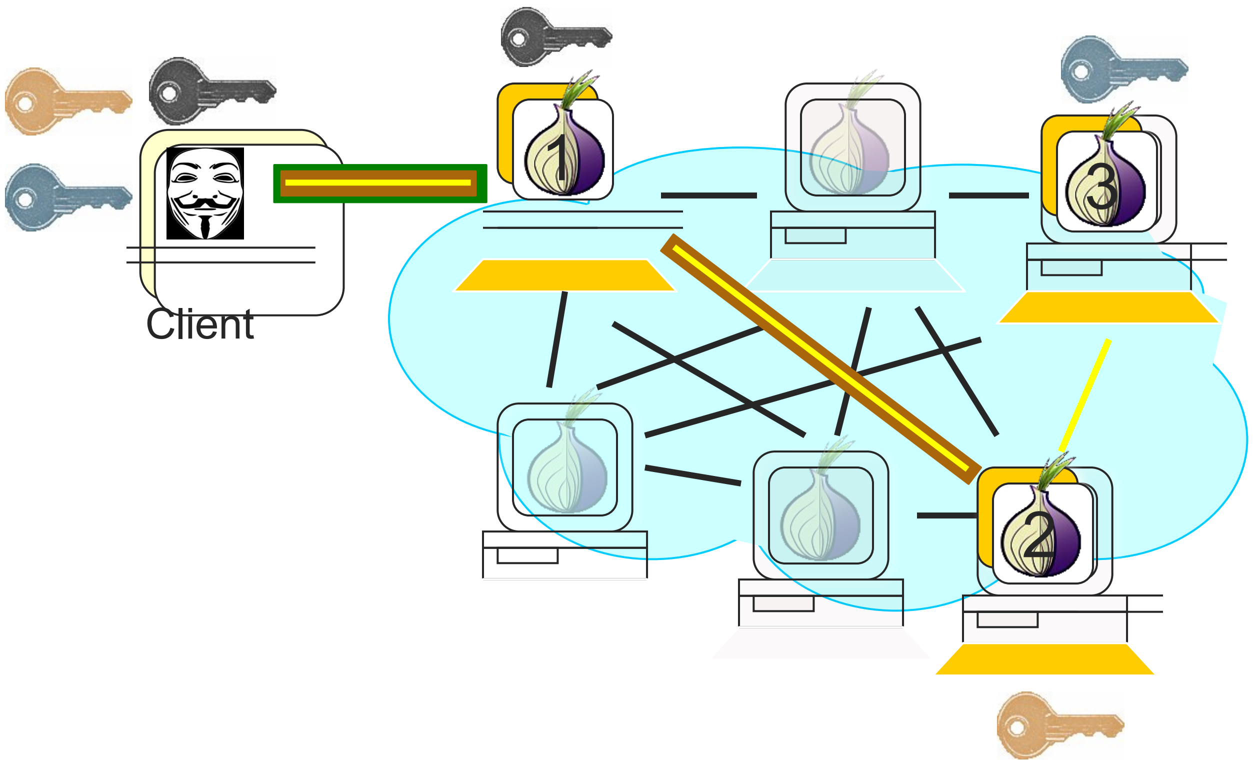
- Client proxy establishes session key+circuit w/ Onion Router 1
- Proxy tunnels through that circuit to extend to Onion Router 2



How Does Tor Work?

[Dingledine et al., 2004]

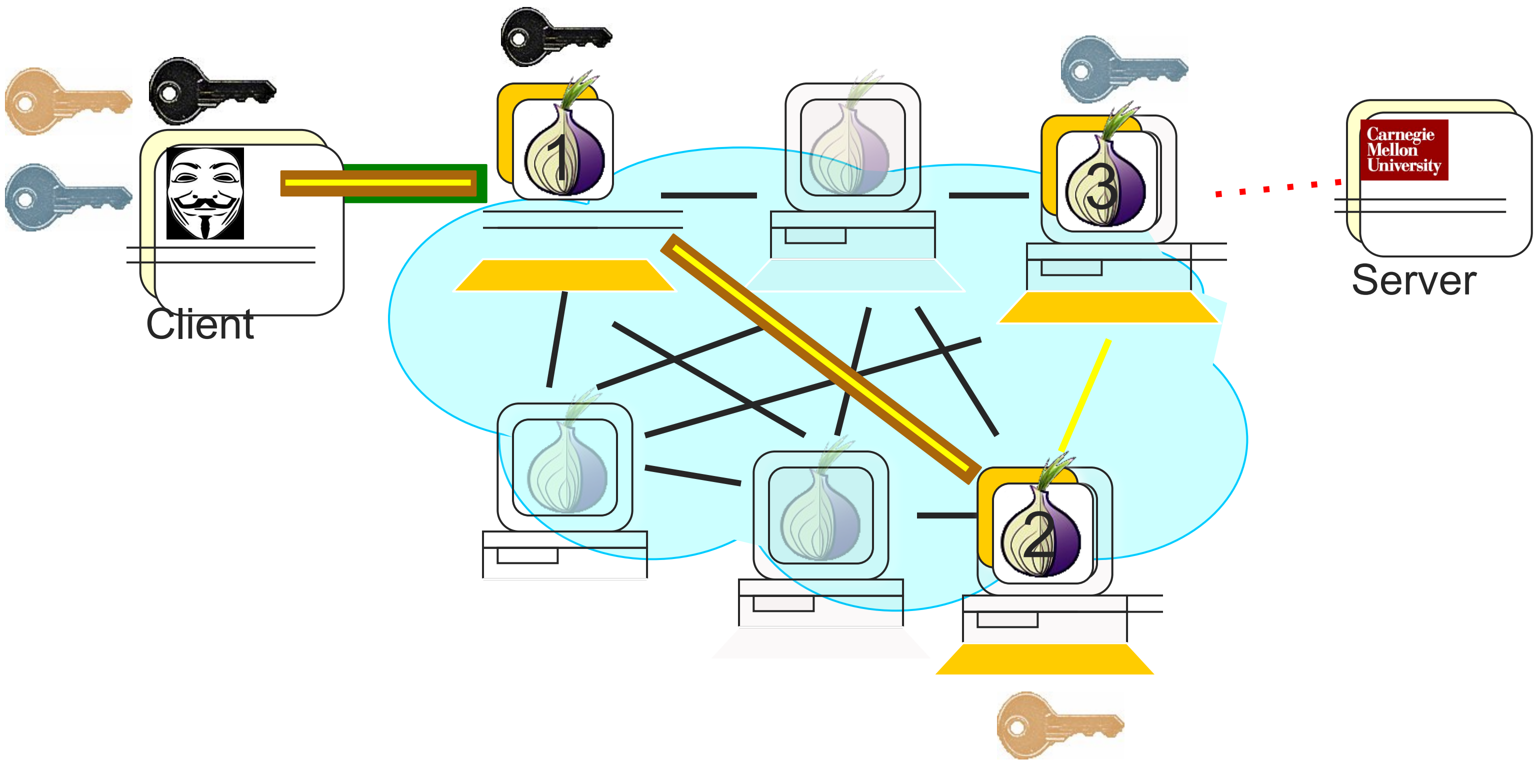
- Client proxy establishes session key+circuit w/ Onion Router 1
- Proxy tunnels through that circuit to extend to Onion Router 2
- Etc.



