Welcome to

Networking Technologies for Cloud Computing

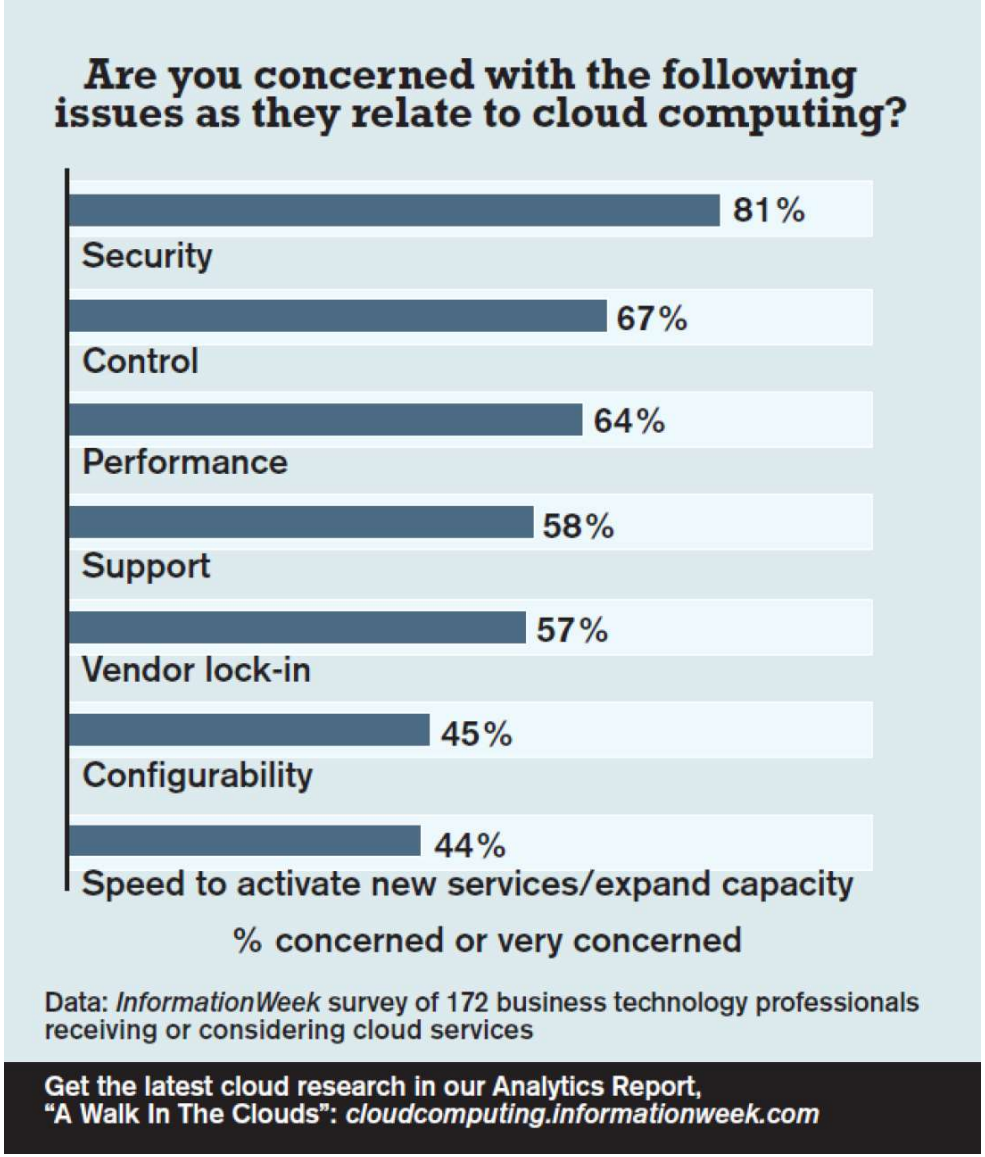
USTC-CYSC6402P
Instructor: Chi Zhang
Fall 2020

Part 7: outline

- S&P Issues for Cloud Computing
- Crypto 2.0
  - Attribute-based Encryption
  - Anonymous Credential
  - Homomorphic Encryption
- PETs
  - PIR/ORAM
  - Differential Privacy
  - Trusted Hardware-SGX

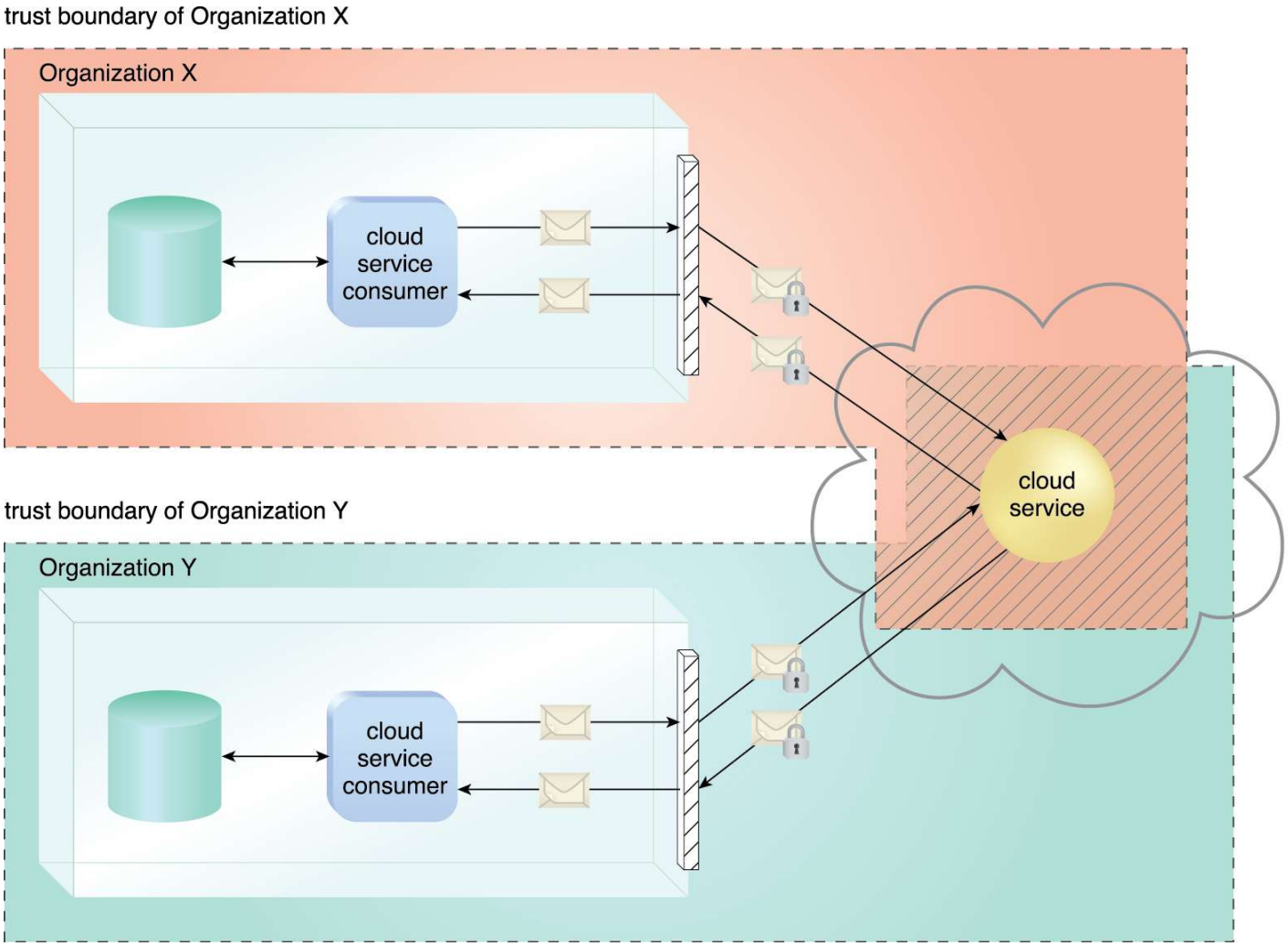
Cloud concerns

- Yep
- Security is No 1 !