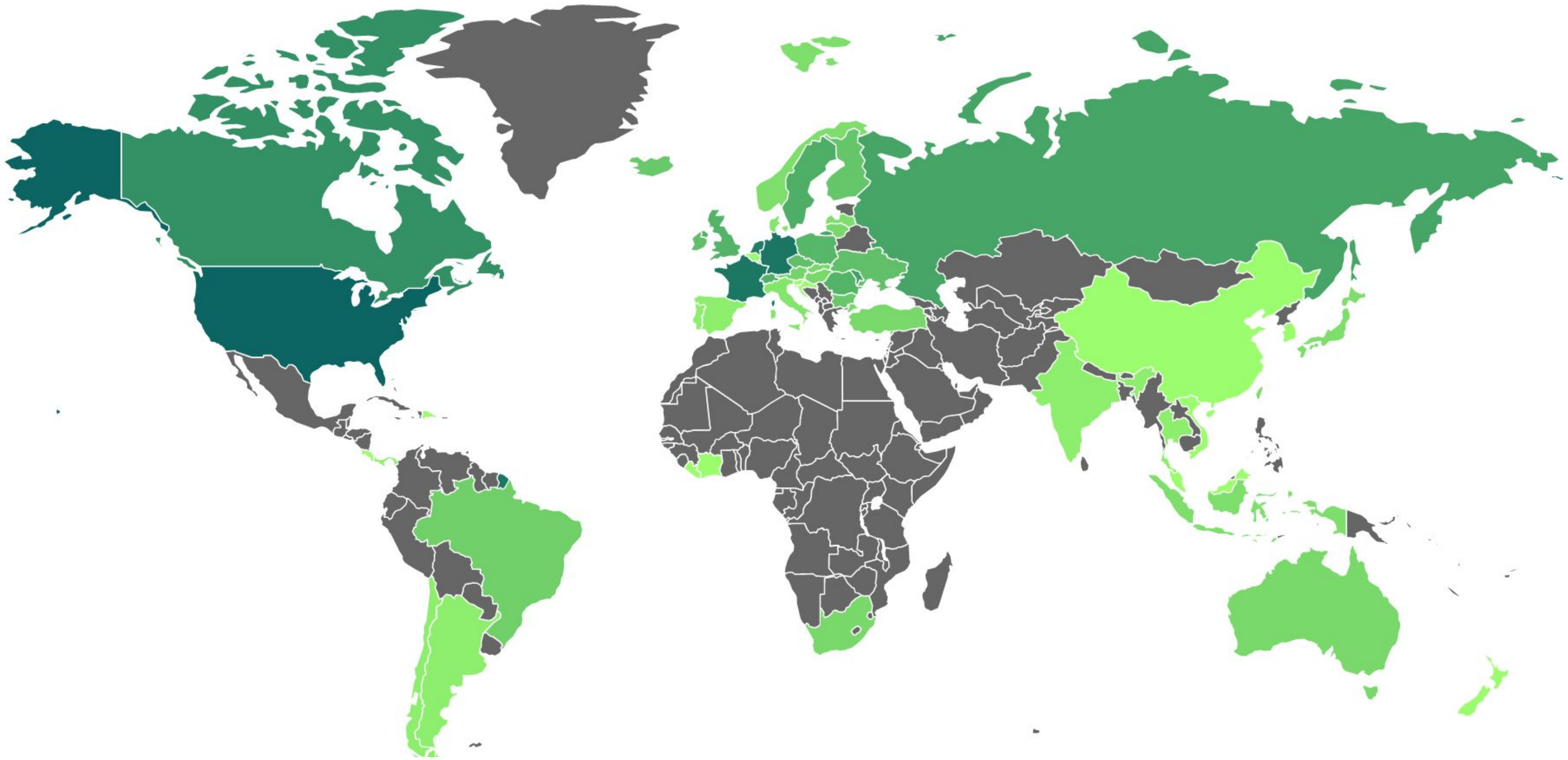
How Does Tor Work?

[Dingledine et al., 2004]

Once circuit is established, applications connect and communicate over Tor circuit



ToR Exit Nodes Mapped



Attacks in practice

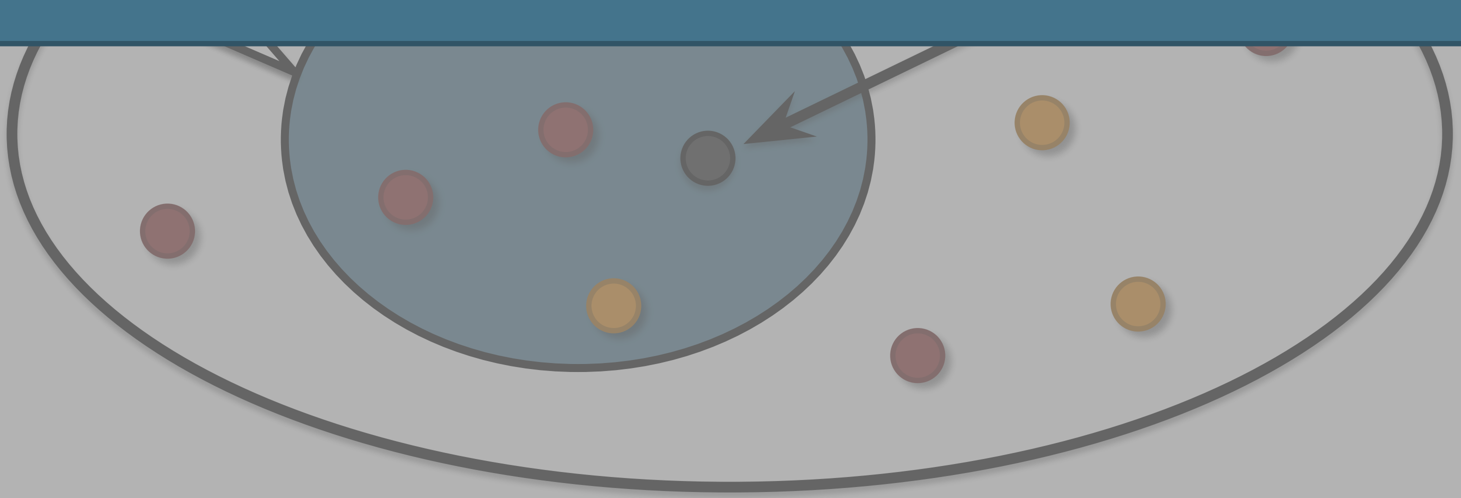
- Traffic patterns watermark sender and receiver
- Sybil attack: attacker runs their own Tor nodes, hoping traffic goes through them
- And more…

Overall, Tor has had a hard time providing anonymity, especially against nation-states. Use with caution.

A Simple Thought Experiment

- Imagine a perfect algorithm for analyzing control flow
  - Guarantees a program always follows intended control flow
- Does this suffice to bootstrap trust? *No!*

We want code *identity*



What Is Code Identity?

- An attempt to capture the behavior of a program
- Current state of the art is the collection of:
  - Program binary
  - Program libraries
  - Program configuration files
  - Initial inputs

Function *f*

Inputs to *f*

- Often condensed into a hash of the above
  - Typically called a *measurement*

Jargon

Code Identity as Trust Foundation

- From code identity, you may be able to infer:
  - Proper control flow
  - Type safety
  - Correct information flow

    …
- Reverse is not true!

What Can Code Identity Do For You?

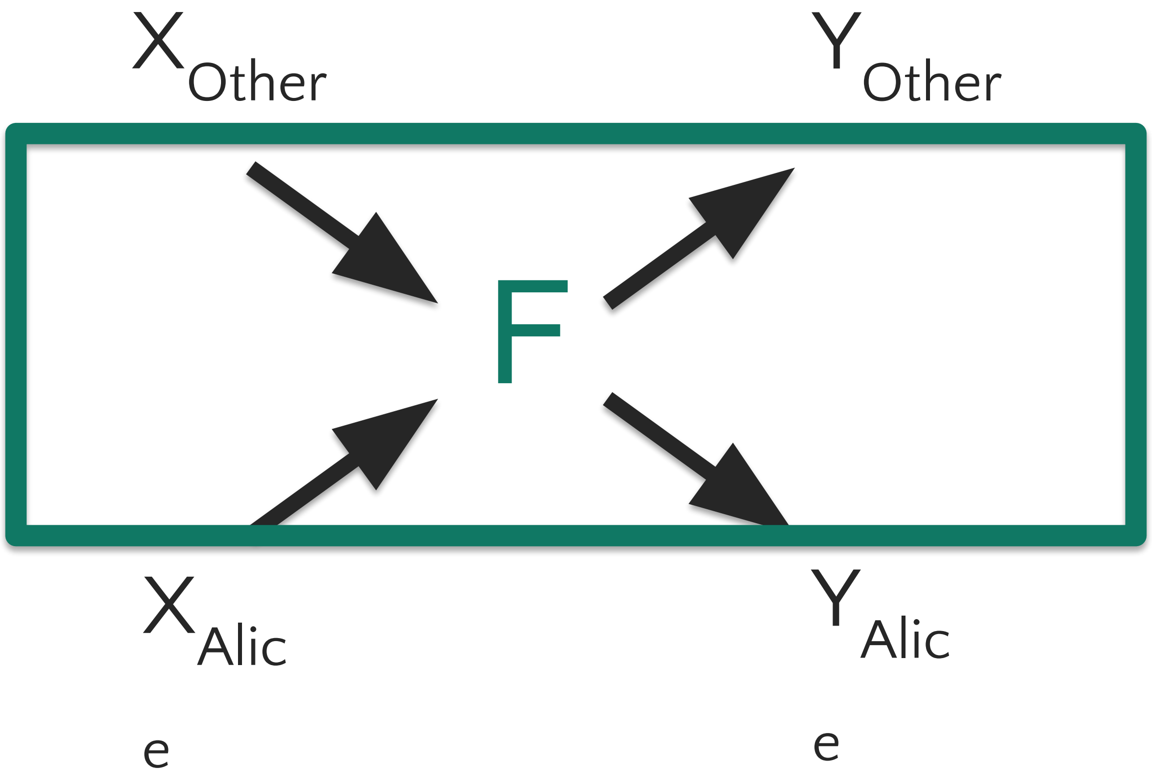
- Research applications:
  - Count-limit objects
  - Improve security of network protocols
  - Thwart insider attacks
  - Protect passwords
  - Create a Trusted Third Party
- Commercial applications:
  - Secure disk encryption (e.g., Bitlocker)
  - Improve network access control
  - Secure boot on mobile phones (e.g., iPhones) and laptops (e.g., Chromebooks)
  - Validate cloud computing platforms

Threat Model

- Network
  - Attacker has complete control over the network (read, intercept, inject messages)
- System
  - Attacker can modify software on disk, reset machine
  - Attacker cannot break hardware protections
- Cryptography
  - Attacker cannot break cryptography
    - Hash function is collision resistant
    - Signatures are unforgeable

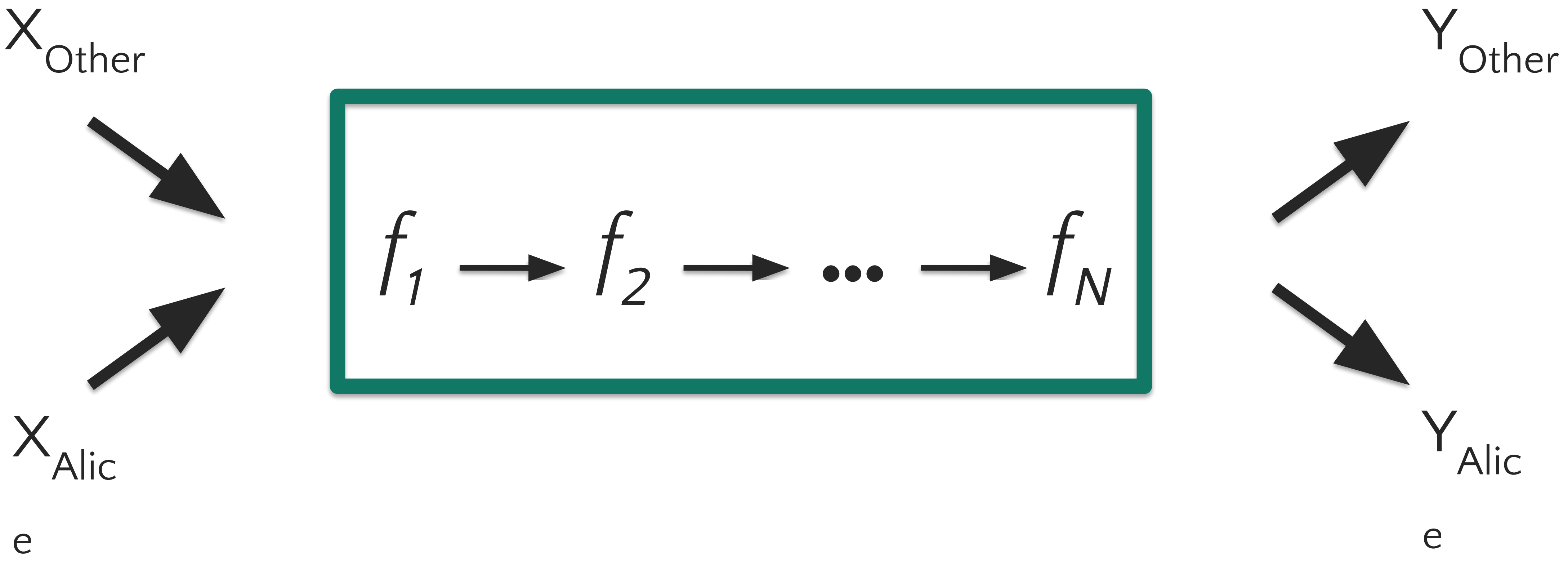
Establishing Code Identity

[Gasser et al. ‘89], [Arbaugh et al. ‘97], [Sailer et al. ‘04], [Marchesini et al. ‘04],…