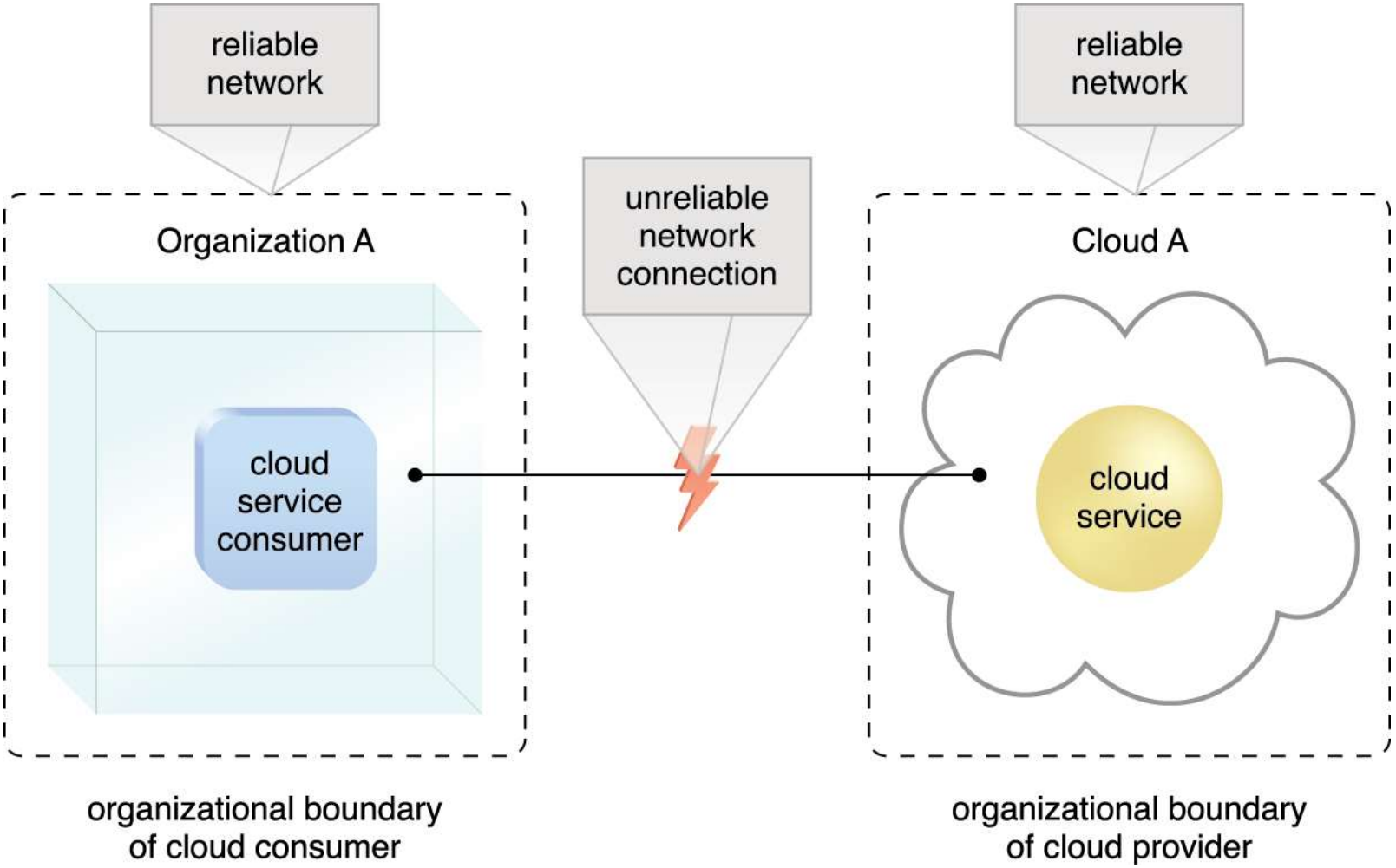
Risks and challenges

- **Increased Security Vulnerabilities**



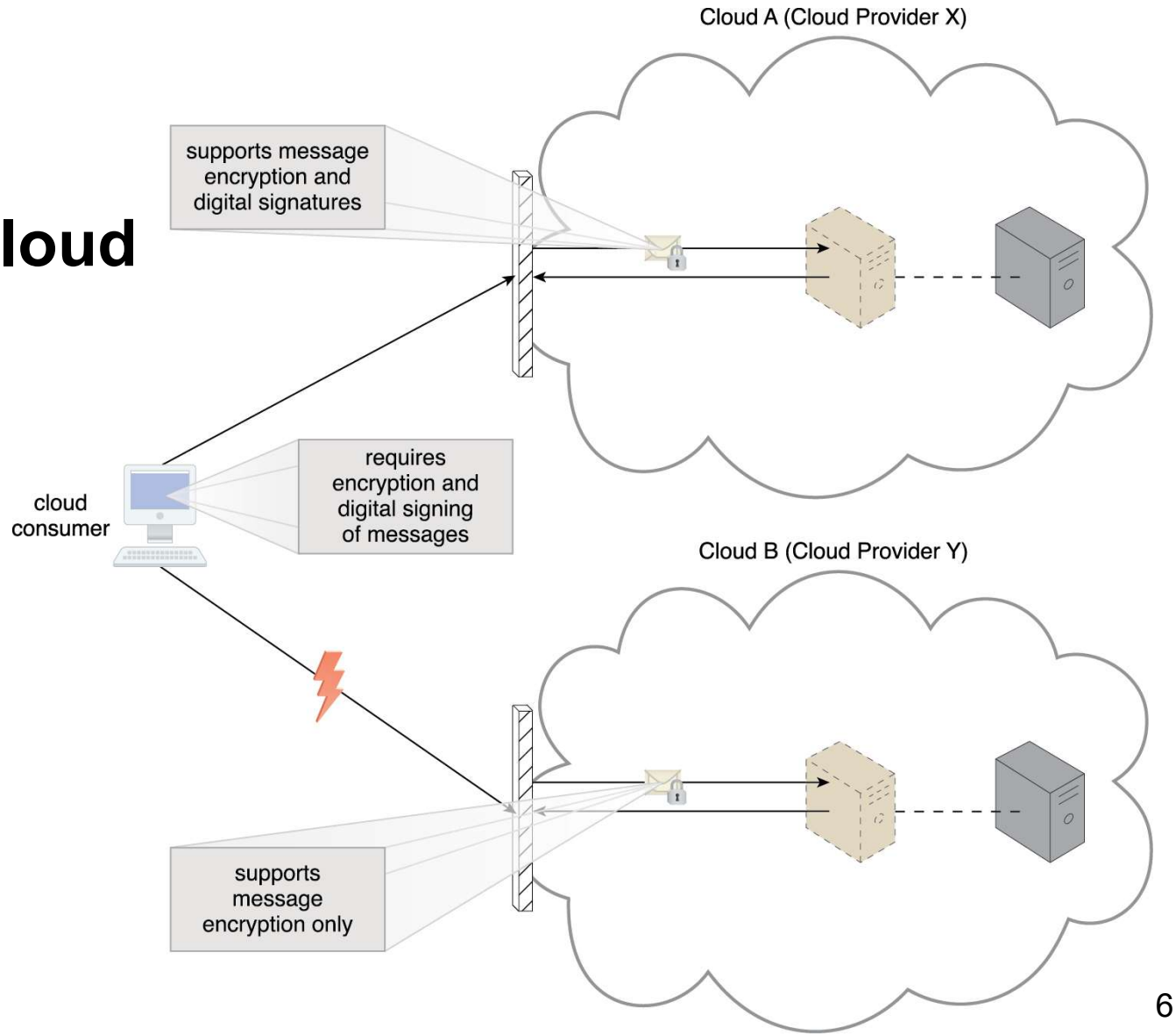
Risks and challenges

- **Reduced Operational Governance Control**



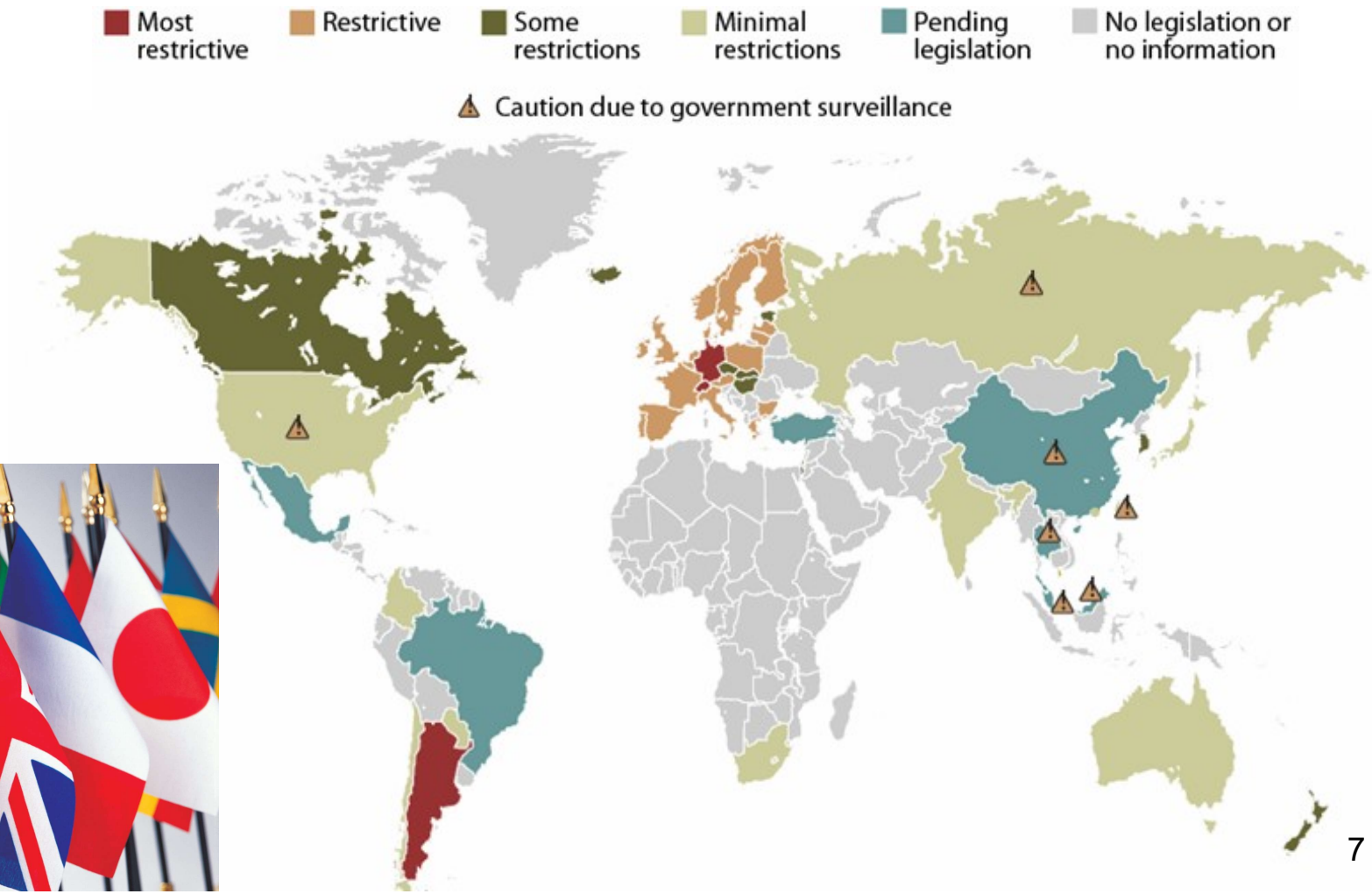
Risks and challenges

- **Limited Portability between Cloud Providers**



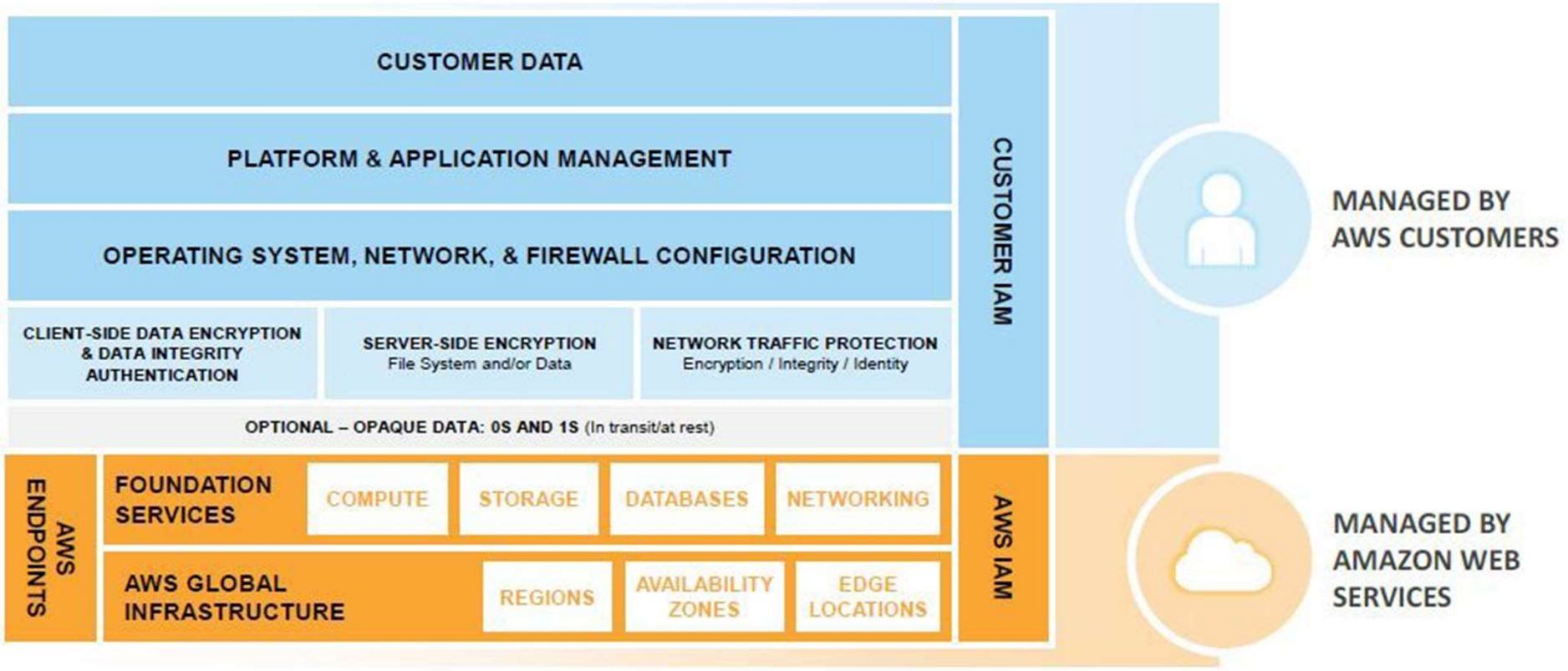
Risks and challenges

- **Multi-Regional Compliance and Legal Issues**



Shared responsibility for security

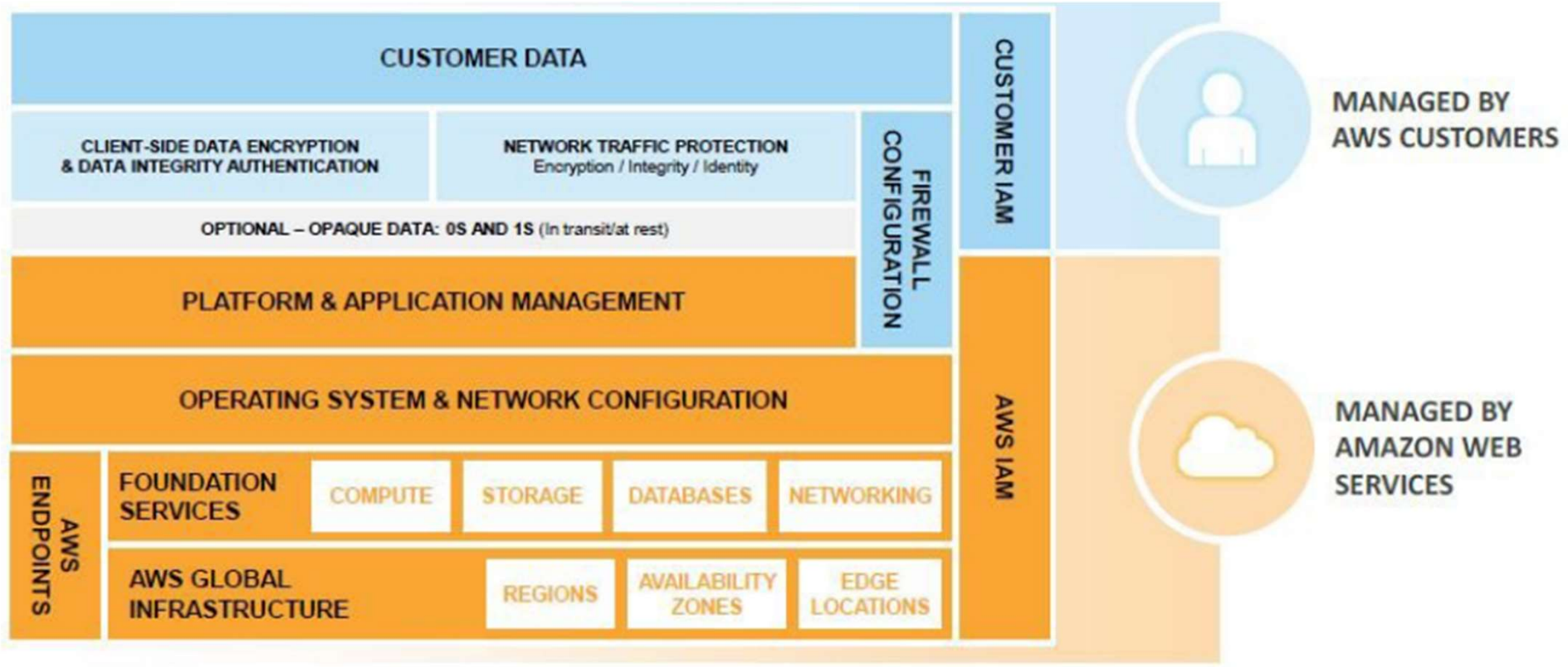
- **AWS Shared Responsibility for Infrastructure Services**



https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/shared-responsibility.html

Shared responsibility for security

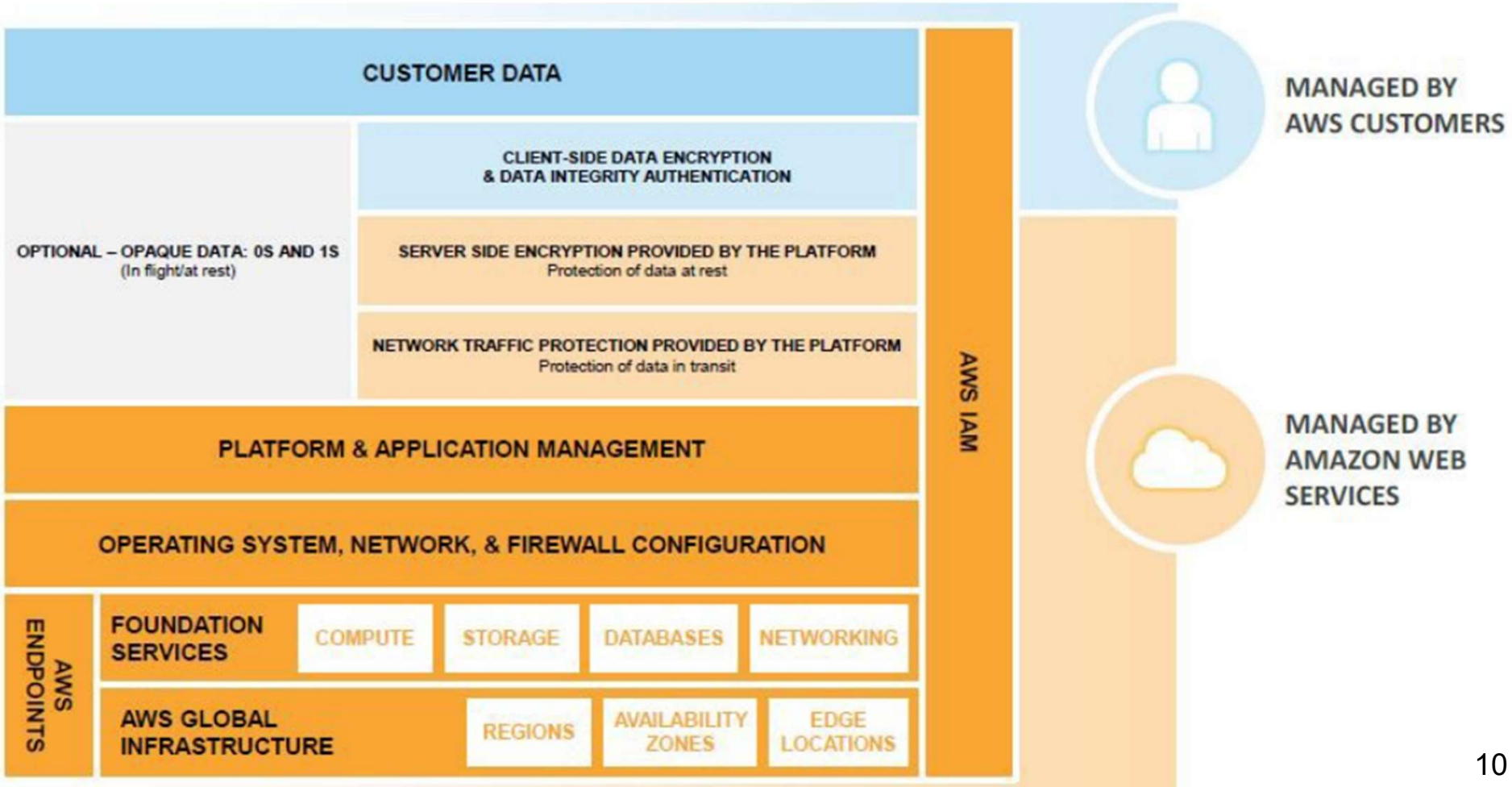
- **AWS Shared Responsibility for Container Services**



https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/shared-responsibility.html

Shared responsibility for security

- **AWS Shared Responsibility for Abstract Services**



https://docs.aws.amazon.com/whitepapers/latest/aws-security-incident-response-guide/shared-responsibility.html

We are naked!


geolocation
| app |
Cellular/Wifi
| server |
geolocation
3D.mesh
3D.data

The privacy puzzle for the cloud

- We have sensors everywhere, including in very sensitive settings.
- They are capturing information you definitely don't want to share.
- … seemingly arguing for brilliant sensors that do all the computing.
  - But sensors are power and compute-limited.
  - Sometimes, only cloud-scale datacenters can possibly do the job!

The virtue of forgetting

The cloud is NOT good on privacy

- Many cloud computing vendors are incented by advertising revenue.
  - Google just wants to show ads that the user will click on.
  - Amazon wants to offer products this user might buy.
- Consider medications: a big business in America. But to show a relevant ad for a drug to treat mental health, or diabetes, entails knowing the user's health status.
- Even showing the ad could leak information that a third party, like the ISP carrying network traffic, might "steal".

The law can’t help (yet)

- Lawrence Lessig: “East code versus West code”.
  - East code is about laws and regulations
  - West code is about new cyber technology ideas
- Main points:
  - The law is far behind the technology curve, in the United States.
  - Europe may be better, but is a less innovative technology community.
  - So our best hope is to just build better technologies here.

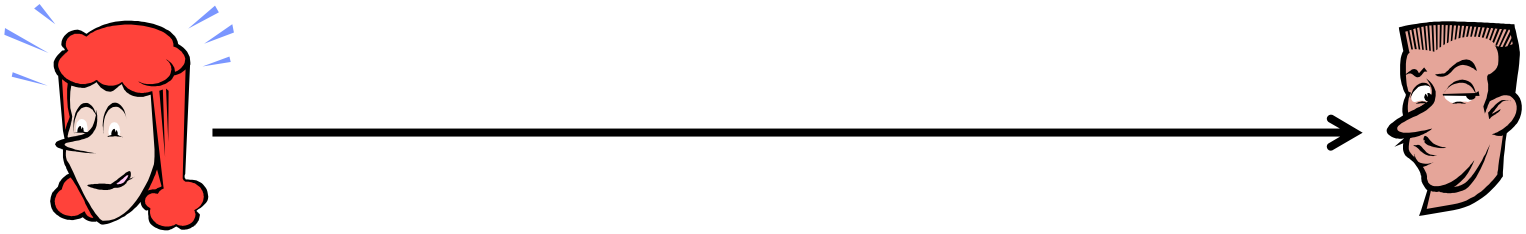
Some providers aren't incented!

- We should separate cloud providers into two groups.
- One group of cloud providers has an inherent motivation to violate privacy for revenue reasons and will "fight against" constraints.
  - Here we need to block their effort to spy on the computation.
- A second group doesn't earn their revenue with ads.
  - These cloud vendors might cooperate to create a secure and private model.

Part 7: outline

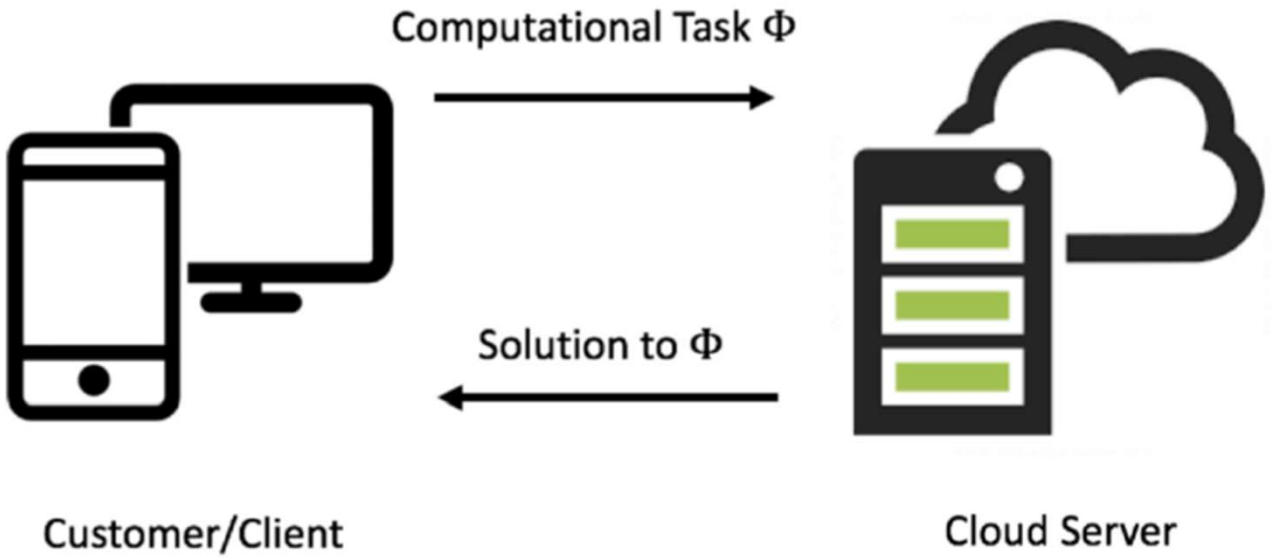
- S&P Issues for Cloud Computing
- Crypto 2.0
  - Attribute-based Encryption
  - Anonymous Credential
  - Homomorphic Encryption
- PETs
  - PIR/ORAM
  - Differential Privacy
  - Trusted Hardware-SGX

Traditional security problems



- 问题1：Alice/Bob：和我通信的真的是Bob/Alice吗？（数据原发性）
- 问题2：Bob：我收到的信息是Alice发给我的原始信息吗？有没有被人篡改过？（数据完整性）
- 问题3：Alice/Bob：我和Bob/Alice的通信内容有没有被别人窃听到？（数据保密性）

New security problems

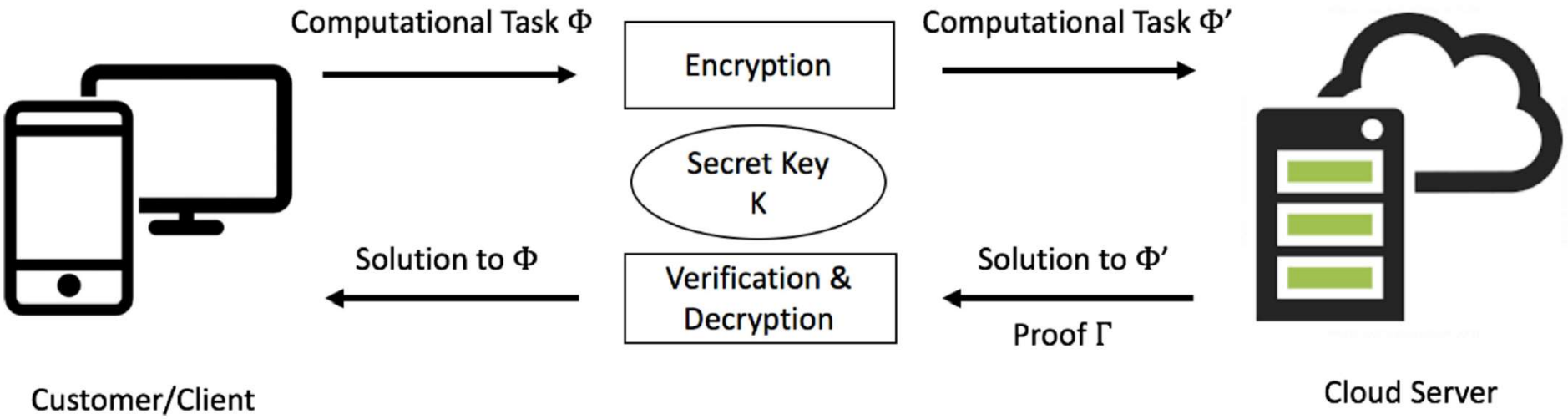


- 云服务器被黑客攻破了怎么办？我们放在云上的（敏感）数据也就泄露了。
- 使用提供加密服务的云服务？云服务商（或其内部员工）本身变成恶意（好奇的），偷看我们的数据怎么办？

Crypto 1.0 vs Crypto 2.0

- Crypto 1.0 encryption and authentication:
  - protect against malicious outsiders:
    ---attacks on storage or communication media
  - can be achieved using symmetric crypto
  - very high performance
- Crypto 2.0 primitives additionally:
  - protect against malicious or corrupt insiders:
    ---attacks by your protocol “partners”
  - uses more powerful cryptographic primitives, essentially asymmetric crypto
  - much harder to do efficiently

Secure computation outsourcing



- Data confidentiality = Client privacy
- Computation integrity = Result verifiability
- Computation Efficiency

Part 7: outline

- S&P Issues for Cloud Computing
- Crypto 2.0
  - Attribute-based Encryption
  - Anonymous Credential
  - Homomorphic Encryption
- PETs
  - PIR/ORAM
  - Differential Privacy
  - Trusted Hardware-SGX

Diffie-Hellman's trick

- Khan Academy's explanation
- https://www.youtube.com/watch?v=MsqqpO9R5Hc

KHAN ACADEMY

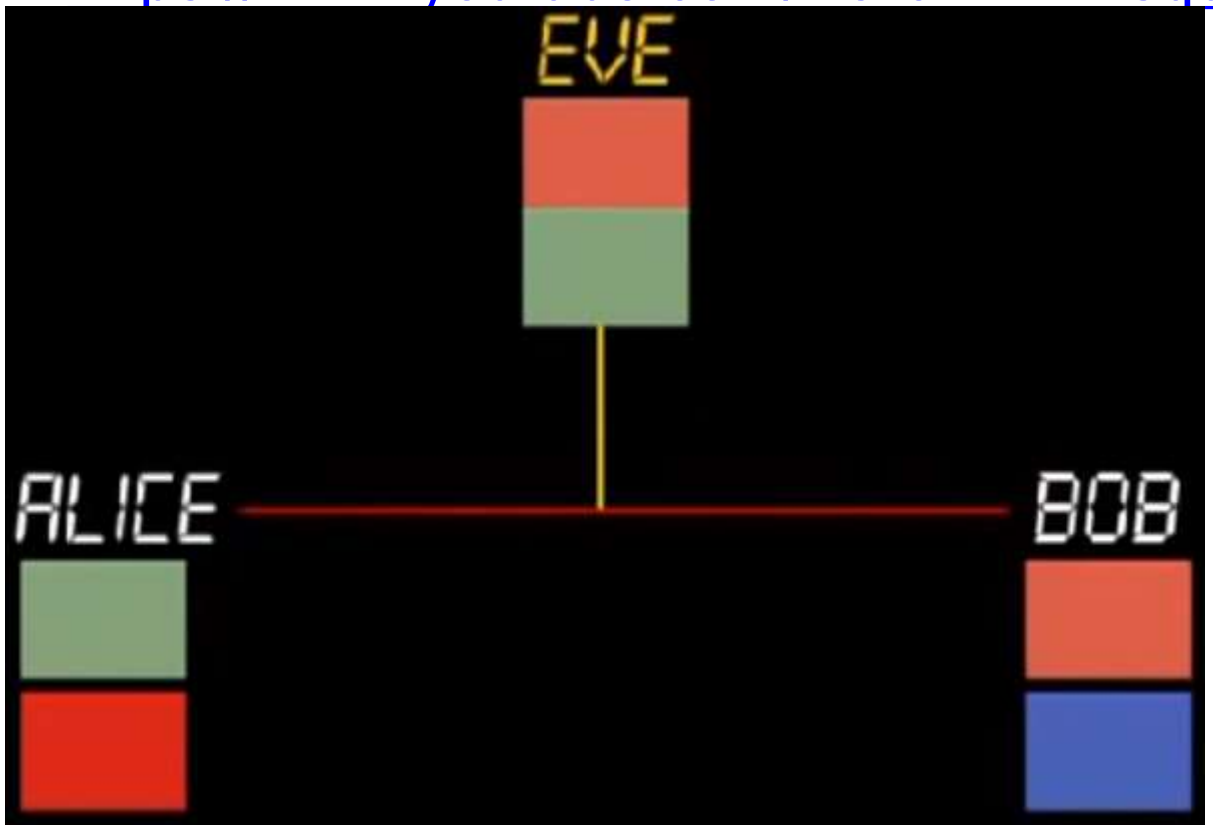
Diffie-Hellman's trick

- Khan Academy's explanation
- https://www.youtube.com/watch?v=MsqqpO9R5Hc

KHAN ACADEMY

Diffie-Hellman's trick

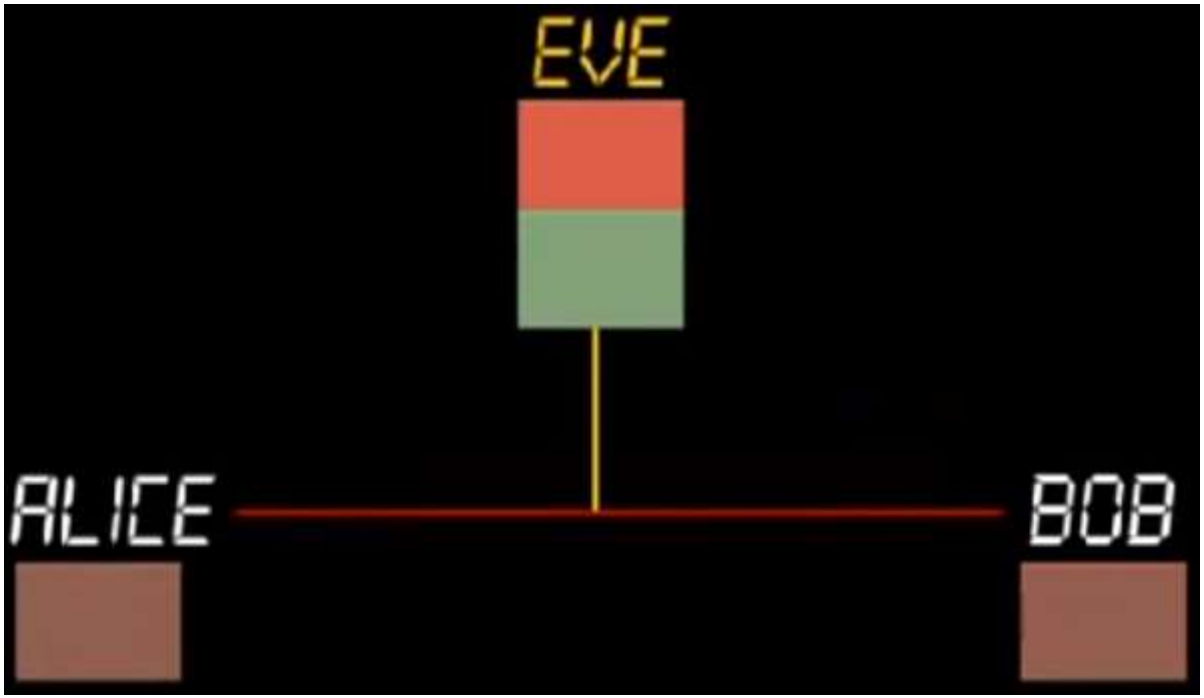
- Khan Academy's explanation
- https://www.youtube.com/watch?v=MsqqpO9R5Hc



KHAN ACADEMY

Diffie-Hellman's trick

- Khan Academy's explanation
- https://www.youtube.com/watch?v=MsqqpO9R5Hc



- One-way function
- Exchangeable order

KHAN ACADEMY

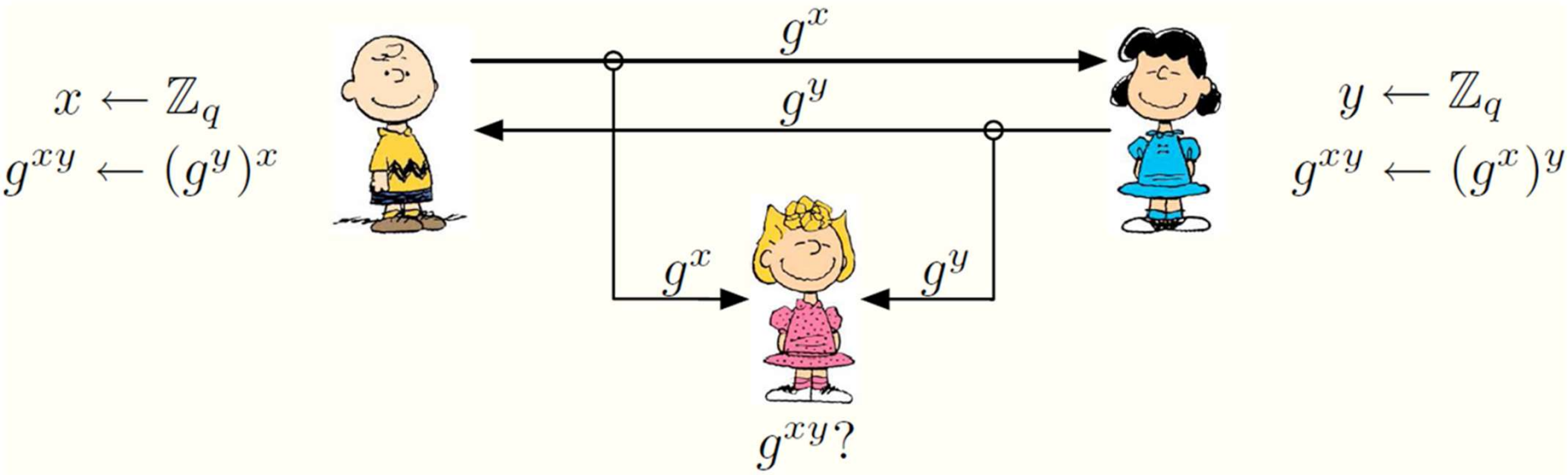
Discrete log related assumptions

Problem	Given	Figure out
Discrete logarithm (DL)	g^x	x
Computational Diffie-Hellman (CDH)	g^x, g^y	g^{xy}
Decisional Diffie-Hellman (DDH)	g^x, g^y, g^z	Is $z \equiv xy \pmod{|G|}$?

- An informal description of three discrete logarithm related problems over a cyclic group *G* with generator *g*. For each problem we indicate the input to the attacker, and what the attacker must figure out to "win."
- One-way function:
  Fast to compute g^λ but difficult to compute λ.

Diffie-Hellman key exchange (1976)

- Diffie-Hellman Key Exchange:



- Session key: g^{xy}

ElGamal encryption (1985)

- ElGamal Encryption:
  - Private key: x; Public key: $h_x = g^x$
  - Encryption: choose a random number r, for plaintext m

    Ciphertext $C(m) = (C_1, C_2) = (g^r, m \bullet (h_x)^r,)$

    Session key (SK): $(h_x)^r = g^{xr}$
  - Decryption: C_1, $X \rightarrow g^{xr}$ (SK); C_2, SK $\rightarrow$ m
- Probabilistic Encryption (random # r)
- Partially Homomorphic Encryption

$$\boldsymbol{C}(m_1) \cdot \boldsymbol{C}(m_2) = (g^{r_1}, m_1 \cdot h^{r_1})(g^{r_2}, m_2 \cdot h^{r_2})$$
$$= (g^{r_1+r_2}, (m_1 \cdot m_2) h^{r_1+r_2}) = \boldsymbol{C}(m_1 \cdot m_2).$$

DL-based crypto

- To use DH in applications, ensure that:
  - legitimate parties only compute linear functions
  - adversary needs to compute/check quadratics

- Example: private key: x; public key: g^x

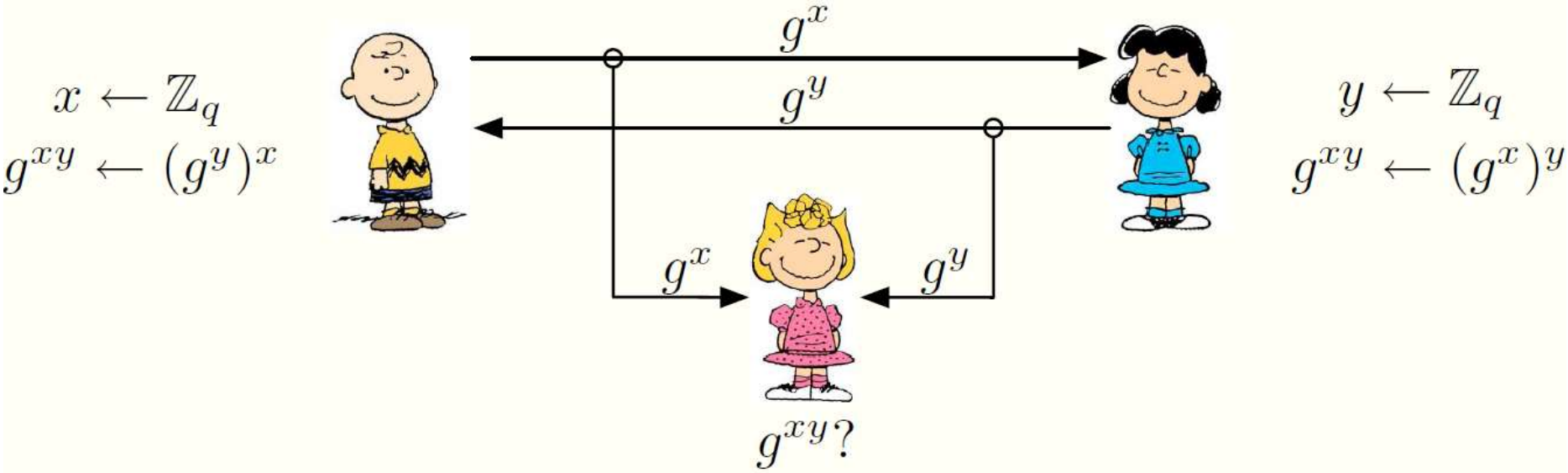
- More examples:
  - Diffie-Hellman key exchange, ElGamal Encryption, Cramer-Shoup CCA-Secure Encryption, Naor-Reingold PRF, Efficient ZKPs, …

DL-based crypto

- Hiding "things" in the exponent and then manipulating those values as they live in the exponent g^x
- All you can do in the exponent are linear (degree-1) combinations of these hidden values. That is, given g^{x1}, g^{x2}, …, g^{xn}, You can obtain $g^{a_0+a_1x_1+a_2x_2+\ldots+a_nx_n}$ for known coefficients a_i

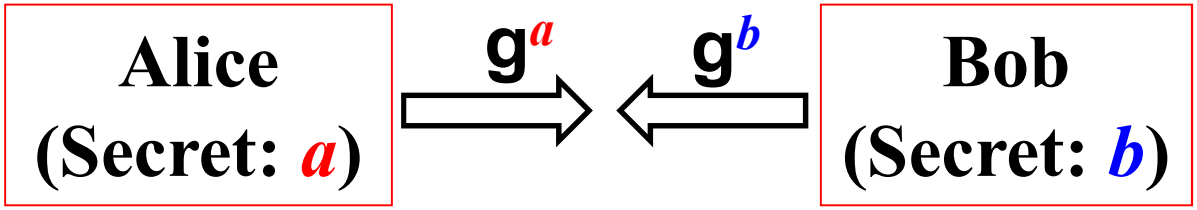
Recall

- Diffie-Hellman Key Exchange
- Only for 2 participants



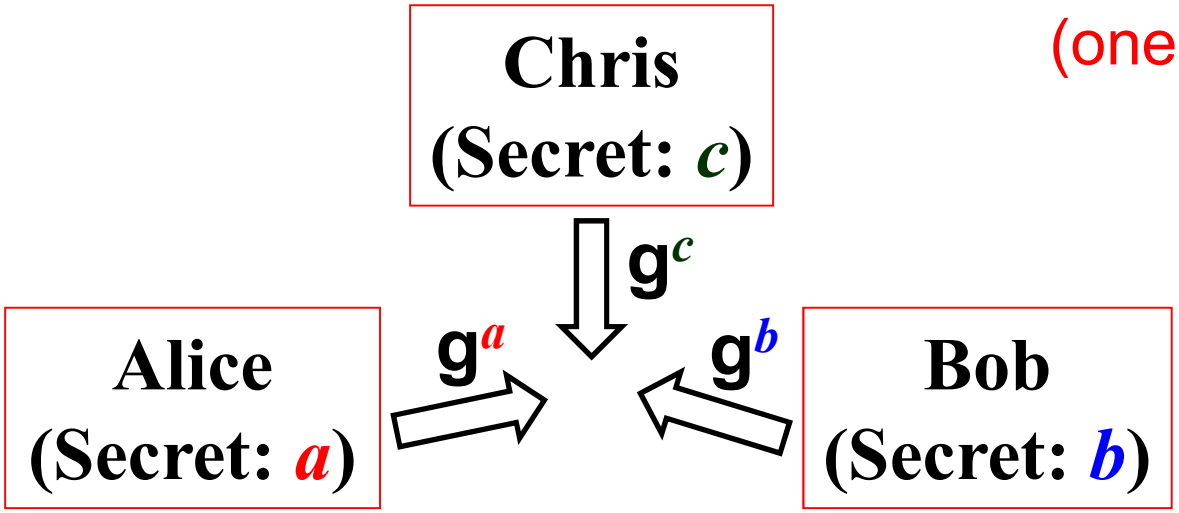
- Session key: g^{xy}
- How about 3 participants?

Multiparty key exchange



- Session key: g^{ab}

One broadcast for each participant!
(one-round condition)



- Session key: g^{abc}? No!

Bilinear pairing

$$e(g^a, h^b) = e(g^b, h^a) = e(g, h)^{ab} = e(g^{ab}, h) = e(g, h^{ab})$$

Let $\hat{g} = e(g, h)$, $e(g, h)^{ab} = \hat{g}^{ab}$

- Symmetric pairing
  - g=h; $e(g^a, g^b) = e(g, g)^{ab}$
- Remember that $g \neq \hat{g}$
  - $e(g^a, g^b) = e(g, g)^{ab} = \hat{g}^{ab}$
  - $e(g^c, \hat{g}^{ab}) = e(g, \hat{g})^{abc}$

Most common new assumptions

Some new problems have been defined and assumed hard in the new bilinear context.

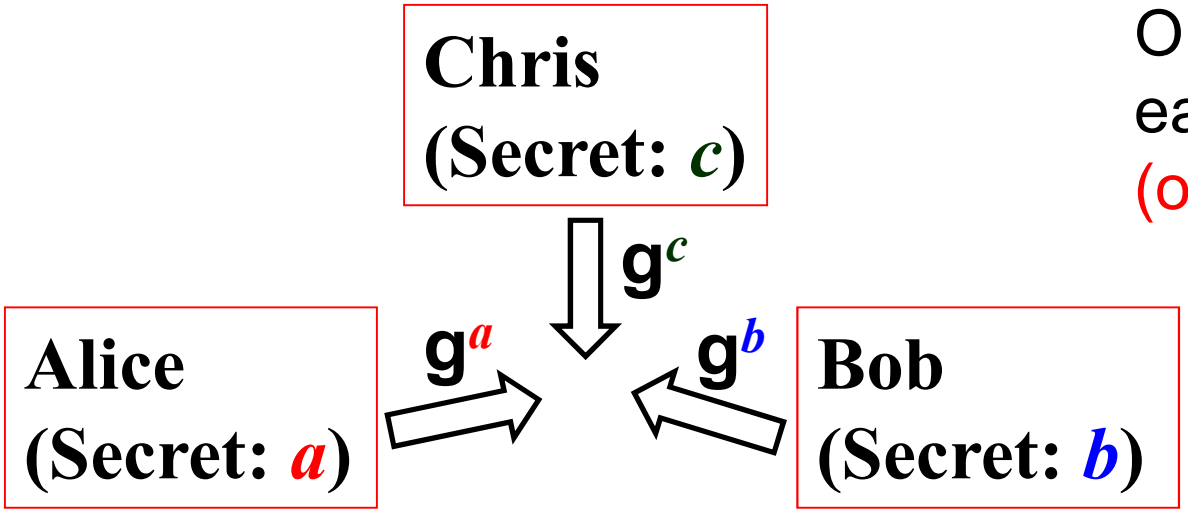
Bilinear Diffie-Hellman Given g, g^a, g^b, g^c, compute $e(g,g)^{abc}$ (something like a "three-way" CDH but across the two groups)

Decisional Bilinear Diffie-Hellman Distinguish $g, g^a, g^b, g^c, e(g,g)^{abc}$ from $g, g^a, g^b, g^c, e(g,g)^z$

k-Bilinear Diffie-Hellman Inversion Given $g, g^y, g^{y^2}, \ldots g^{y^k}$, compute $e(g,g)^{\frac{1}{y}}$

k-Decisional Bilinear Diffie-Hellman Inversion Distinguish $g, g^y, g^{y^2}, \ldots g^{y^k}, e(g,g)^{\frac{1}{y}}$ from $g, g^y, g^{y^2}, \ldots g^{y^k}, e(g,g)^z$

Joux's 3-party Diffie-Hellman



One broadcast for each participant!
(one-round condition)

- Session key: $e(g,g)^{abc} = \hat{g}^{abc}$
- For Alice: $e(g^b,g^c)^a = e(g,g)^{abc}$
- For Bob: $e(g^a,g^c)^b = e(g,g)^{abc}$
- For Chris: $e(g^a,g^b)^c = e(g,g)^{abc}$

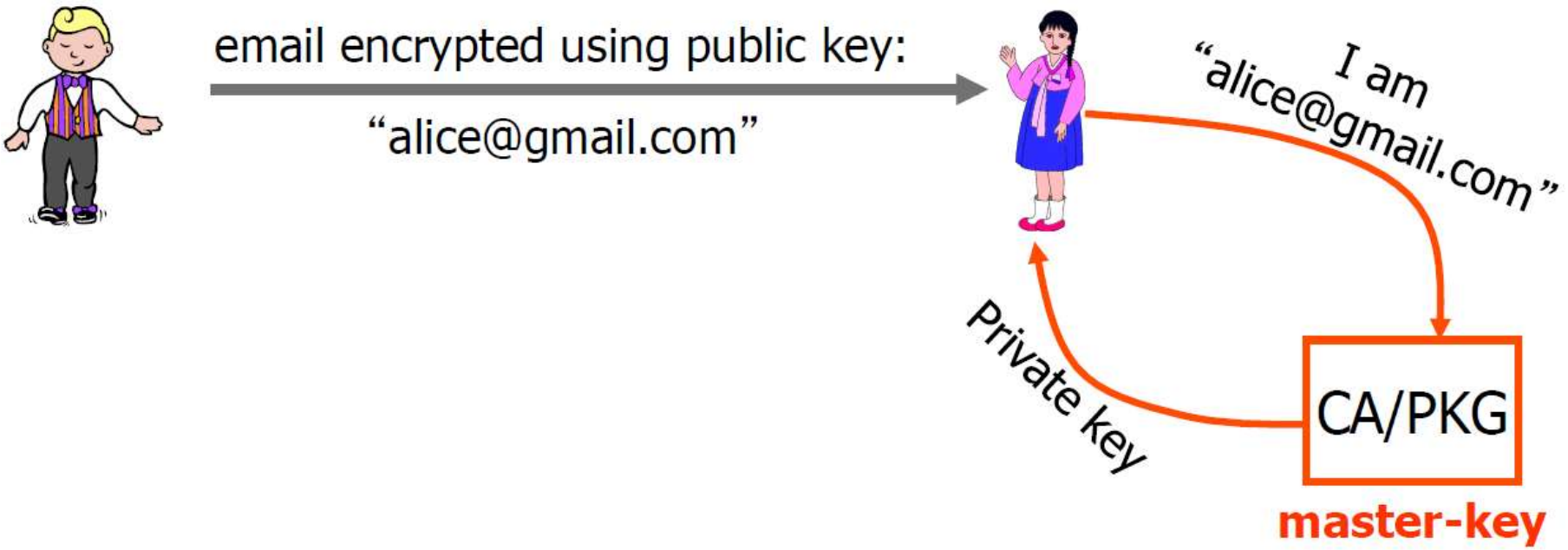
Bilinear Diffie-Hellam: **Given g, g^a, g^b, g^c, compute $e(g,g)^{abc}$**

Intuition

- In DLog groups
  - legitimate parties only compute linear functions
  - adversary needs to compute/check quadratics
- In bilinear-map groups you can compute quadratic functions in the exponent
  - But computing/checking cubics is hard
- Now the legitimate parties can do a lot more, which leads to new capabilities

Identity Based Encryption (IBE)

- IBE: PKE system where PK is an arbitrary string
  - e.g. e-mail address, phone number, IP addr…