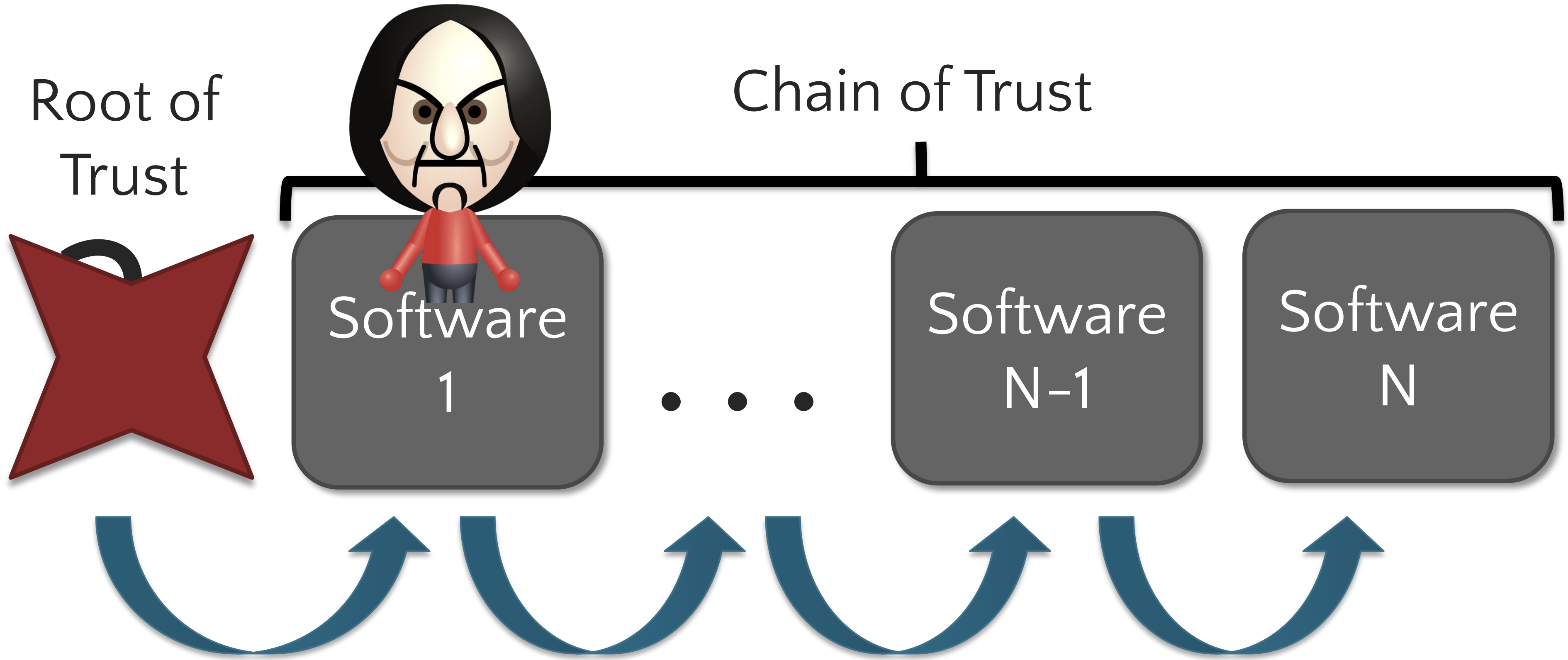
Establishing Code Identity

[Gasser et al. ‘89], [Arbaugh et al. ‘97], [Sailer et al. ‘04], [Marchesini et al. ‘04],…



Establishing Code Identity

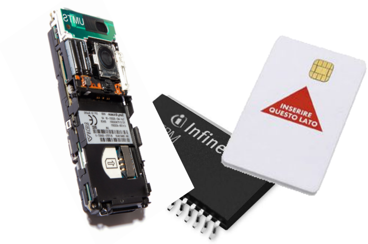
[Gasser et al. ‘89], [Arbaugh et al. ‘97], [Sailer et al. ‘04], [Marchesini et al. ‘04],…



Roots of Trust

- General purpose
- Tamper responding

[Weingart '87]
[White et al. '91]
[Yee '94]
[Smith et al. '99]
...

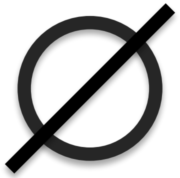


- General purpose
- Limited physical defenses

[ARM TrustZone '04]
[TCG '04]
[Zhuang et al. '04]
...

- Special purpose

[Chun et al. '07]
[Levin et al. '09]

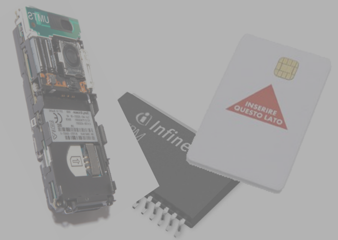


- Timing-based attestation
- Require detailed HW knowledge

[Spinellis et al. '00]
[Seshadri et al. '05]
...

Cheaper →

Roots of Trust



∅

- General
- Tamper

- Timing-based
ion
detailed
wledge

Open Question:

What functionality do we need in hardware?

[Weingart '
[White et al. '91]
[Yee '94]
[Smith et al. '99]
...

[TCG '04]
[Zhuang et al. '04]
...

[Levin et al. '09]

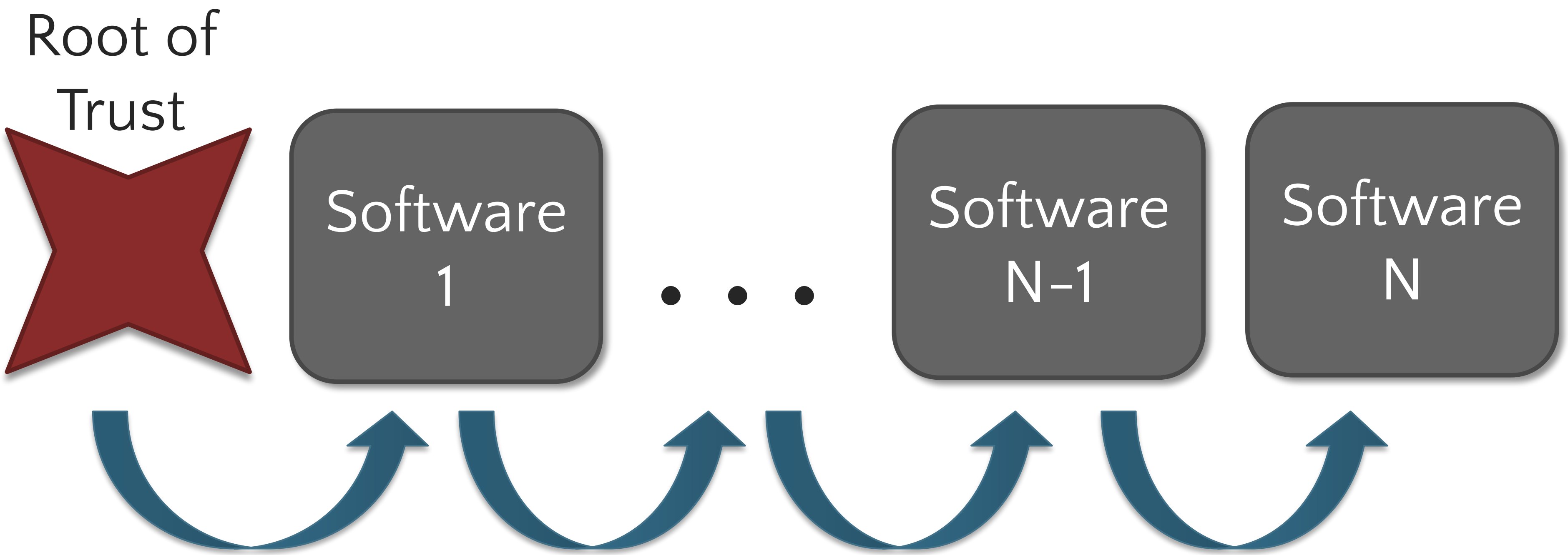
et al. '00]
[Seshadri et al. '05]
...

Cheaper →

Securely Recording Code Identity

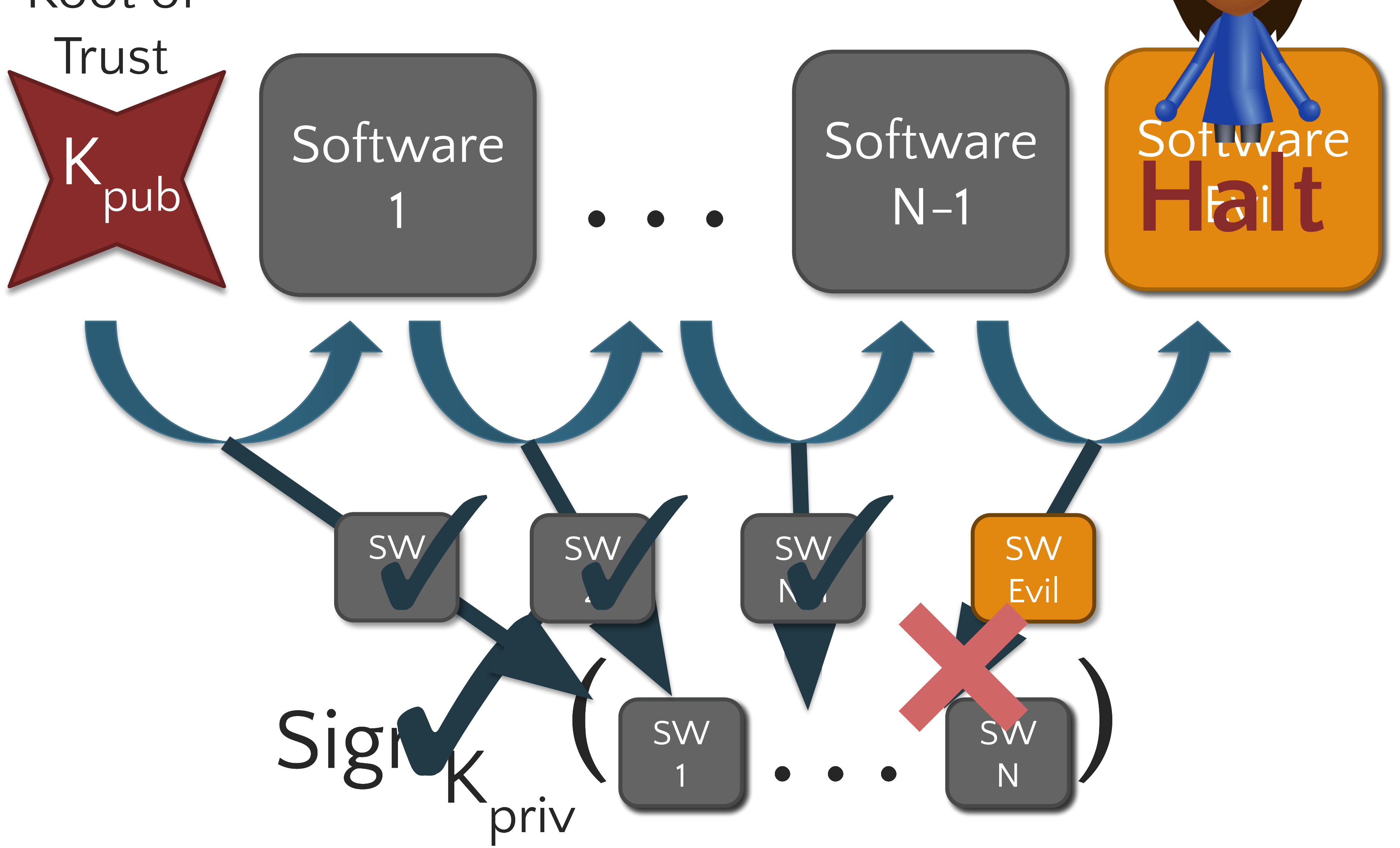
Secure Booting Based on Code Identity

[Gasser et al. ‘89], [Arbaugh et al. ’97], …



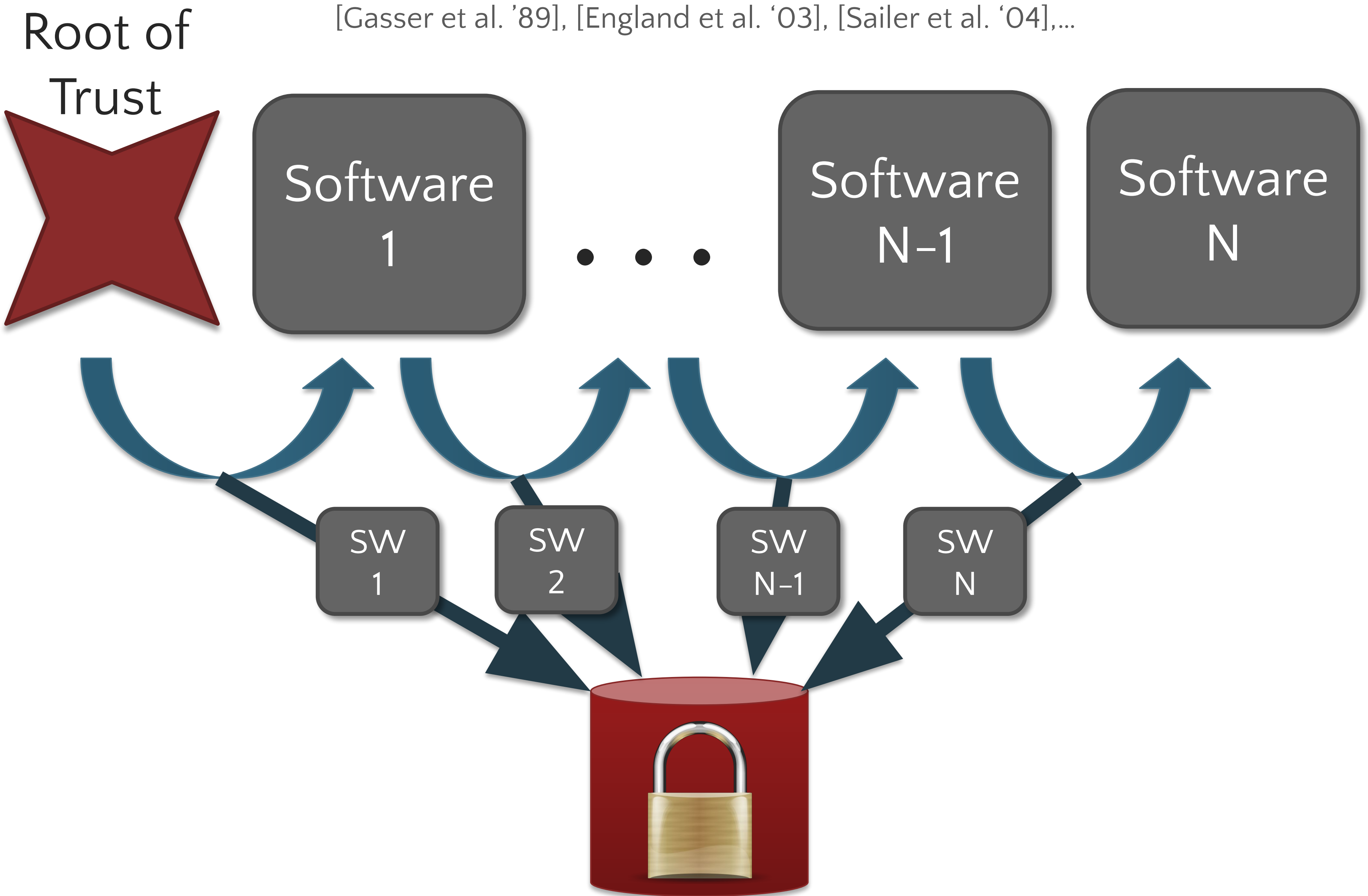
Secure Booting Based on Code Identity

[Gasser et al. ‘89], [Arbaugh et al. ’97], …



Trusted Boot: Recording Code Identity

[Gasser et al. '89], [England et al. '03], [Sailer et al. '04],...



How Can We Secure the Records?

<u>*Certificate Chains*</u>

1. Software X generates key pair for software X + 1

   Generate: $K^{X+1}_{pub}, K^{X+1}_{priv}$

2. Software X signs the new key & code's identity

   $Sign_{K^{X}_{priv}}\left(K^{X+1}_{pub}, \text{SW X+1}\right)$

   Prevents handoff attack