Boneh & Franklin's IBE

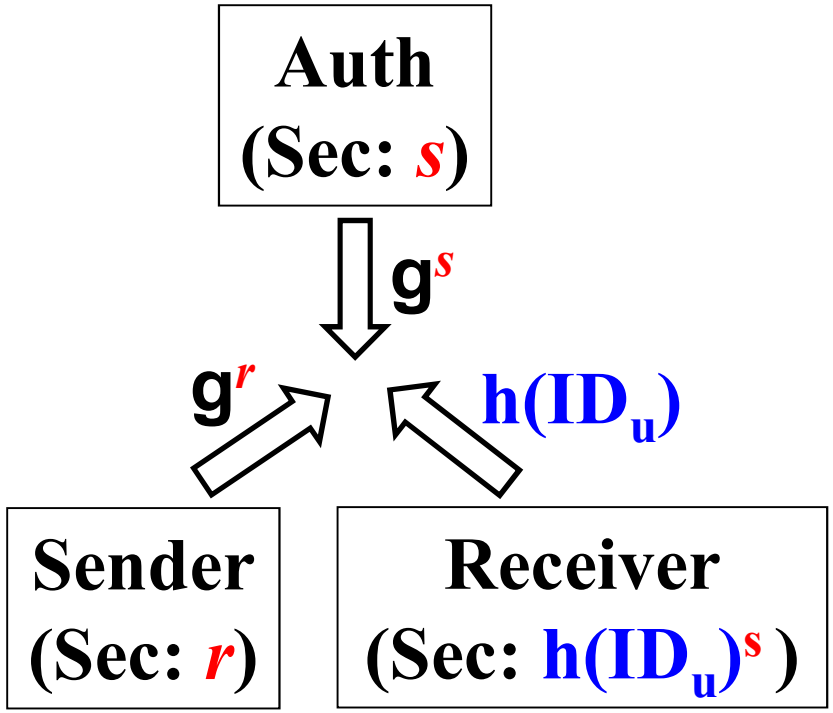
- Authority:
  - Public key: g^s
  - Private key: s (msk)
- User with ID_u:
  - Public key: $h(ID_u)$
  - Private key: $h(ID_u)^s$
- Session key:
  - Encryption key: $e(h(ID_u), g^s)^r = e(h(ID_u), g)^{sr}$; g^r
  - Decryption key: $e(h(ID_u)^s, g^r) = e(h(ID_u), g)^{sr}$

$$\mathbf{C}(m) = (\mathbf{C}_1, \mathbf{C}_2) =$$

$$(g^r, m \oplus h_2(e(h_1(\text{“Bob”}), g^s)^r)$$

Boneh & Franklin's IBE

- Authority:
  - Public key: g^s
  - Private key: s (msk)
- User with ID_u:
  - Public key: $h(ID_u)$
  - Private key: $h(ID_u)^s$
- Session key:
  - Encryption key: $e(h(ID_u), g^s)^r = e(h(ID_u), g)^{sr}$; g^r
  - Decryption key: $e(h(ID_u)^s, g^r) = e(h(ID_u), g)^{sr}$



Understanding BF-IBE

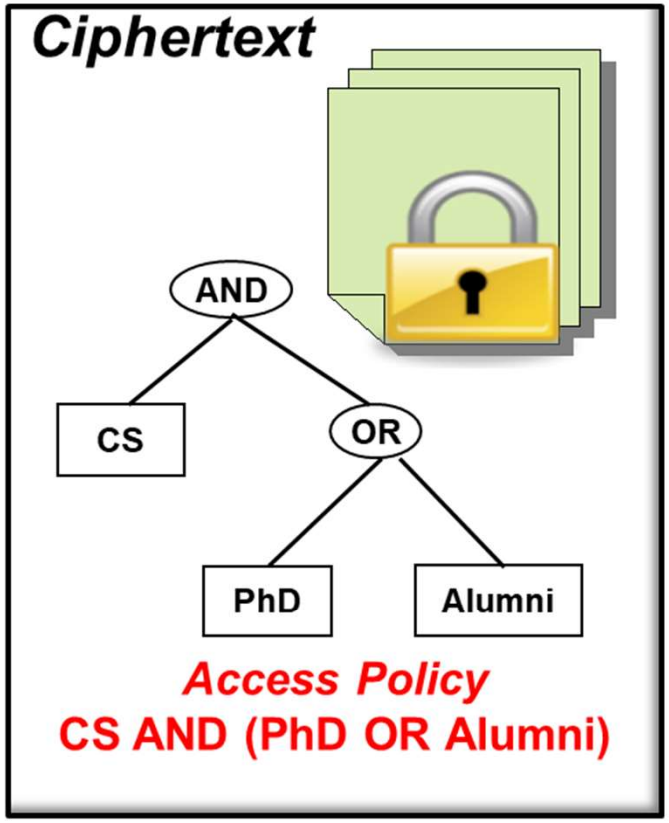
- How to understand this? (From Joux's point of view)
- Let t be the discrete log of $h_1(\text{"Bob"})$ base g
- We don't know what it is, but it is well defined
- Now the situation is like 3-party Diffie-Hellman
  - Alice has public g^r, private r
  - PKG has public g^s, private s
  - Bob has public g^t, unknown (!) t
- $e(h_1(\text{"Bob"}), g)^{rs} = e(g^t, g)^{rs} = \hat{g}^{rst}$ is like session key for encryption

基于属性的加密

- 发展源头：2005年，从身份基加密发展而来
  - ✓ 以生物信息（例如指纹）作为用户身份时，需要支持模糊匹配：加密的时候使用的“身份”是一组数据的集合，解密的时候使用的“身份”也是一组数据的集合，两组数据并不能完全一致，而是相同的数据达到一定门限值的时候，就认为是同一个“身份”。
- 进一步发展：
  - ✓ 每个用户以一组属性来描述，并产生相应的秘钥
  - ✓ 加密的时候加密者决定一个基于用户的属性的访问策略
  - ✓ 当用户的秘钥的属性集合能够满足密文的访问策略时，解密成功

密文策略属性基加密
Ciphertext-Policy ABE (CP-ABE)

密文的访问策略由加密者决定，用户的秘钥由其属性集合决定。
当且仅当用户的属性集合满足密文的访问策略时，解密成功。



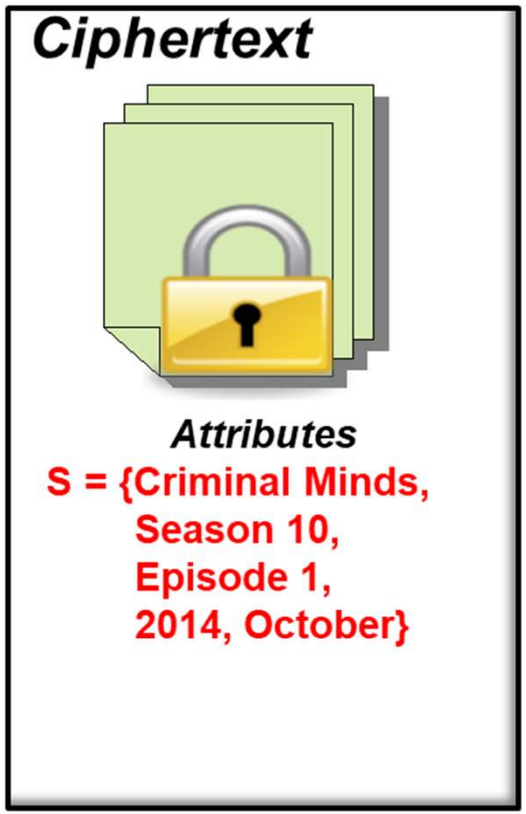
S_A={CS, PhD, Alice}

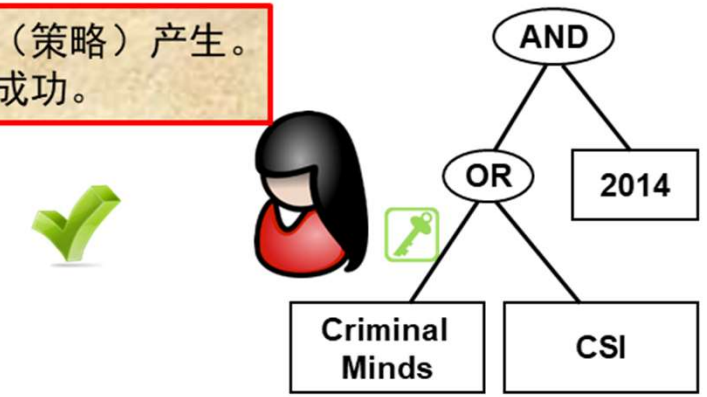
S_B={CS, Alumni, Bob}

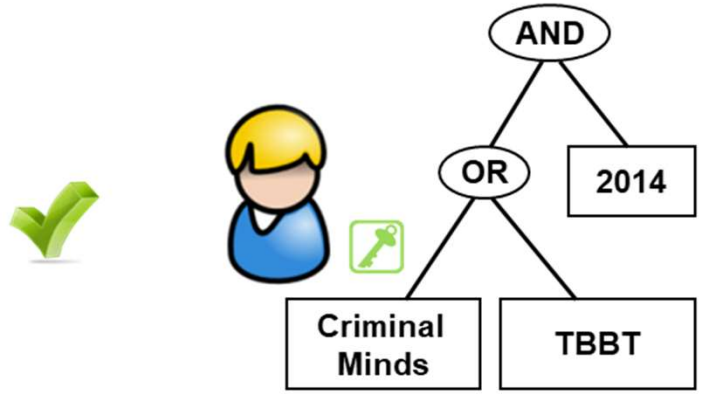
S_C={CS, UnderGrad, Carl}

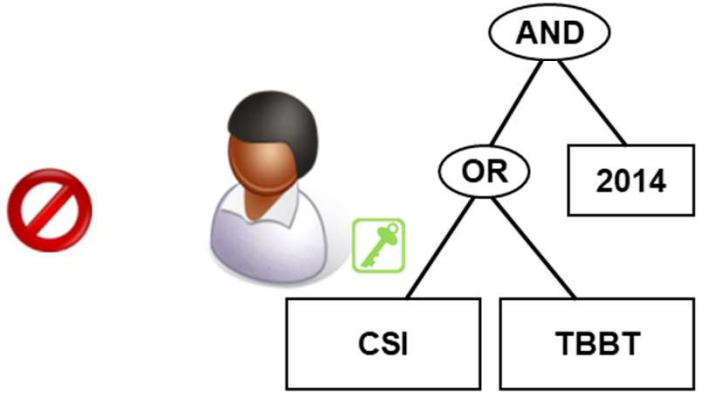
秘钥策略属性基加密 Key-Policy ABE (KP-ABE)

密文由其描述性属性加密，用户的秘钥根据其访问权限（策略）产生。
当且仅当密文的属性集合满足用户的访问策略时，解密成功。







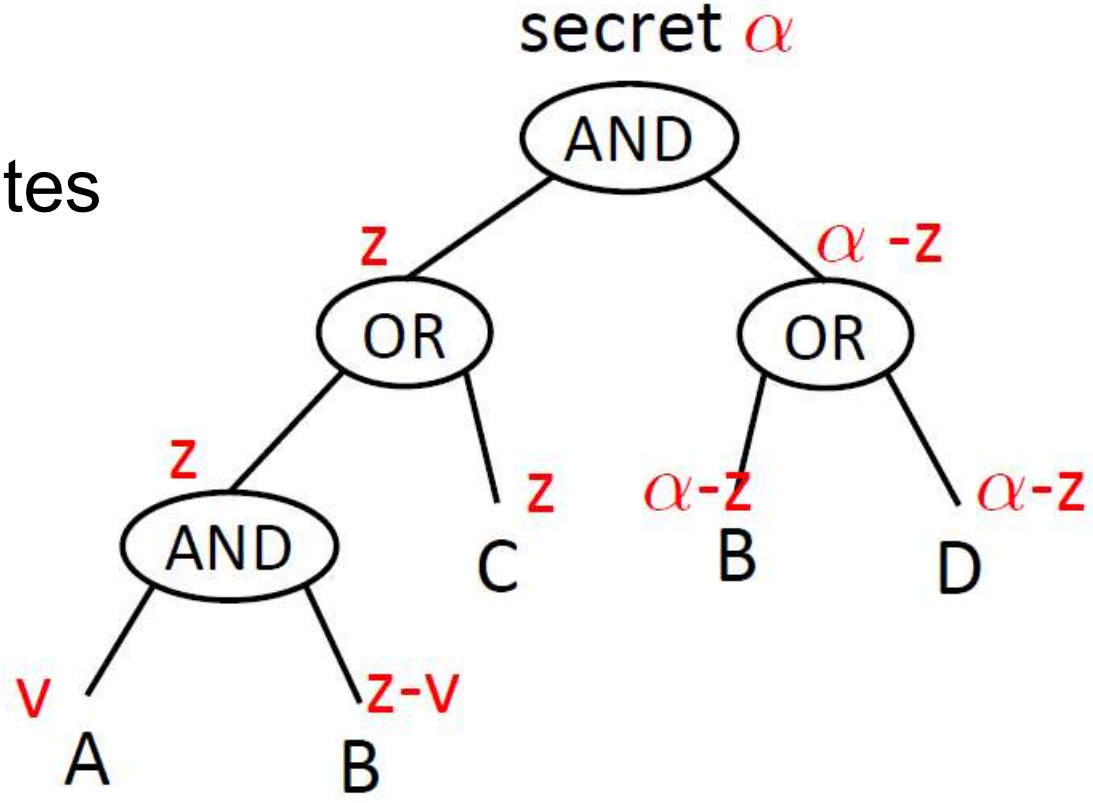


Access policy

- Access policy is represented as a tree structure where the interior nodes consist of threshold gates and the leaves consist of attributes.

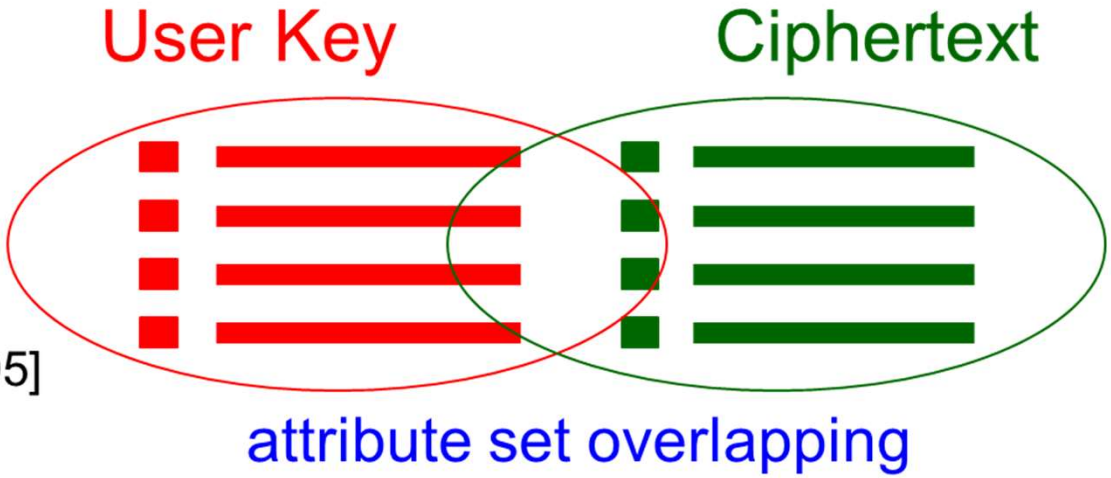
- AND, OR gates are special threshold gates
  - AND = n-out-of-n
  - OR = 1-out-of-n

- Linear secret sharing schemes

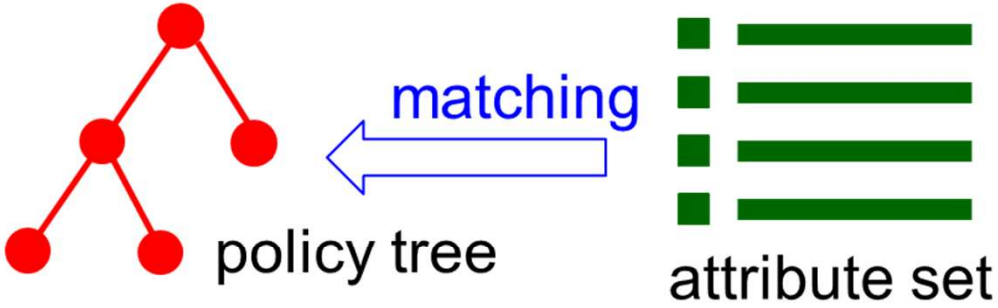


Classification of ABE

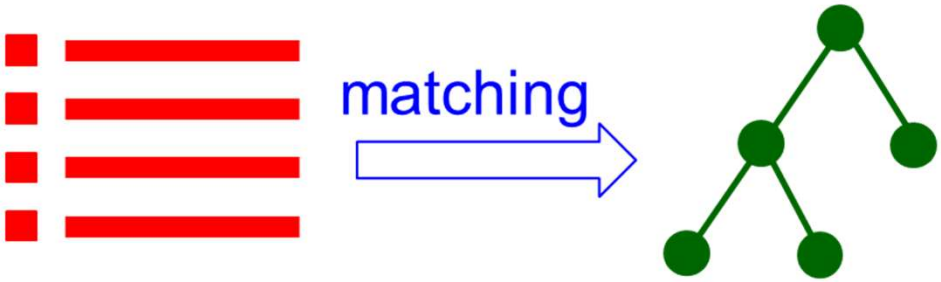
(1) Threshold-ABE [Sahai-Waters'05]



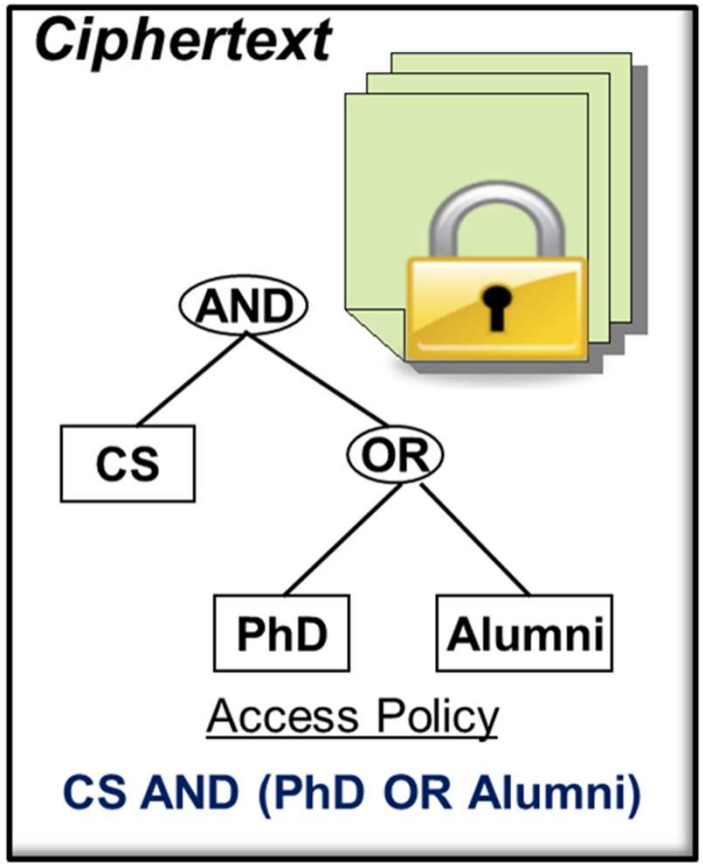
(2) KP-ABE [Goyal'06]

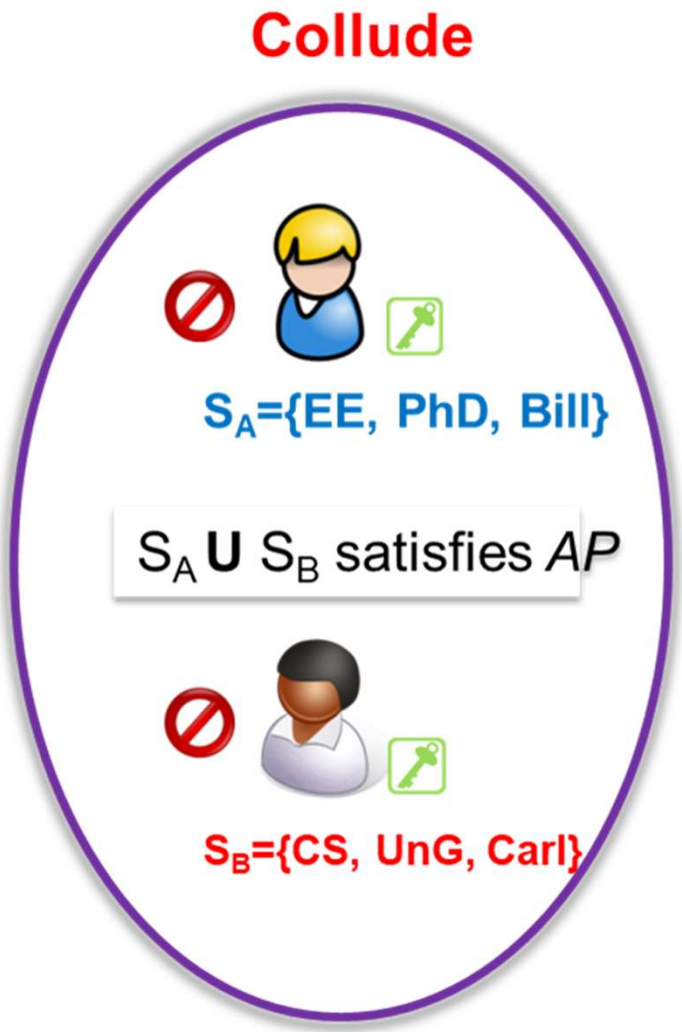


(3) CP-ABE [Bethencourt'07]



基于属性的加密：抗串谋攻击





基于属性的加密

- 特点：
  - ✓ 是一种一对多的公钥加密机制
    - ➢ 此前只有广播加密是一对多的加密机制
  - ✓ 加密的时候加密者不需要知道接收者的具体身份，只需要使用描述性属性制定访问策略
    - ➢ 第一个具备这种功能的加密机制
  - ✓ 能够实现对加密数据的细粒度访问控制
    - ➢ 第一个具有这种功能的加密机制
    - ➢ 被认为是实现安全访问控制的最有前途的机制之一

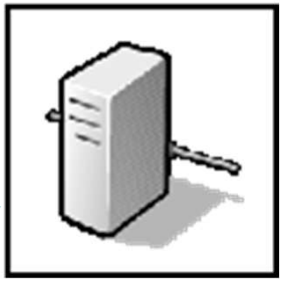
Part 7: outline

- S&P Issues for Cloud Computing
- Crypto 2.0
  - Attribute-based Encryption
  - Anonymous Credential
  - Homomorphic Encryption
- PETs
  - PIR/ORAM
  - Differential Privacy
  - Trusted Hardware-SGX

Credentials

Organization

Alice



Service

Org says
Name Alice
Address
Birthdate
Birthplace
Citizenship
…

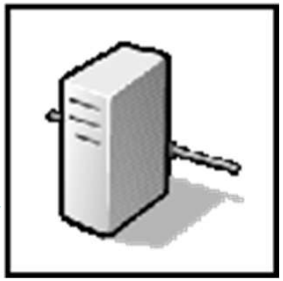
Org says
Name Alice
Address
Birthdate
Birthplace
Citizenship
…

Reveals a lot of info on Alice!

Anonymous credentials

Organization

Alice



Service

"I have a cred from Org saying
WA resident
Age >21"

Cred from Org
Name Alice
Address
Birthdate
Birthplace
Citizenship
…

Reveals only what Alice chooses to reveal

Need not reveal her name

(Need Accountability)

Anonymous credentials

Organization

Alice

Service

"I have a cred from Org saying
WA resident

Cred from Org
Name Alice
Address
Birthdate
Birthplace
Citizenship
…

- Cannot
  - Identify Alice
    (if her name is not provided)
  - Learn anything beyond the info she gives
    (and what can be inferred)
  - Distinguish two users with the same attributes
  - Link multiple uses of the same credentials

Limitations of PKEs

- Single, known receiver
  - Unknown recipient?
  - Many recipient?
  - More may join system later?

  ⇨ Attribute-based encryption

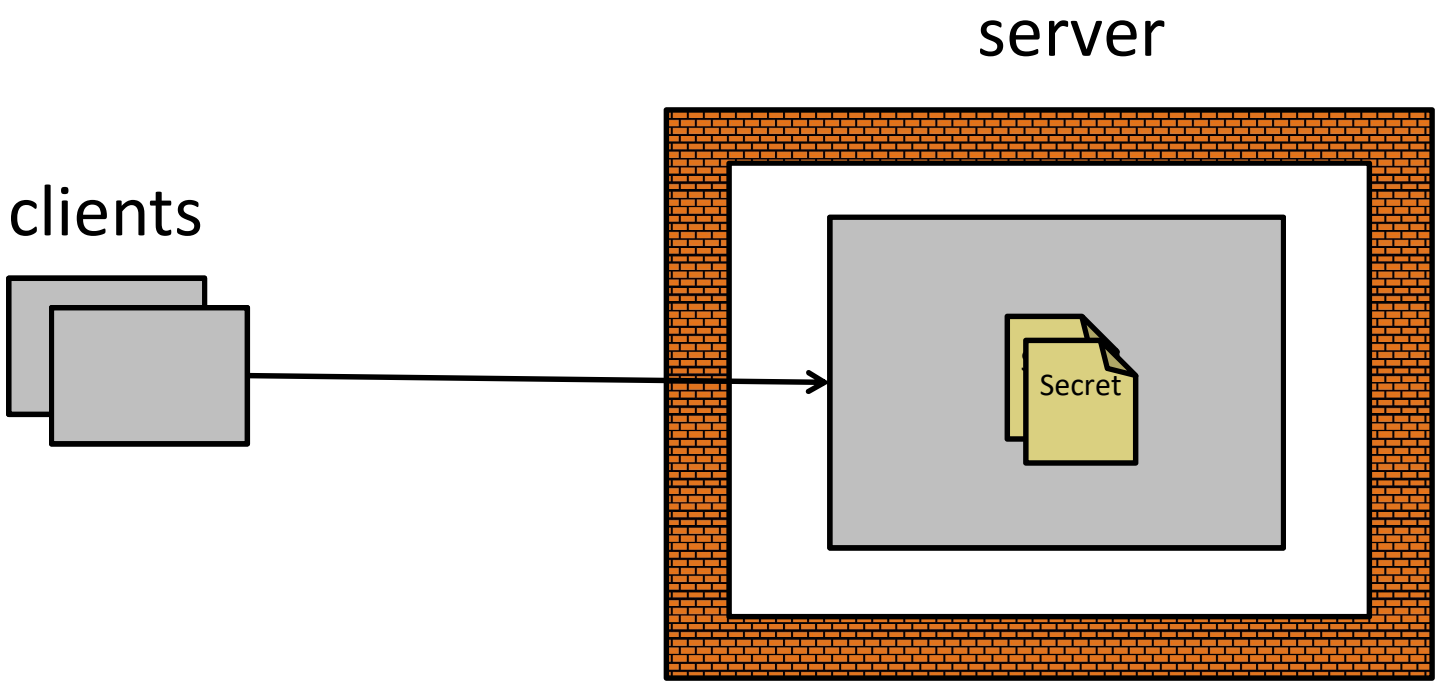
- All-or-nothing decryption
  - Should each allowed recipient see everything?

  ⇨ Functional encryption

Part 7: outline

- S&P Issues for Cloud Computing
- Crypto 2.0
  - Attribute-based Encryption
  - Anonymous Credential
  - Homomorphic Encryption
- PETs
  - PIR/ORAM
  - Differential Privacy
  - Trusted Hardware-SGX

Current system strategy



- Prevent attackers from breaking into servers

Lots of existing work

- Checks at the operating-system level
- Language-based enforcement of a security policy
- Static or dynamic analysis of application code
- Checks at the network level
- Trusted hardware
- …

Data still leaks even with these mechanisms

Because

attackers eventually break in!

Attacker examples

Attacker:

Reason they succeed:

hackers 

software is complex

cloud employees 

insiders: legitimate server access!

increasingly many companies store data on external clouds

government 

e.g., physical access

accessed private data according to 

Privacy in the cloud: encryption

- Encryption is the key element for privacy in the cloud
- Different encryption schemes
  - Symmetric
  - Asymmetric (with PKI)
- Different encryption implementation
  - At cloud provider
  - At third party
  - At the access device

Computing on encrypted data

- Privacy by design can be reached for cloud storage, but what if you want to process data in the cloud ?

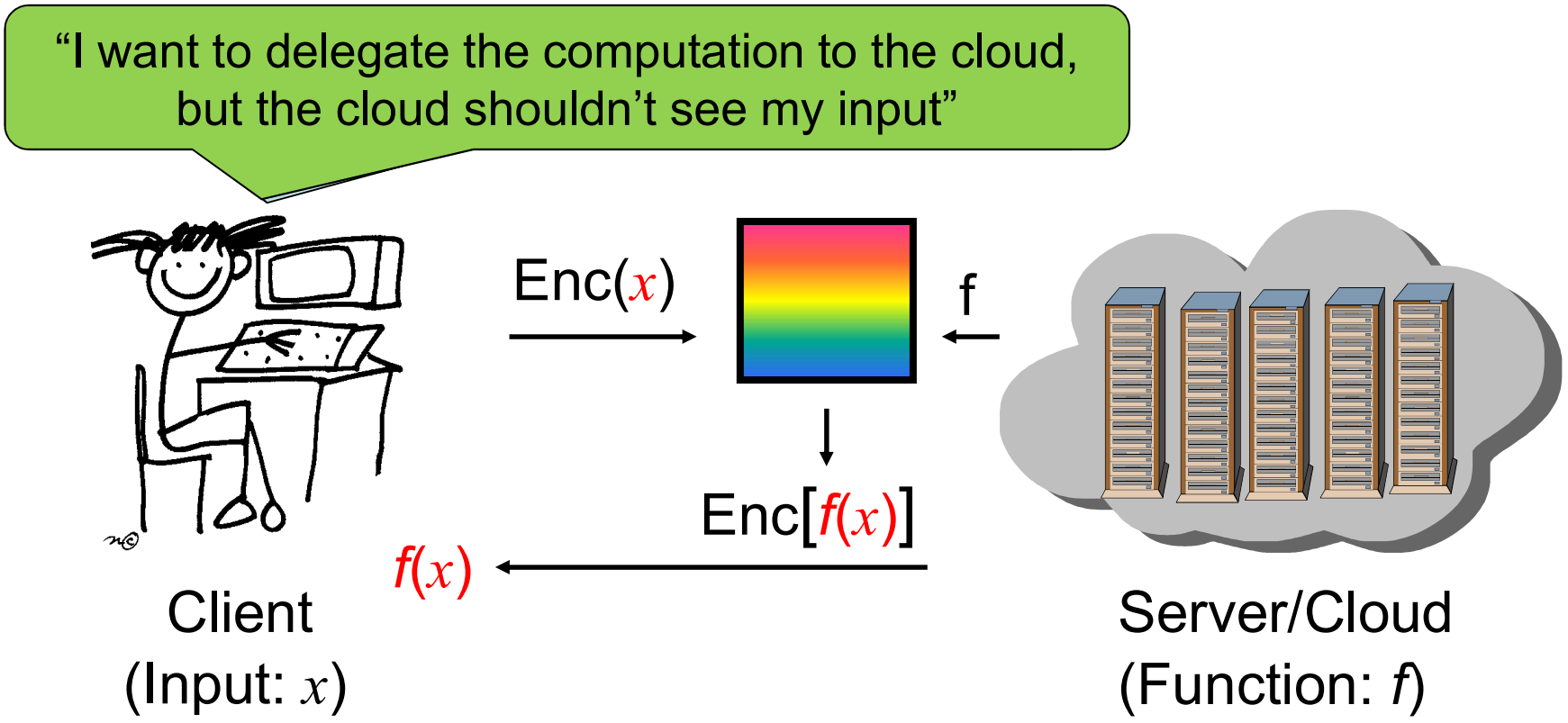
- (Fully) Homomorphic encryption
- Searchable encryption
- Privacy-preserving techniques

Computing on encrypted data

- Wouldn't it be nice to be able to…
  - Encrypt my data before sending to the cloud
  - While still allowing the cloud to search/sort/edit/… this data on my behalf
  - Keeping the data in the cloud in encrypted form
    - Without needing to ship it back and forth to be decrypted

I want to delegate processing of my data, without giving away access to it.

Outsourcing computation privately



Homomorphic encryption

- Computing (f) on encrypted data (Enc)

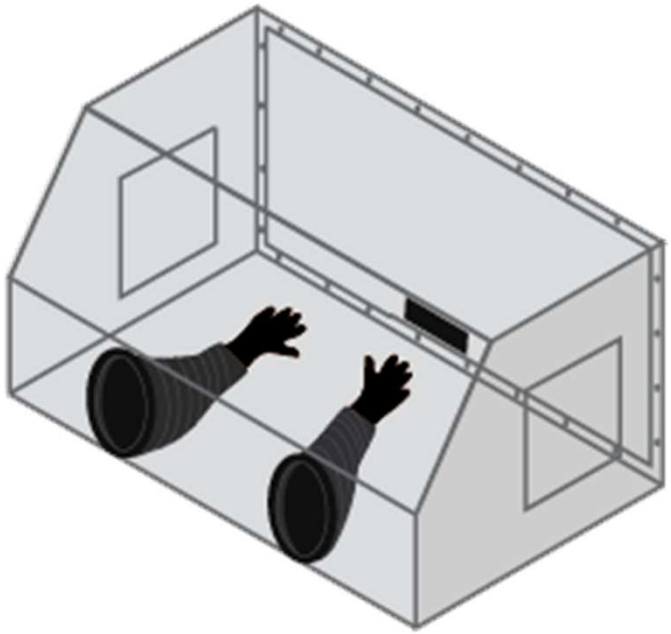
$$f(Enc[x]) = Enc[f(x)]$$

Alice's jewelry store

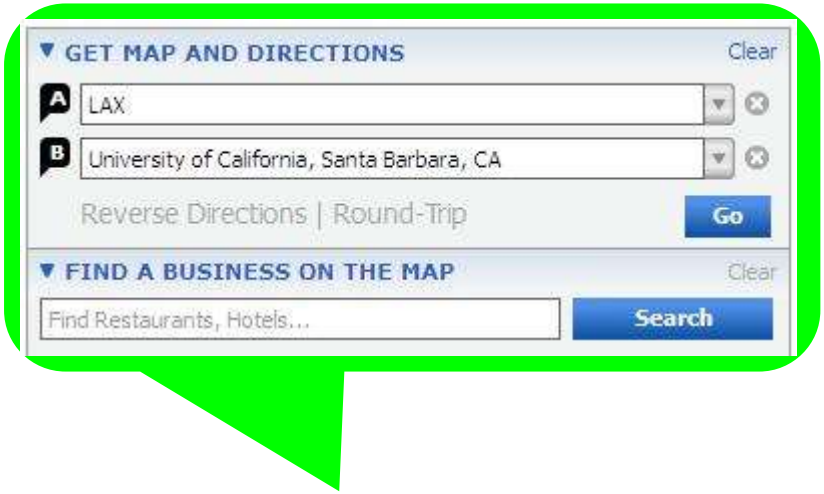
- Alice's workers need to assemble raw materials into jewelry
- But Alice is worried about theft
  - How can the workers process the raw materials without having access to them?

Alice's jewelry store

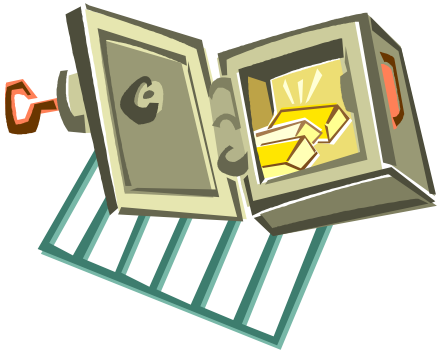
- Alice puts materials in locked glovebox
  - For which only she has the key
- Workers assemble jewelry in the glovebox
- Alice unlocks box to get "results"

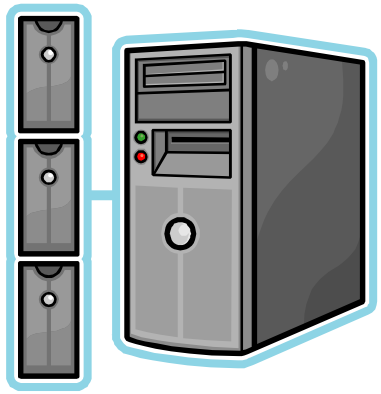


Outsourcing computation privately

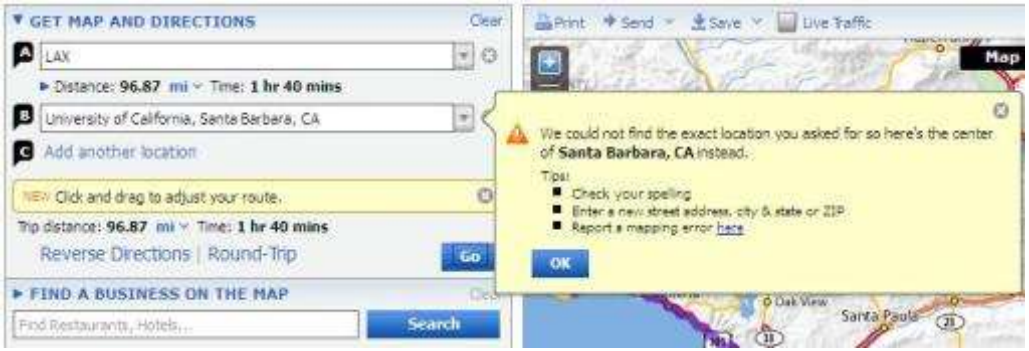


$skj#hS28ksytA@ …

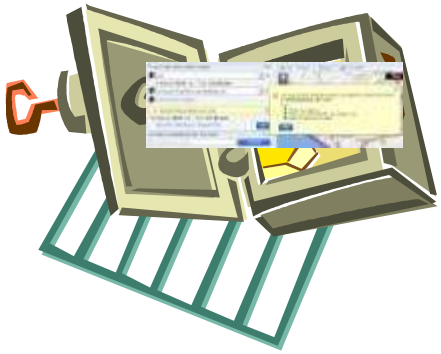


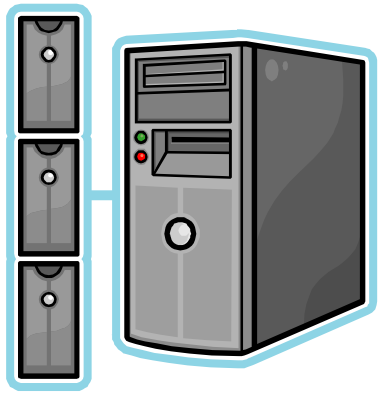


Outsourcing computation privately




```
$kjh9*mslt@na0
&maXxjq02bflx
m^00a2nm5,A4.
pE.abxp3m58bsa
(3saM%w,snanba
nq~mD=3akm2,A
Z,ltnhde83|3mz{n
dewiunb4]gnbTa*
kjew^bwJ^mdns0
```

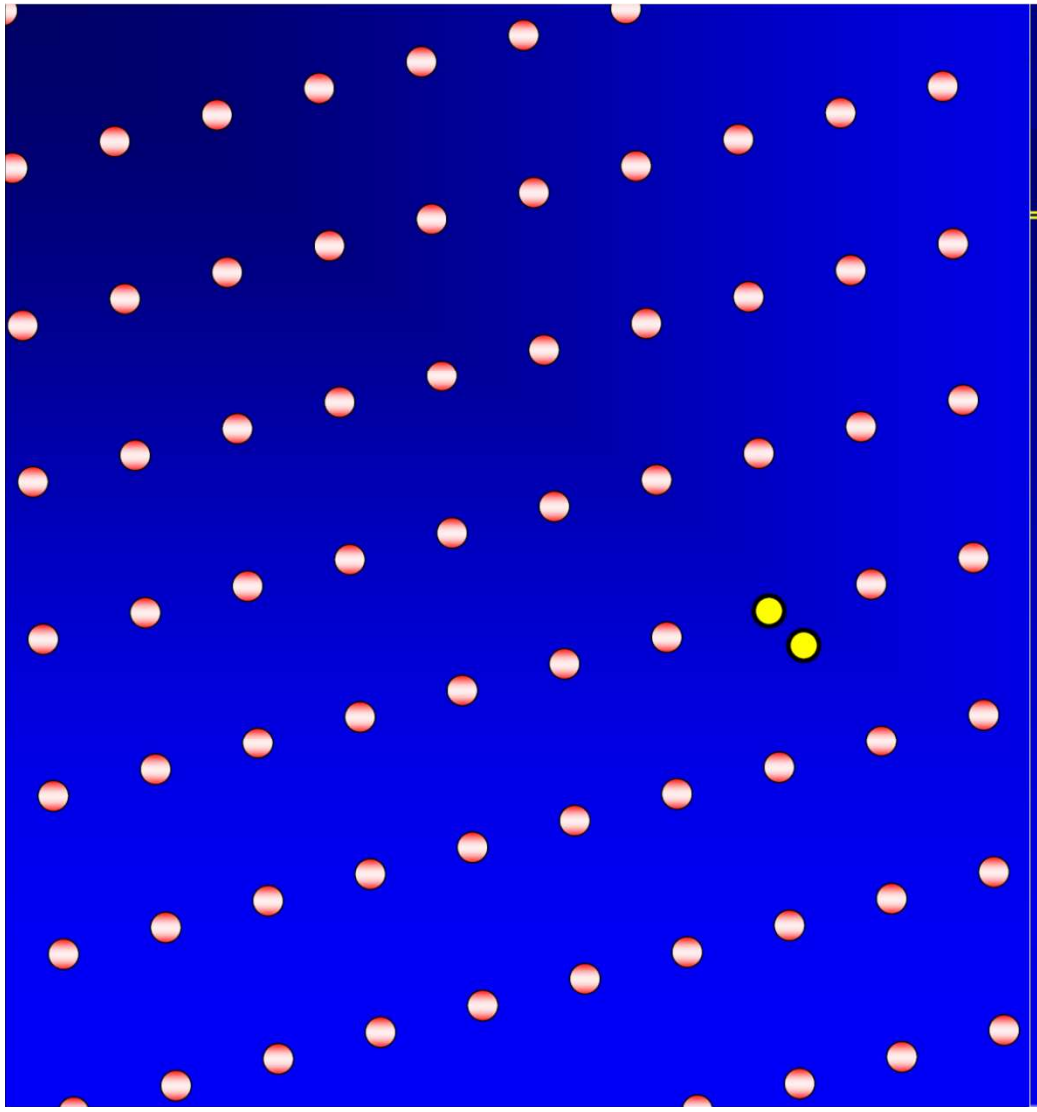




Homomorphic encryption

- (Fully) Homomorphic Encryption
  - Homomorphic encryption: Form of encryption which allows operations on ciphertext without decrypting the ciphertext before
  - Big breakthrough: First Construction of fully homomorphic encryption (Gentry 09)
  - For practical implementation currently far too inefficient
  - High research activity
- Searchable Encryption
  - Symmetric searchable encryption
  - Asymmetric searchable encryption