   Most recent must be stored in HW

3. Software X deletes its private key

4. Software X launches software X + 1

Easy to create secure channel to X + 1

How Can We Secure the Records?

Hash Chains

- Software X hashes new code with previous value

$$V_{N+1} \leftarrow \text{Hash}\left(V_N \;\; \text{SW}_{X+1}\right)$$

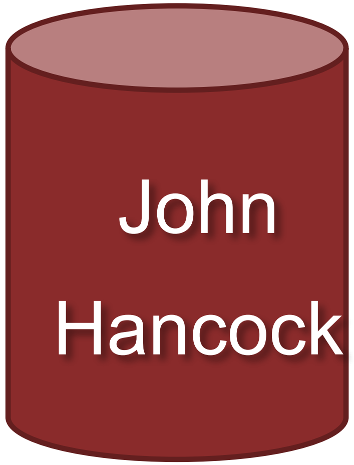
Most be stored & updated in HW

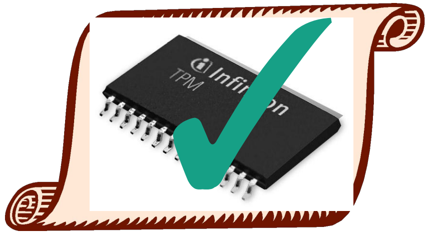
More efficient to compute hashes!

Hardware-Supported Logging Example

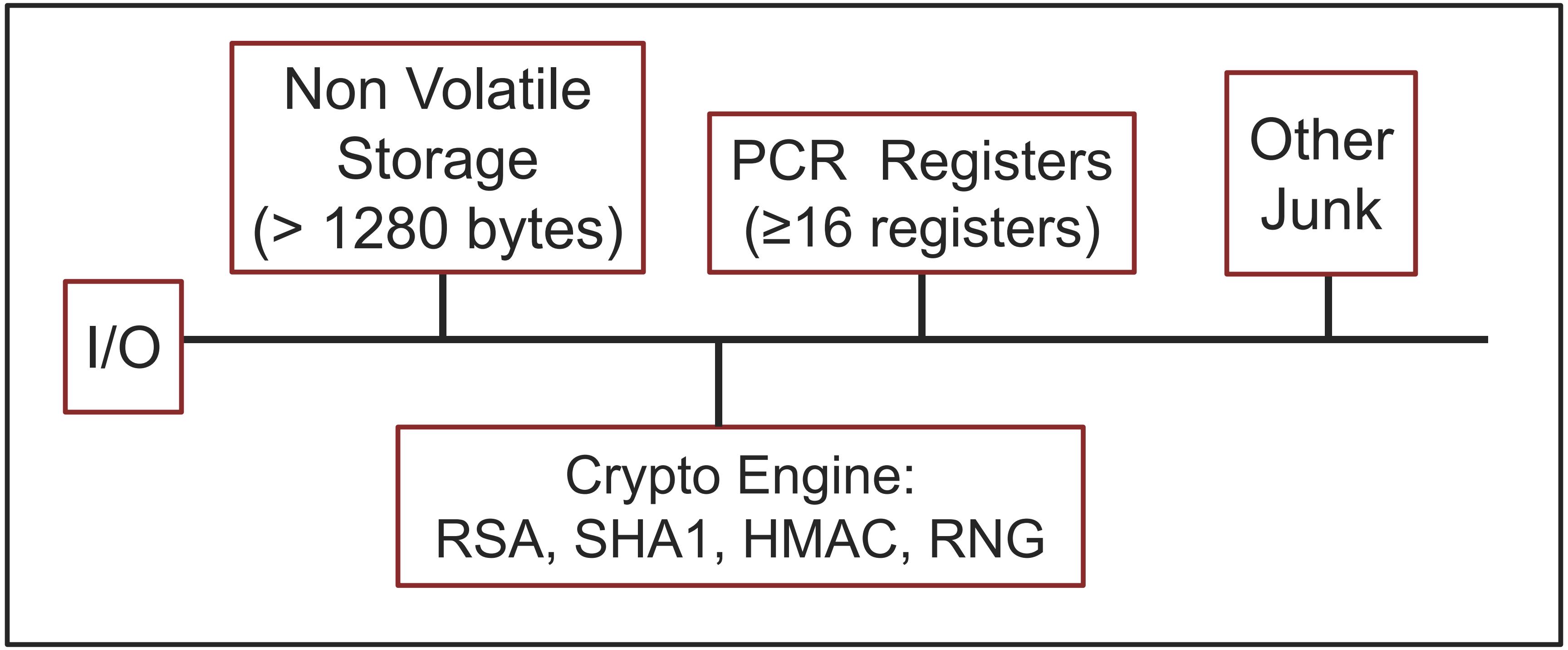
Trusted Platform Module (TPM)

- Provides integrity for append-only logs
- Can digitally sign logs
- Equipped with a certificate of authenticity