Lattice-based crypto



Lattice-based crypto

- ☺ Provably secure
- ☺ Security based on a worst-case problem
- ☺ Based on hardness of lattice problems
- ☺ (Still) Not broken by quantum algorithms
- ☺ Very simple computations
- ☺ Can do more things

Problem of FHE

Computing on encrypted data in cryptography

[Rivest-Adleman-Dertouzos'78]

Fully homomorphic encryption (**FHE**) [Gentry'09]

prohibitively slow, e.g., slowdown X 10^9

We need: **practical systems**

real-world performance + large class of real applications + meaningful security

FHE performance

	Dimension	KeyGen	PK size	AND
Small	2048 800,000-bit integers	40 sec	70 MByte	31 sec
Medium	8192 3,200,000-bit integers	8 min	285 MByte	3 min
Large	32768 13,000,000-bit integers	2 hours	2.3 GByte	30 min

- Butler Lampson: "I don't think we'll see anyone using Gentry's solution in our lifetimes…"

  – Forbes, Dec-19, 2011

Homomorphic encryption

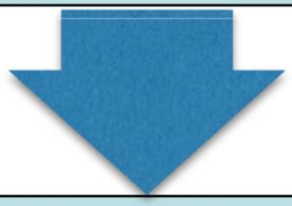
- Computing (f) on encrypted data (Enc)

$$f(Enc[x]) = Enc[f(x)]$$

Program obfuscation

- Make programs “unintelligible” while maintaining their functionality

```
for (i=0; i < M.length; i++) {
// Adjust position of clock hands
   var ML=(ns)?document.layers['nsMinutes'+i]:ieMinutes[i].style;
   ML.top=y[i]+HandY+(i*HandHeight)*Math.sin(min)+scrll;
   ML.left=x[i]+HandX+(i*HandWidth)*Math.cos(min);
}
```



```
for(O79=0;O79<l6x.length;O79++){var O63=(l70)?document.layers
["nsM\151\156u\164\145s"+O79]:ieMinutes[O79].style;
O63.top=l61[O79]+O76+(O79*O75)*Math.sin(O51)+l73;
O63.left=l75[O79]+l77+(O79*l76)*Math.cos(O51);}
```

Program obfuscation

- Rename Obfuscation

Original Source Code Before Rename Obfuscation	Reverse-Engineered Source Code After Rename Obfuscation
private void CalculatePayroll (SpecialList employee-Group) { while (employeeGroup.HasMore()) { employee = employeeGroup.GetNext(true) ; employee.UpdateSalary() ; Distribute Check(employee) ; } }	private void a(a b) { while (b.a()) { a = b.a(true); a.a () ; a.(a) ; } }

- String Encryption

Original Source Code Before String Encryption	Reverse-Engineered Source Code After String Encryption
... MessageBox.show("Invalid Authentication - Try Again") ...	... MessageBox.show(a.b("¥∑¶§ƒжњж•₡")) ...

- Control Flow Obfuscation

Original Source Code Before Control Flow Obfuscation	Reverse-Engineered Source Code After Control Flow Obfuscation
public int CompareTo (Object o) { int n = occurrences - ((WordOccurrence)o) .occurrences ; if (n == 0) { n = String.Compare (word, ((WordOccurrence)o) .word) ; } return (n) ; }	private virtual int _a(Object A+0) { int local0 ; int local1 ; local 10 = this.a - (c) A_0.a; if (local10 != 0) goto i0 ; while (true) { return local1; } i1: local10 = System.String.Compare(this.b, (c) A_0.b) ; goto i0; }

- Instruction Pattern Transformation

Original Source Code To Swap Two Variables	Machine Level Instructions Created by the Complier	Machine Level Instructions After Transient Variable Caching	Original Source Code To Swap Two Variables

temp = a;
b = temp;
a = b;
	iload_1
	istore_3
	iload_2
	istore_1
	iload_3
	istore_2
		iload_1
		iload_2
		istore_1
		istore_2
			?
			There is no equivalent high level source

- Dummy Code Insertion

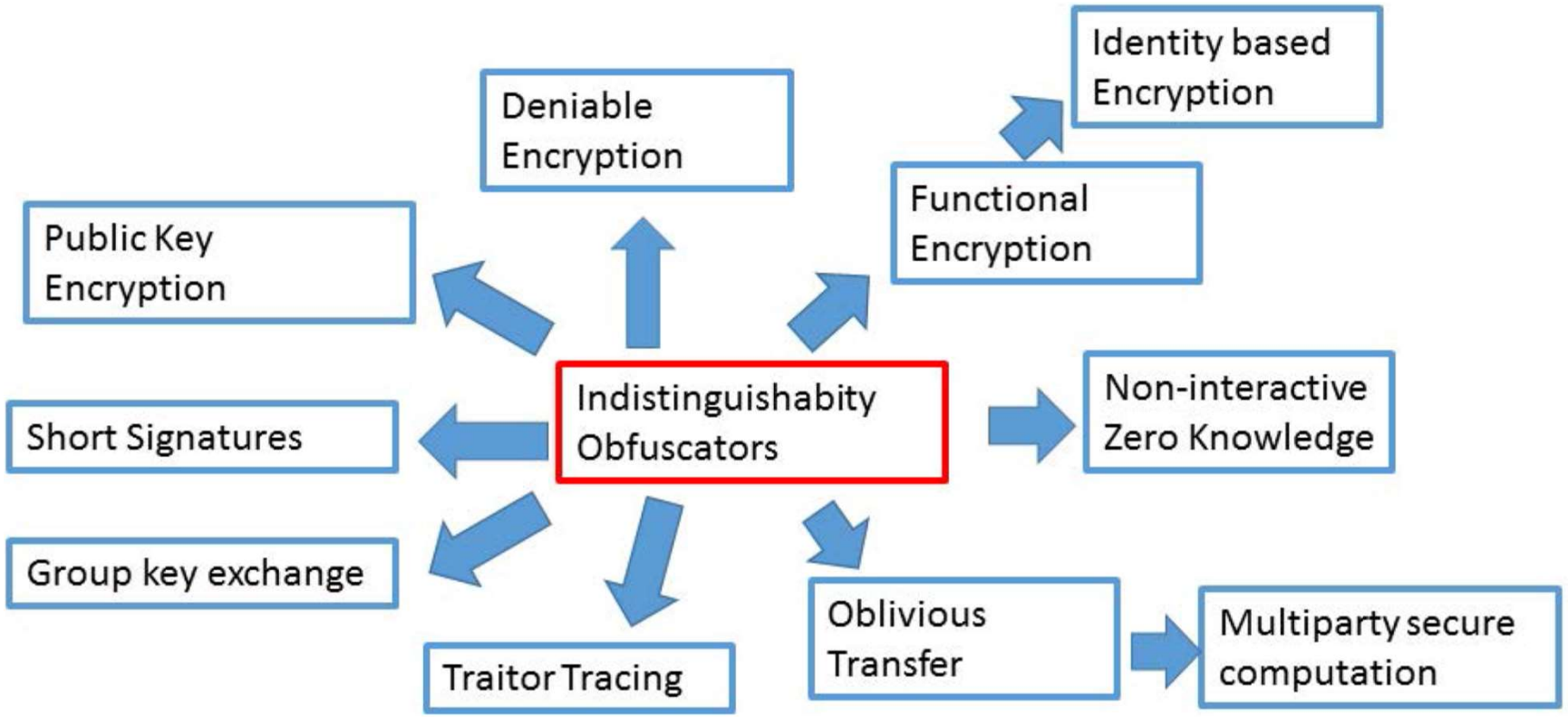
…

- Binary Linking/Merging

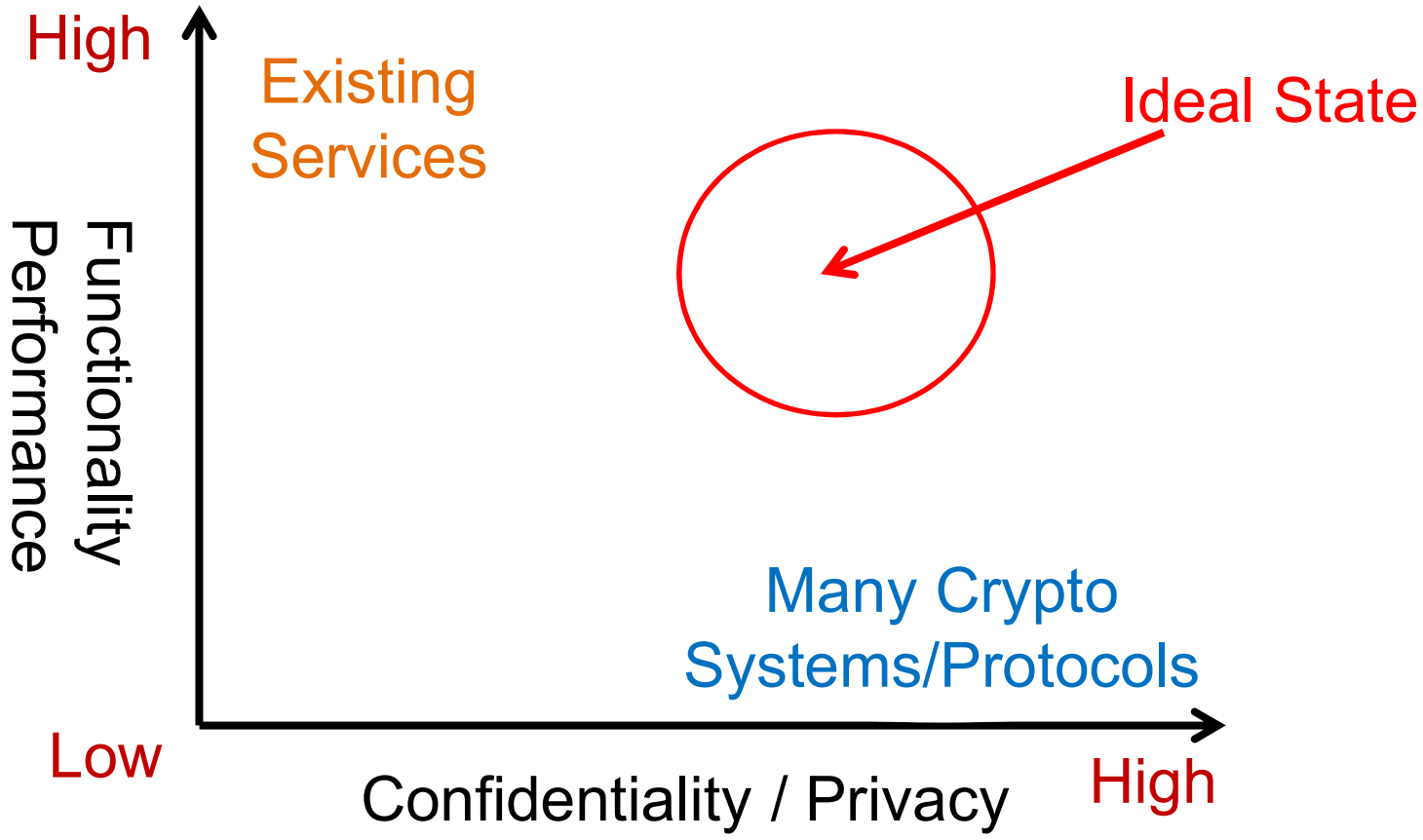
…

Program obfuscation

- Make programs “unintelligible” while maintaining their functionality

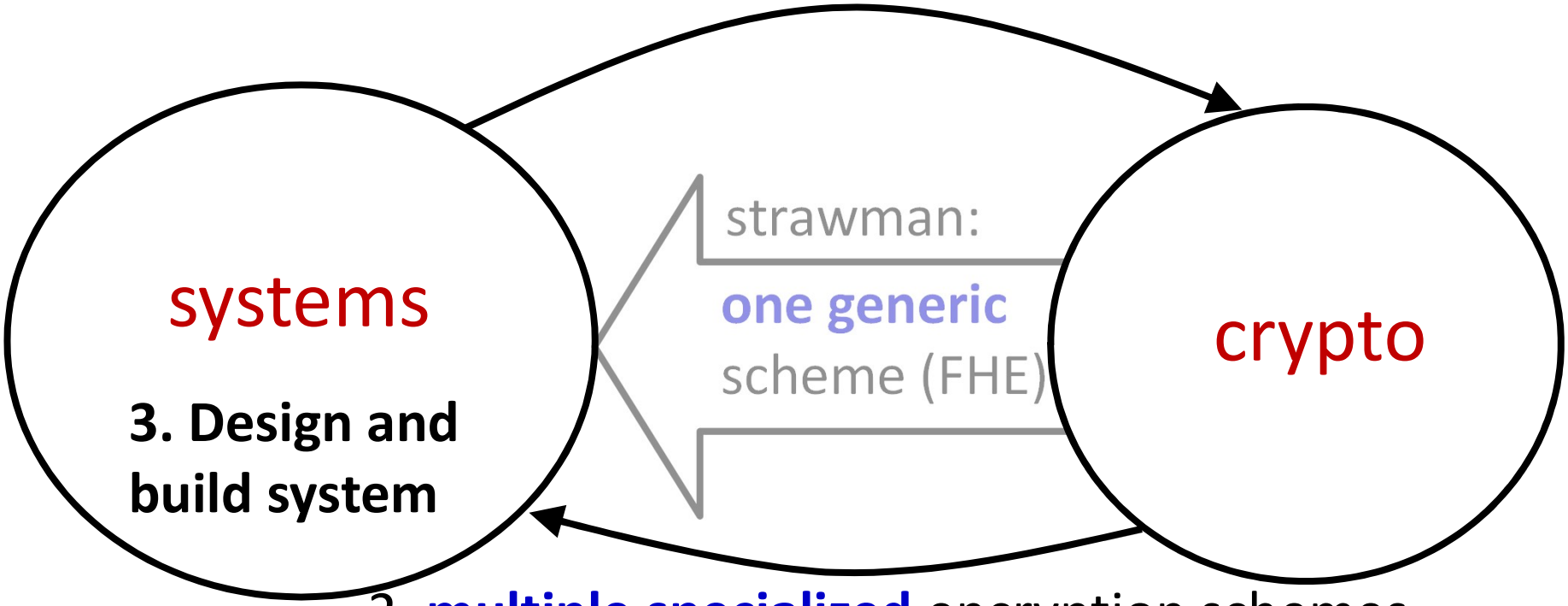


Challenges: conflicting goals



Combine systems and crypto

1. identify core operations needed

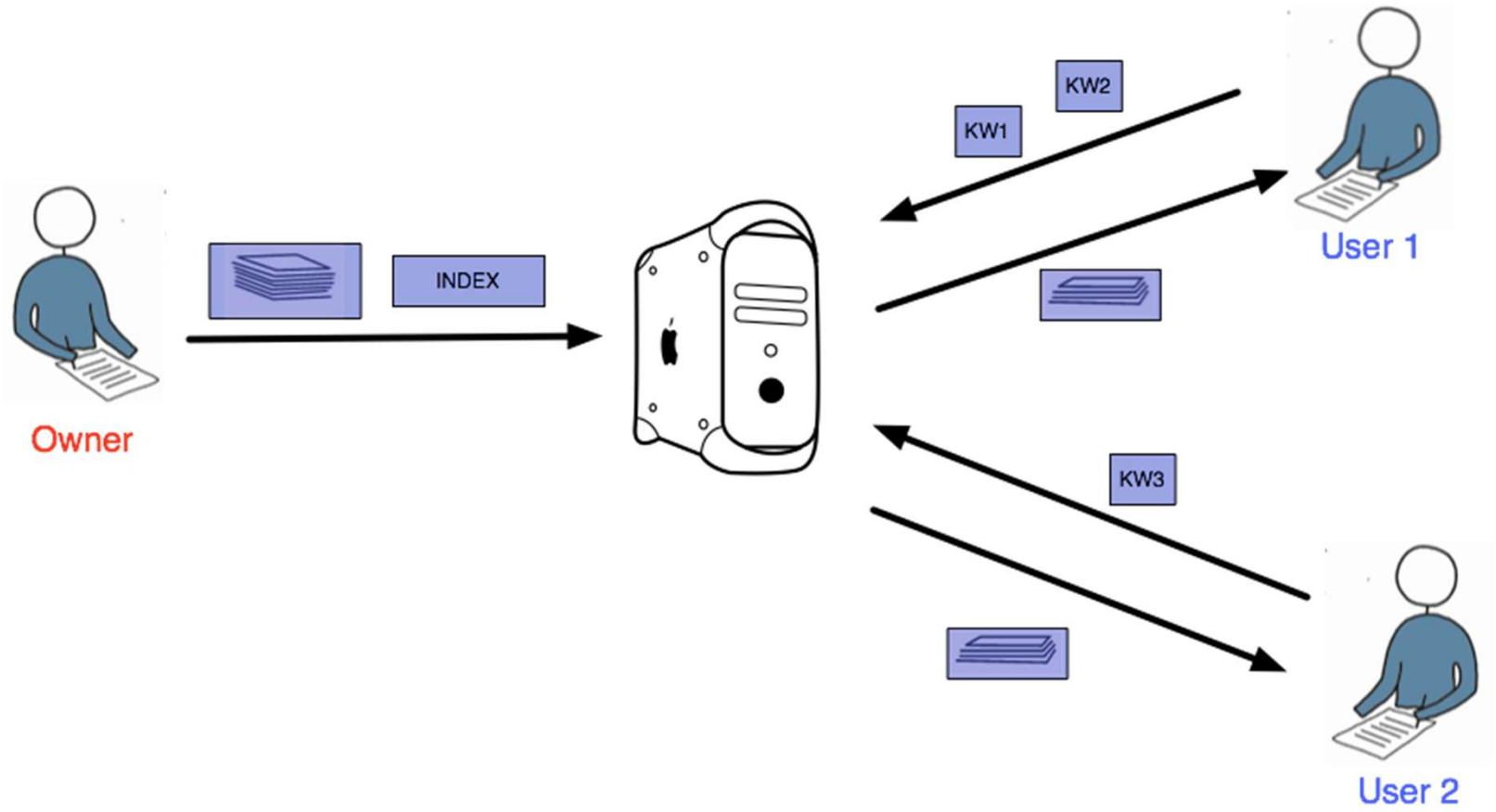


2. **multiple specialized** encryption schemes

New schemes:

- Searchable Encryption
- mOPE, adjJOIN for CryptDB

Searchable encryption

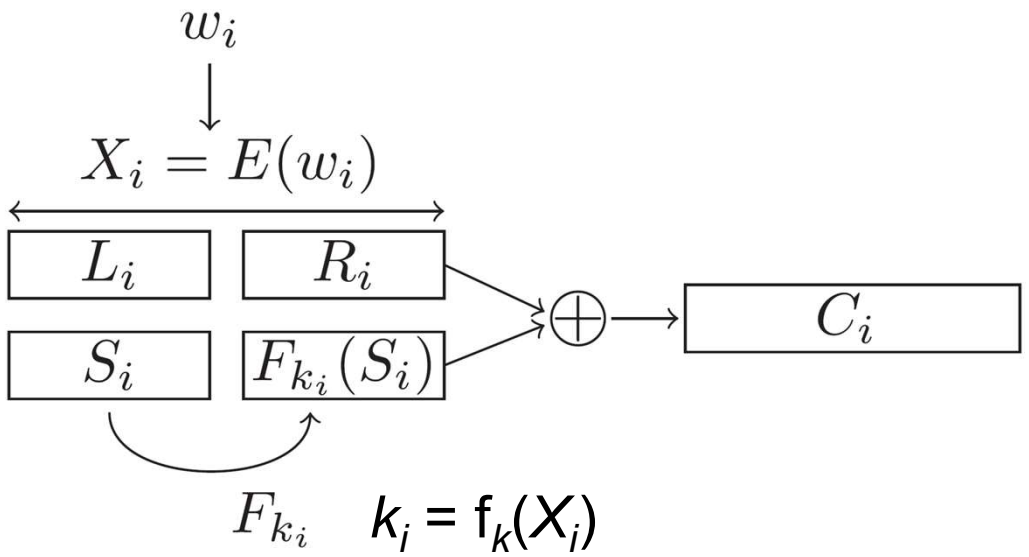


Searchable encryption

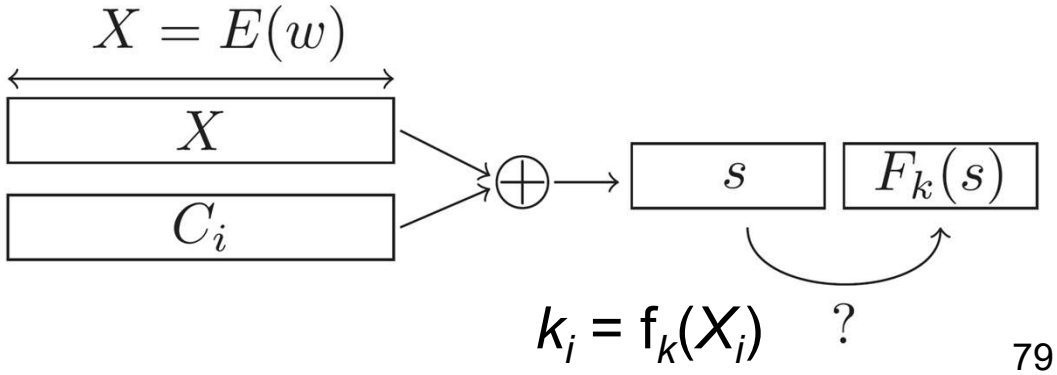
- Directly search on encrypted data without decryption on server side
- Encrypt word by word. For word W_i
  - Block_ciphertext $X_i = E_k(W_i)$, Word key $k_i = f_k(X_i)$, Pseudorandom sequence $T_i = <S_i, F_{ki}(S_i)>$
  - Searchable_ciphertext $C_i = X_i \oplus T_i$
- Search for a word W
  - Block_ciphertext $X = E_k(W)$, Word key $k_i = f_k(X)$
  - Check ciphertexts one by one to see if $C \oplus X = (X_i \oplus T_i) \oplus X$ is of the form $<s, F_{ki}(s)>$ for some random value s

Searchable encryption

- Directly search on encrypted data without decryption on server side
- Encrypt word W_i

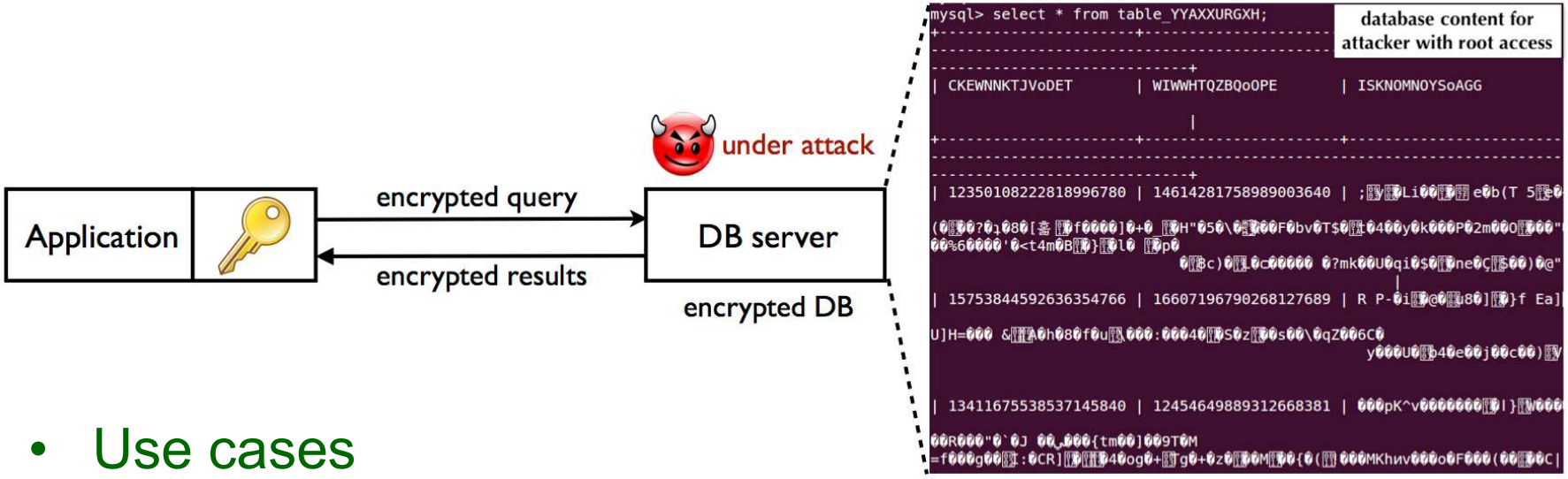


- Search for a word W



CryptDB @ MIT

- Executing SQL queries over encrypted data using a collection of efficient SQL-aware encryption schemes



- Use cases

SAP AG's system SEEED

Google's Encrypted BigQuery

Lincoln Laboratory

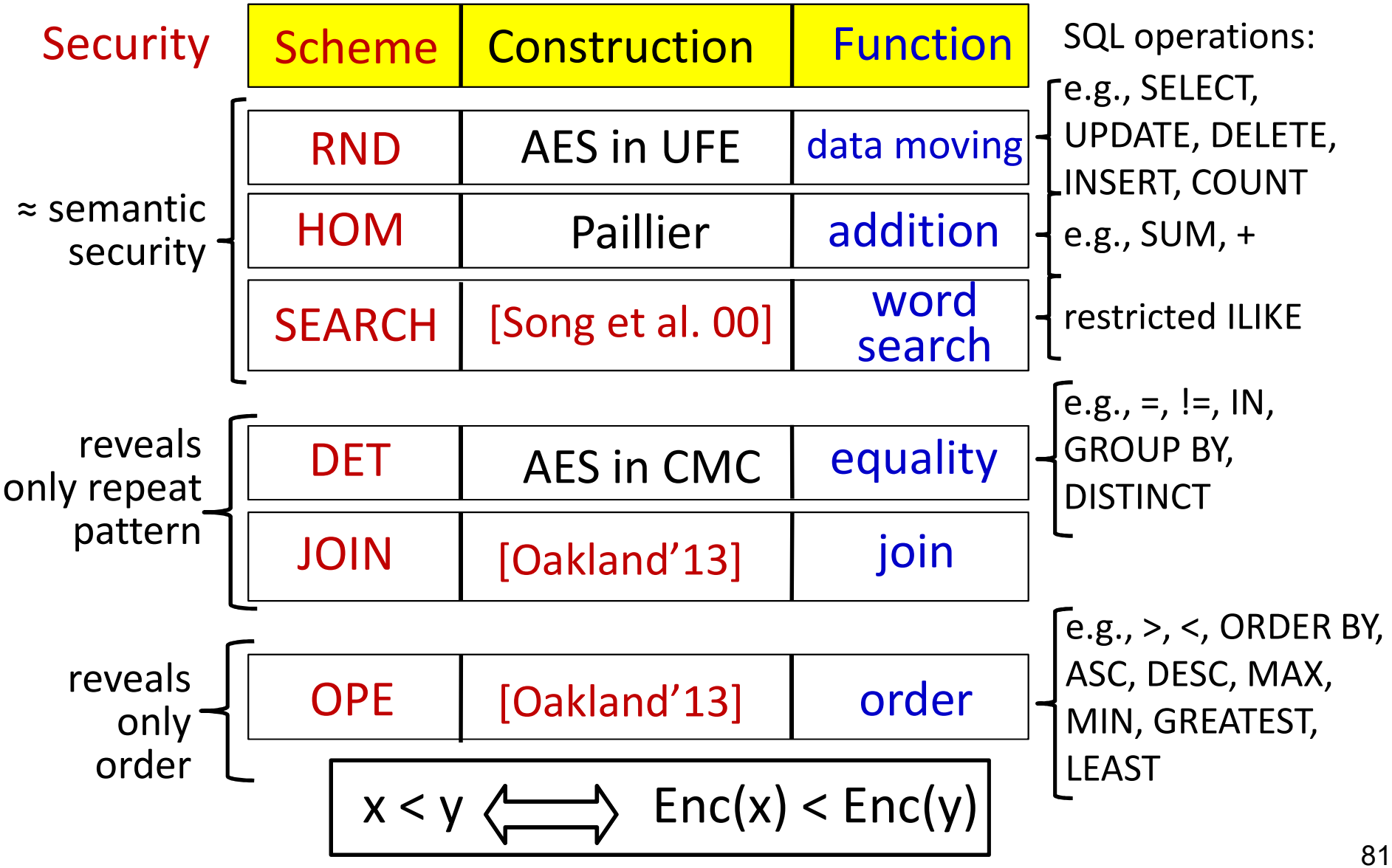
Microsoft's Always Encrypted SQL Server

Skyhigh Networks

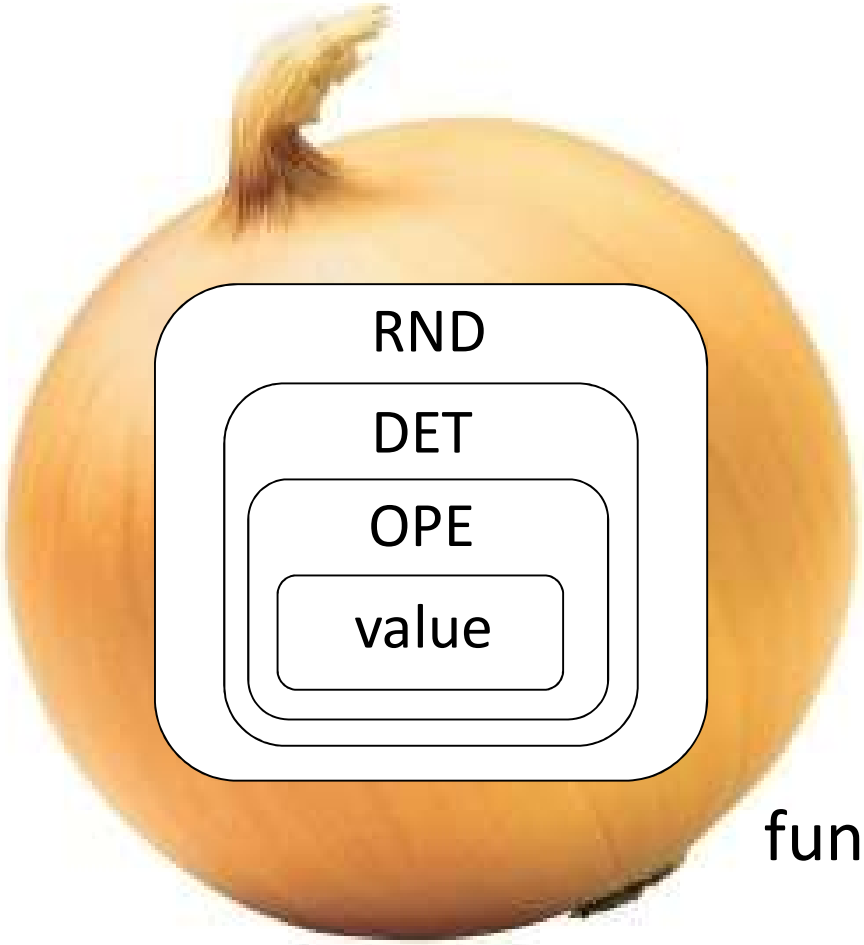
sql.mit.edu

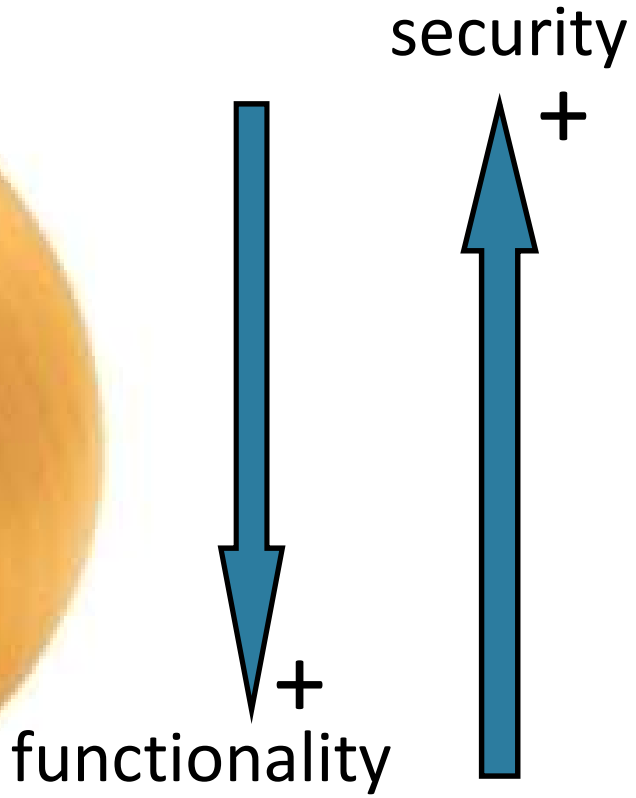
http://css.csail.mit.edu/cryptdb/

SQL-aware encryption



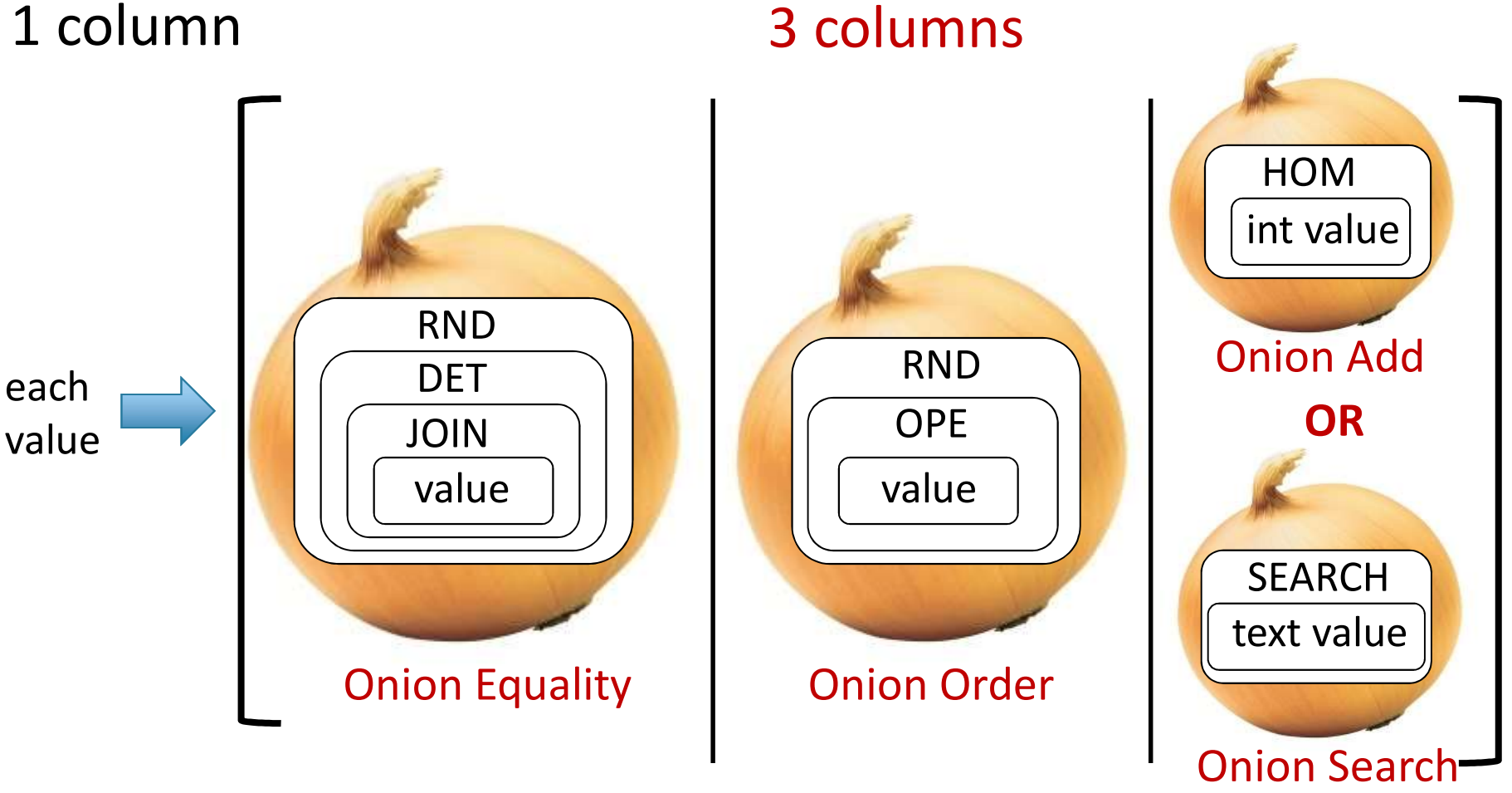
Onion of encryptions





Adjust encryption: strip off layer of the onion

Onion of encryptions



Same key for all items in a column for same onion layer

Part 7: outline

- S&P Issues for Cloud Computing
- Crypto 2.0
  - Attribute-based Encryption
  - Anonymous Credential
  - Homomorphic Encryption
- PETs
  - PIR/ORAM
  - Differential Privacy
  - Trusted Hardware-SGX

Metadata leakage problem

Metadata

Data about data

"Metadata was traditionally in the card catalogs of libraries"
-Wikipedia

Former National Security Agency (NSA) & Central Intelligence Agency (CIA) Director Michael Hayden (2014):

We kill people based on metadata

The National Security Agency's $1.5 billion data storage facility in Bluffdale, Utah, June 2013

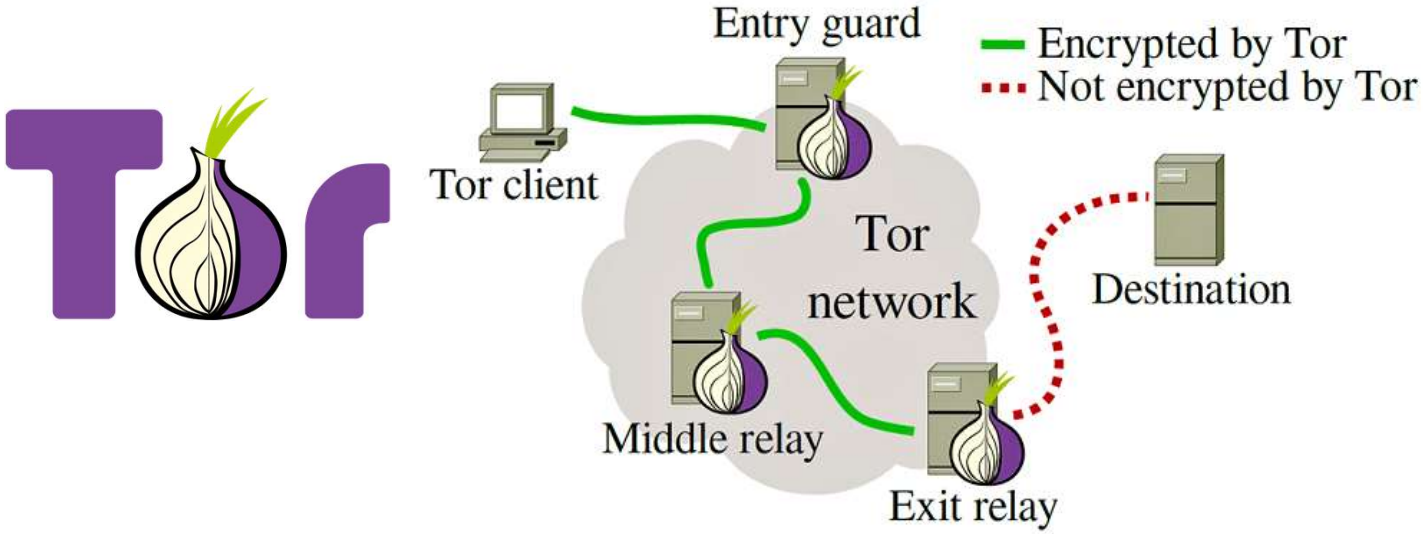
Protect communication metadata

- Who talks with whom, and what you browse is sensitive
  - Alice talks to Bob, Bob is a cancer doctor.
  - Alice Browses the NHS website, looking at pages on STDs.
- Extensive research shows a lot can be inferred from metadata:
  - Sender, receiver, length & time of communication, pattern.
  - Eg. mental condition, personality, language, emotions, political opinion.
  - Even if the content is encrypted!