Components on (Example) TPM Chip



RSA: 1024, 2048 bit modulus

SHA1: Outputs 20 byte digest

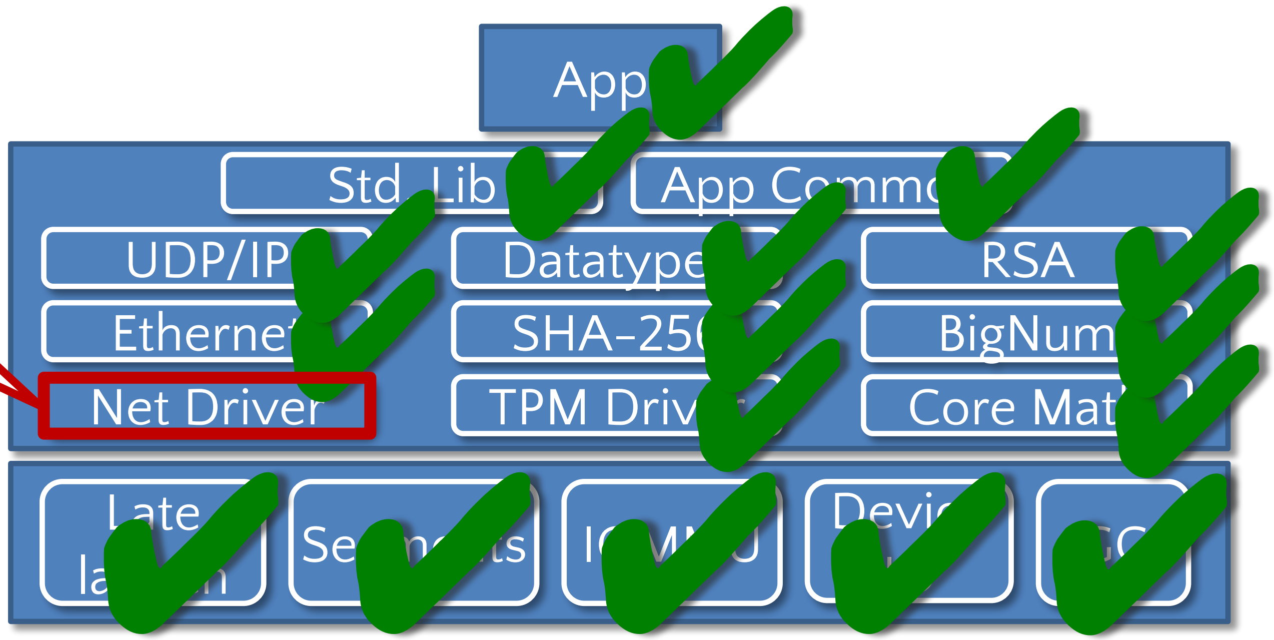
Verification

Imagine a World…

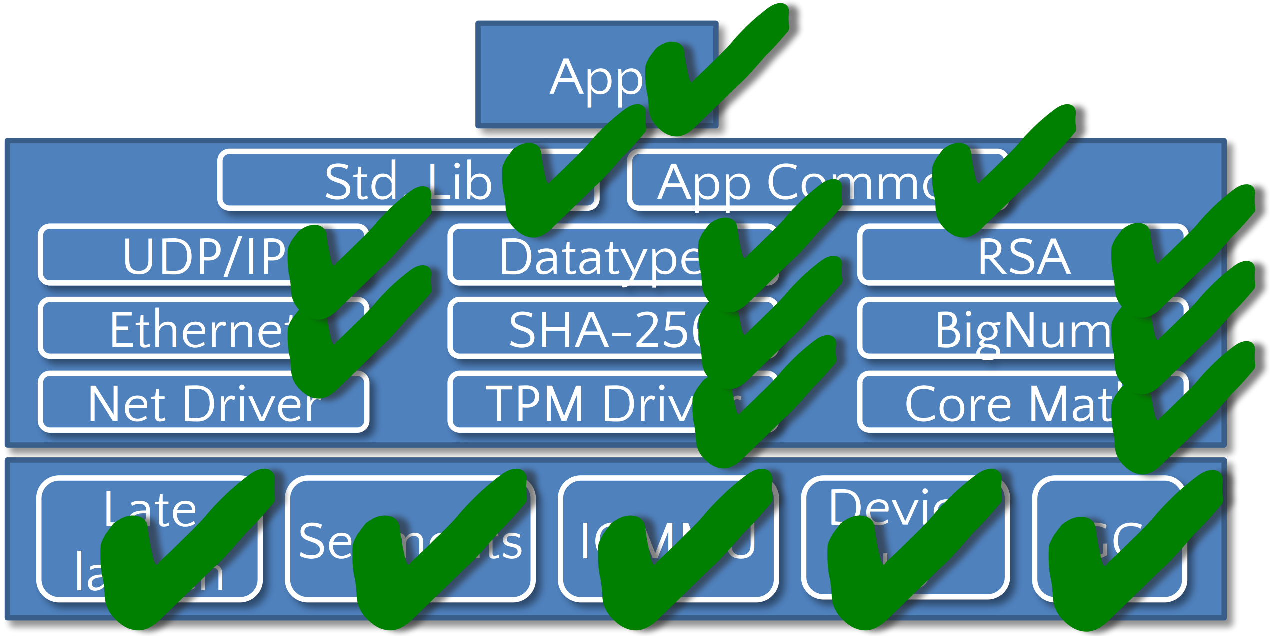
- code worked the 1st time you ran it
- where code was proven to be
  - correct
  - secure
  - reliable
- at compile time!

Verification Can Make This Happen!

1st Version: Secure, but non-functional



Does Verification Sound Too Good to Be True?



Halting Problem

"does program *P* halt given input *i*?"

Can we build a program that computes
halts(P, i) → *bool*
for all *P* and *i*?

Theorem: *halts(P, i)* is undecidable

i.e., there exists no program *P'* that can always compute *halts(P, i)*

Halting Problem

Theorem: *halts(P, i)* is undecidable

Implication for program analysis:

Program analysis, in general, ***cannot*** be both sound and complete

Program analysis works around this by:

- allowing unsoundness/incompleteness
- imposing restrictions on P or properties supported
- requiring human assistance

Is This Program Correct?

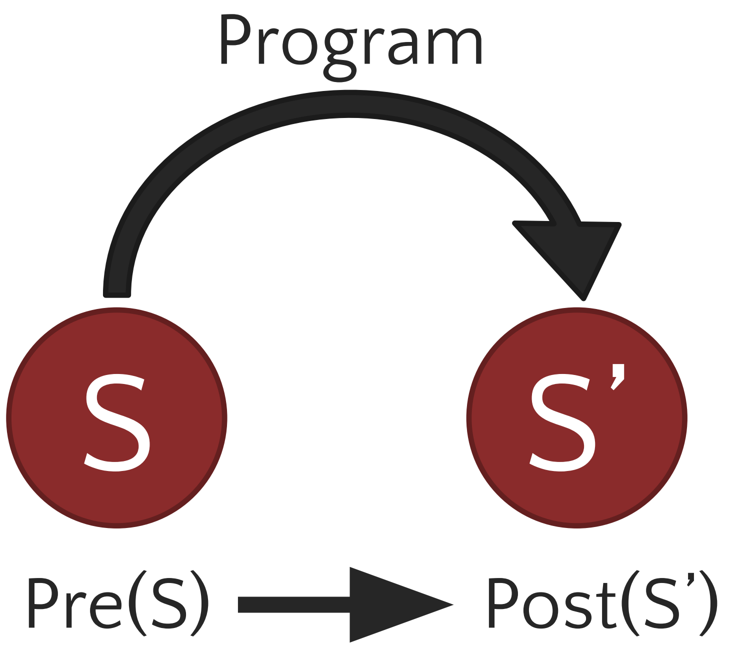
```
method find_elt(elt:int, elts:array<int>)
returns (index:int, found:bool)
{
  var i:int := 0;
  while (i < |elts|) {
    if (elts[i] == elt) {
      index := i;
      found := true;
      return;
    }
    i := i + 1;
  }
  found := false;
}
```

Participation Question

A. Yes
B. No
C. Unsure

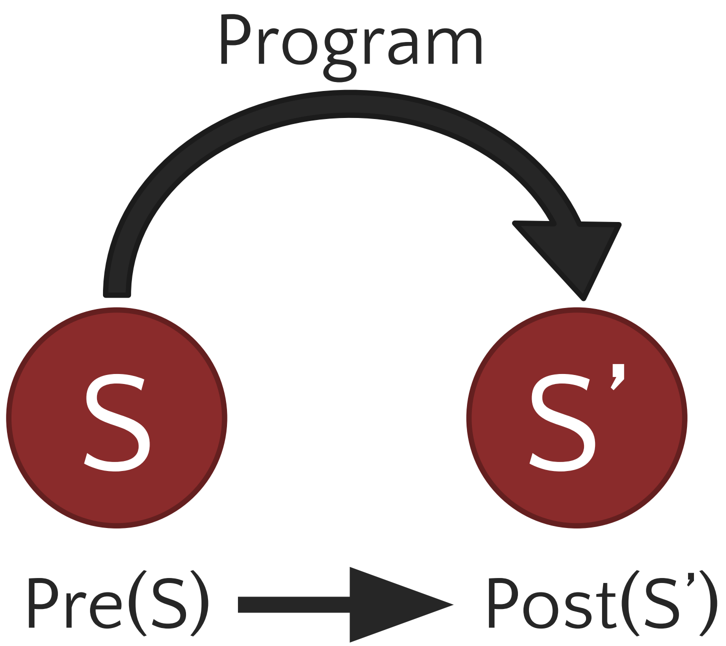
How Would You Argue This Program Is Correct?

```
method find_elt(elt:int, elts:array<int>)
returns (index:int, found:bool)
{
  var i:int := 0;
  while (i < |elts|) {
    if (elts[i] == elt) {
      index := i;
      found := true;
      return;
    }
    i := i + 1;
  }
  found := false;
}
```



How Would You Argue This Program Is Correct?

```
method find_elt(elt:int, elts:array<int>)
returns (index:int, found:bool)
  requires elts != null;
  ensures  found ==> 0 <= index < elts.Length && elts[index] == elt;
  ensures !found ==>
             forall i :: 0 <= i < elts.Length ==> elts[i] != elt;
{
  var i:int := 0;
  while (i < elts.Length) {
    if (elts[i] == elt) {
      index := i;
      found := true;
      return;
    }
    i := i + 1;
  }
  found := false;
}
```

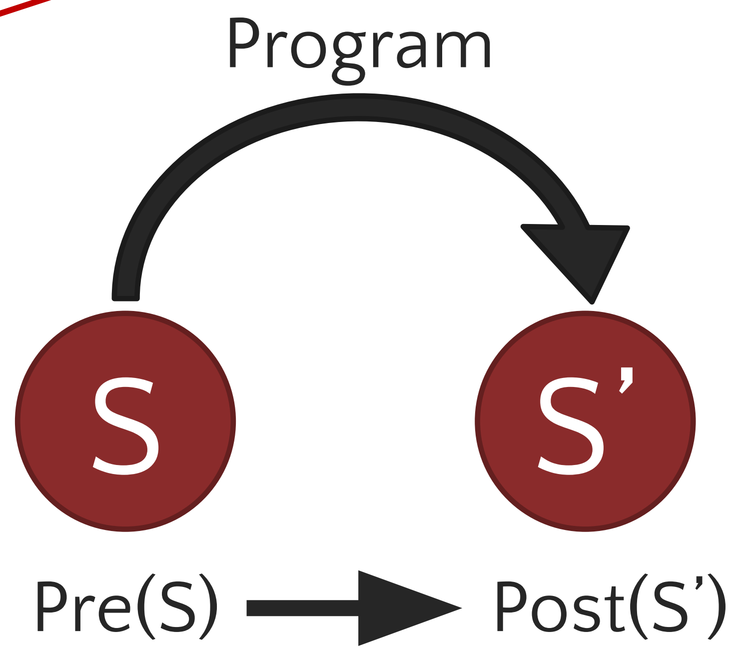


Hoare Logic

[Floyd, 1967], [Hoare, 1969]

- Formal reasoning about program correctness using pre- and post-conditions
- Syntax: {*Pre*} Program {*Post*}
  - *Pre* and *Post* are predicates over program state
- "If we start executing Program when *Pre* is true, it will terminate and *Post* will be true"

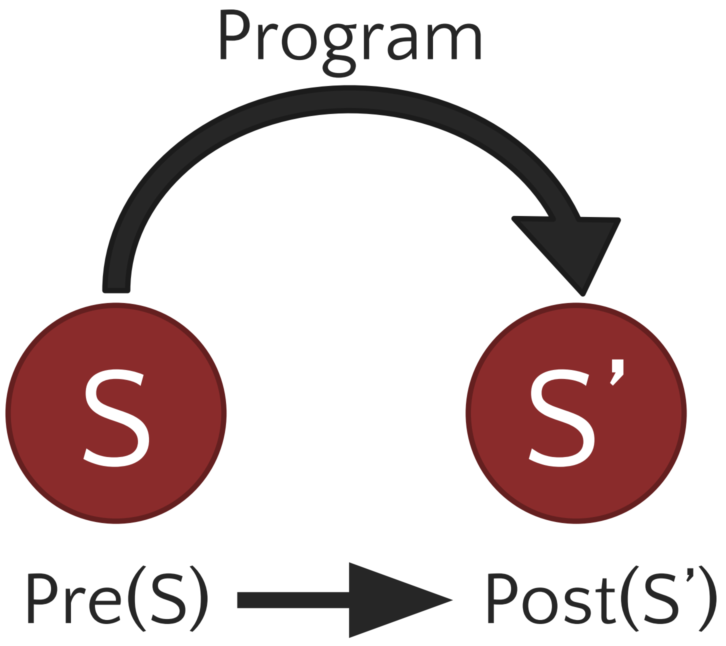
Hoare Triple



Hoare Triple Examples

{Pre} Program *{Post}*

1. *{ true }* **x := 5** *{ x == 5 }*
2. *{ x = y }* **x := x + 3** *{ x == y + 3 }*
3. *{ x = a }* **if x < 0 then x := –x** *{ x = |a| }*
4. *{ false }* **x := 3** *{ x = 8 }*



Weakest Preconditions

[Dijkstra, 1976]

Different triples for the same code:

1) { false } z := x / y { z < 1}
2) { x == 5 && y == 10 } z := x / y { z < 1 }
3) { y != 0 && x / y < 1} z := x / y { z < 1 }

Quiz Question: Which are valid?

A. All
B. Only 2 and 3
C. Only 3
D. None

Ευχαριστώ και καλό καλοκαίρι!

Keep hacking!