Anonymous communications

- Anonymous communication systems hide such information:
  - Best known as Tor (the Onion Router)
  - How? Use a set of relays:

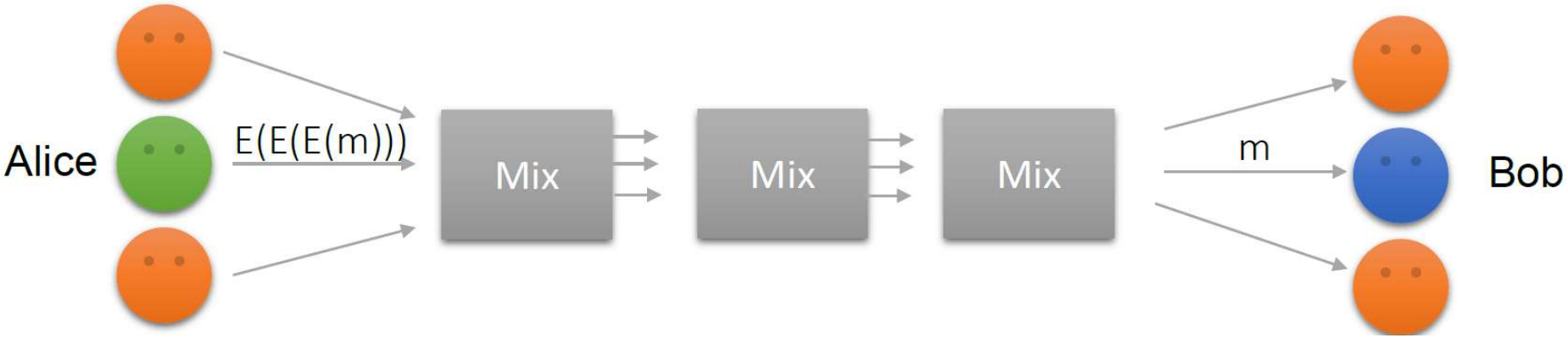
  - Illustrates: distribute trust, chose who to trust, crypto …



Proxies for anonym. comm.

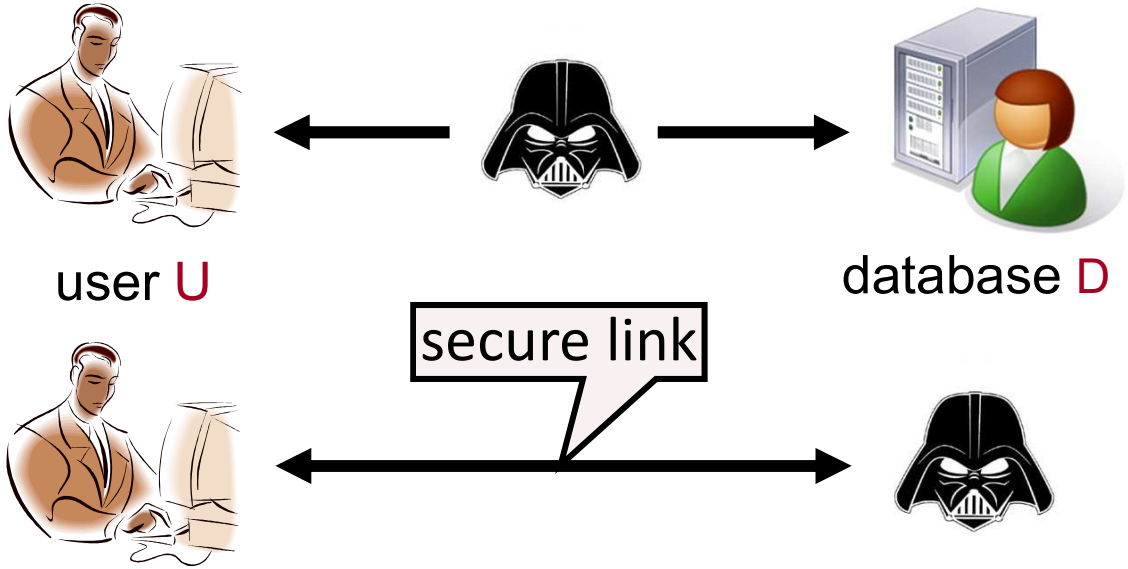
- Alice wants to hide the fact she is sending a message to Bob
  - The proxy decrypts the message.
  - The proxy batches many messages.
  - The proxy is the TCB
- Problem:
  - Low throughput.
  - Corrupt Proxy or Proxy hacked / coerced.

Proxies for anonym. comm.



- Solution: Use multiple cryptographic relays (mix)
  - Sender encrypts messages using multiple keys, through a set of mixes.
  - Each mix batches multiple messages.
  - TCB: Not a single mix, or client. No single place to coerce to trace everyone.
- From mix-networks to Onion Routing
  - OR: sender sends a stream of messages through the sequence of relays.
  - Problem: timing of traffic leads to correlation (c^2 attack)
  - Distributed TCB: adversary can compromise some circuits not all.

Threat model



- Key problem: which database record you access is sensitive!
  - Example: which book you are looking at the library?
  - Which friend you check if they are on-line?
  - What music you are listening?
  - Which minister you look up in your online address book?

Private Information Retrieval (PIR)

- Goal: allow user to query database while hiding the identity of the data-items she is after.
- Note: hides identity of data-items; not existence of interaction with the user.
- Motivation: patient databases; stock quotes; web access; many more....
- Paradox(?): imagine buying in a store without the seller knowing what you buy.

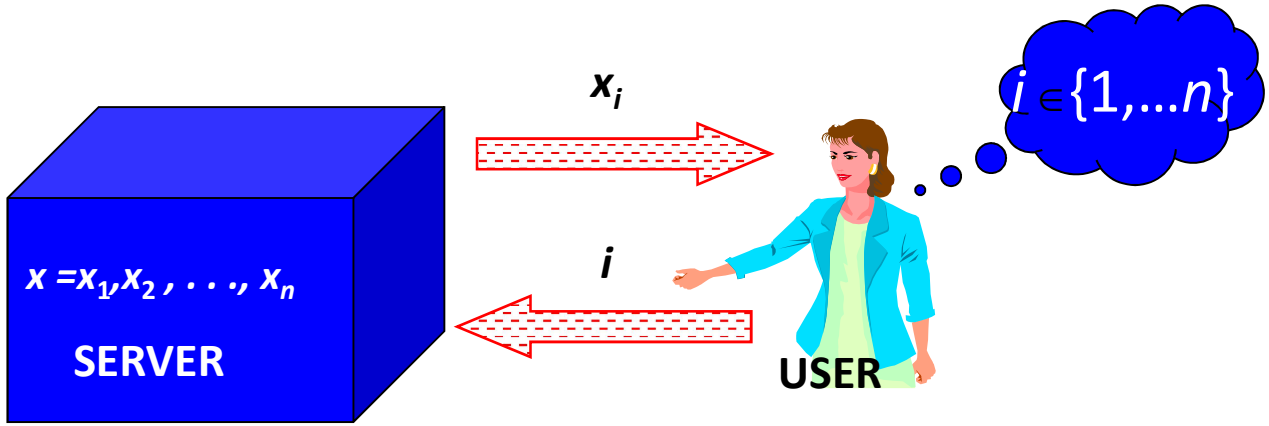
  (Encrypting requests is useful against third parties; not against owner of data.)

PIR model

- Server: holds n-bit string x

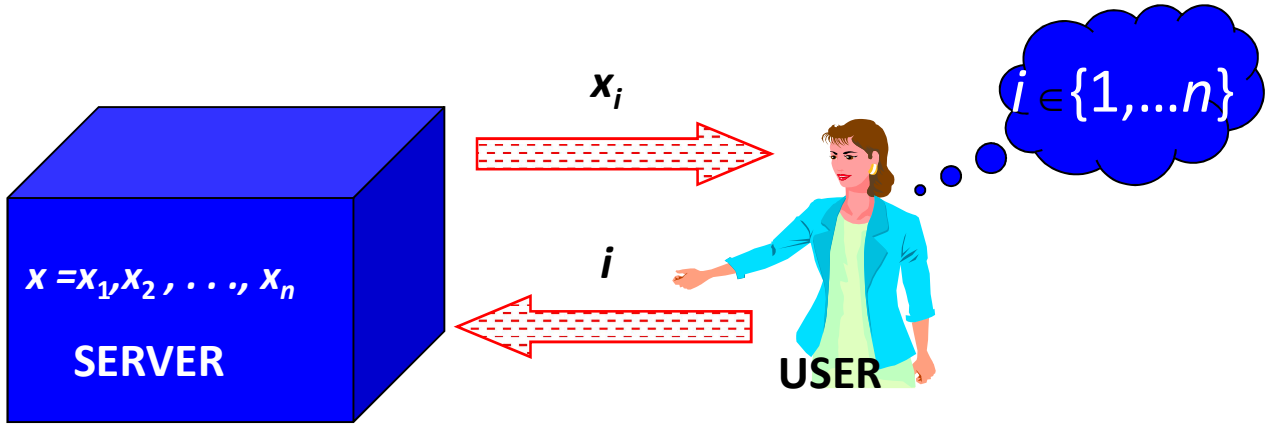
  n should be thought of as very large

- User: wishes
  - to retrieve x_i , and
  - to keep i private

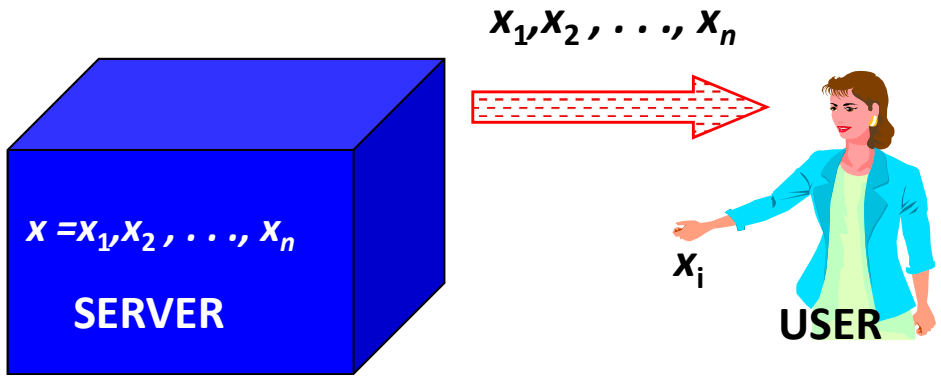


ORAM model

- PIR: access a public record without leaking which –even to the provider!
- ORAM: access your own private encrypted records, without divulging which (cheap) to cloud store.



Trivial private protocol



Server sends entire database ***x*** to User.

Information theoretic privacy.

Communication: ***n***

Not optimal !

Other solutions?

- User asks for additional random indices.
  Drawback: leaks information, reduces communication efficiency
- Employ general crypto protocols to compute x_i privately.
  Drawback: highly inefficient (polynomial in n).
- Anonymity (e.g., via Anonymizers).
  Note: different concern: hides identity of user; not the fact that x_i is retrieved.

Two approaches for PIR

- Information-Theoretic PIR [CGKS95,Amb97,...]
  - Replicate database among k servers
  - User queries all the servers

- Computational PIR [CG97,KO97,CMS99,...]
  - Computational privacy, based on cryptographic assumptions

Known comm. upper bounds

Multiple servers, information-theoretic PIR:

- 2 servers, comm. $n^{1/3}$ [CGKS95]
- k servers, comm. $n^{1/\Omega(k)}$ [CGKS95, Amb96,…,BIKR02]
- $\log n$ servers, comm. Poly($\log(n)$) [BF90, CGKS95]

Single server, computational PIR:

Comm. Poly($\log(n)$)

Under appropriate computational assumptions [KO97,CMS99]

Sub-linear with n

Computation PIR（CPIR）

- Only one server, no need to trust
- Based on cryptographic assumptions
- Downside: Server has to run over the whole database, otherwise leaks information
  - High computation load on the server

CPIR example

- Database has n records $\langle r_1,\ldots,r_i,\ldots,r_n\rangle$
- (Non-)private query via dot product with a unit vector

$$\langle 0,0,\ldots,1,\ldots,0\rangle \bullet \langle r_1,\ldots,r_i,\ldots,r_n\rangle = r_i$$

i^{th} position dot product single record

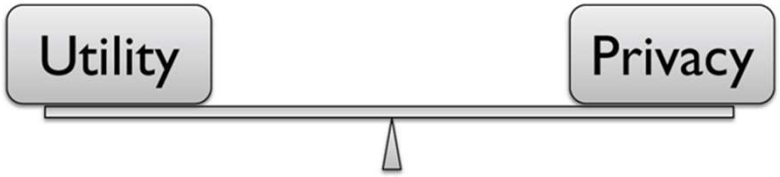
- Idea: encrypt query vector component-wise w/ some IND-CPA secure, XOR-homomorphic encryption scheme

Part 7: outline

- S&P Issues for Cloud Computing
- Crypto 2.0
  - Attribute-based Encryption
  - Anonymous Credential
  - Homomorphic Encryption
- PETs
  - PIR/ORAM
  - Differential Privacy
  - Trusted Hardware-SGX

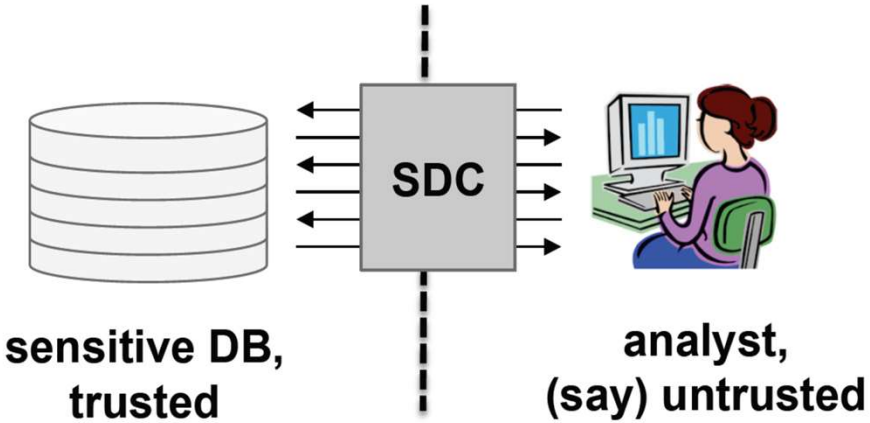
What is privacy?

- You might still want to share some data with entities you do NOT fully trust
  - e.g., epidemiological research
  - data from patients are both highly sensitive & potentially highly valuable for society
- In this context, could we still guarantee privacy while allowing for useful exploitation of patients' data?
  - Impossible!
- A tradeoff question: privacy loss vs data exploitability
  - How to quantify privacy loss (& data utility)?



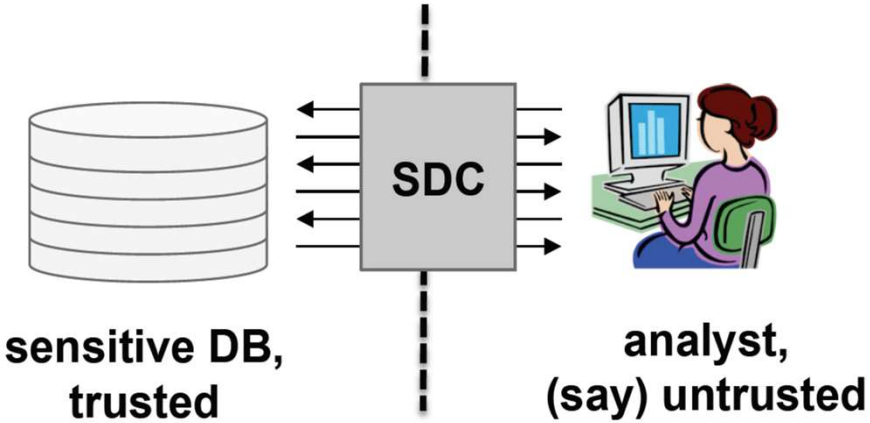
Basic model for privacy

- Add a statistical data control (SDC) component that separates the DB from the analyst and controls what info the analyst can learn.
- The key is to make the SDC enforce a rigorous definition of privacy.



- Two requirements (based on failures of other approaches):
  - Work regardless of auxiliary information
  - Persist under composition

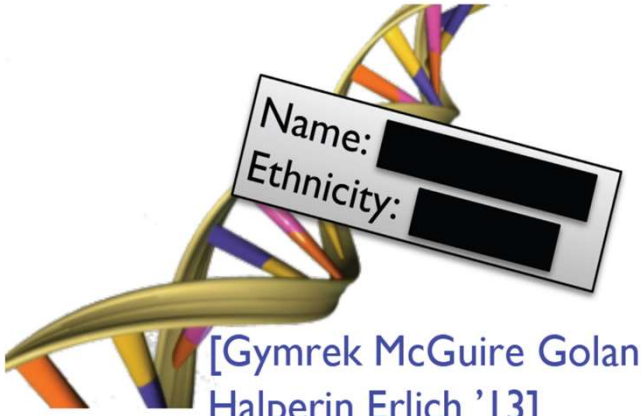
Anonymizing



Removing obvious identifiers

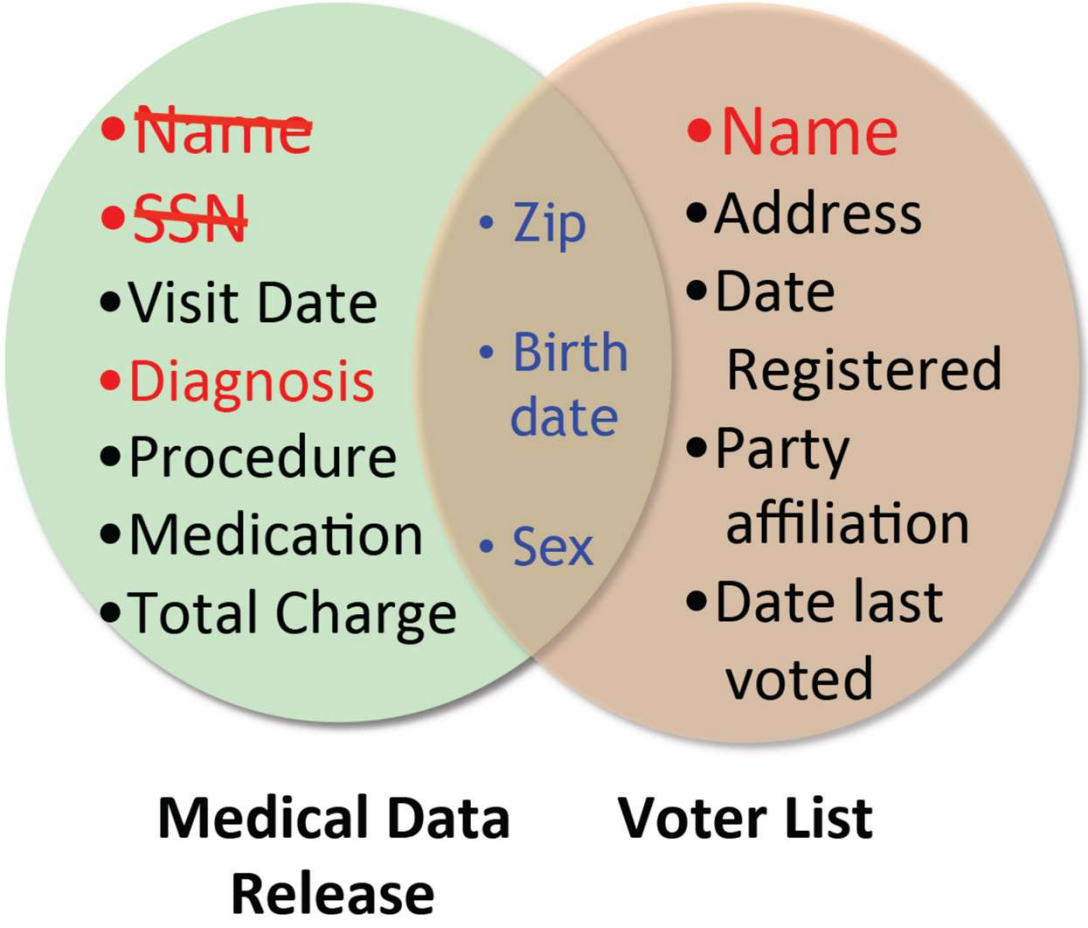
Name	Birth Date	Gender	ZIP	Disease
Alice	1960/01/01	F	10000	flu
Bob	1965/02/02	M	20000	dyspepsia
Cathy	1970/03/03	F	30000	pneumonia
David	1975/04/04	M	40000	gastritis

Medical Records



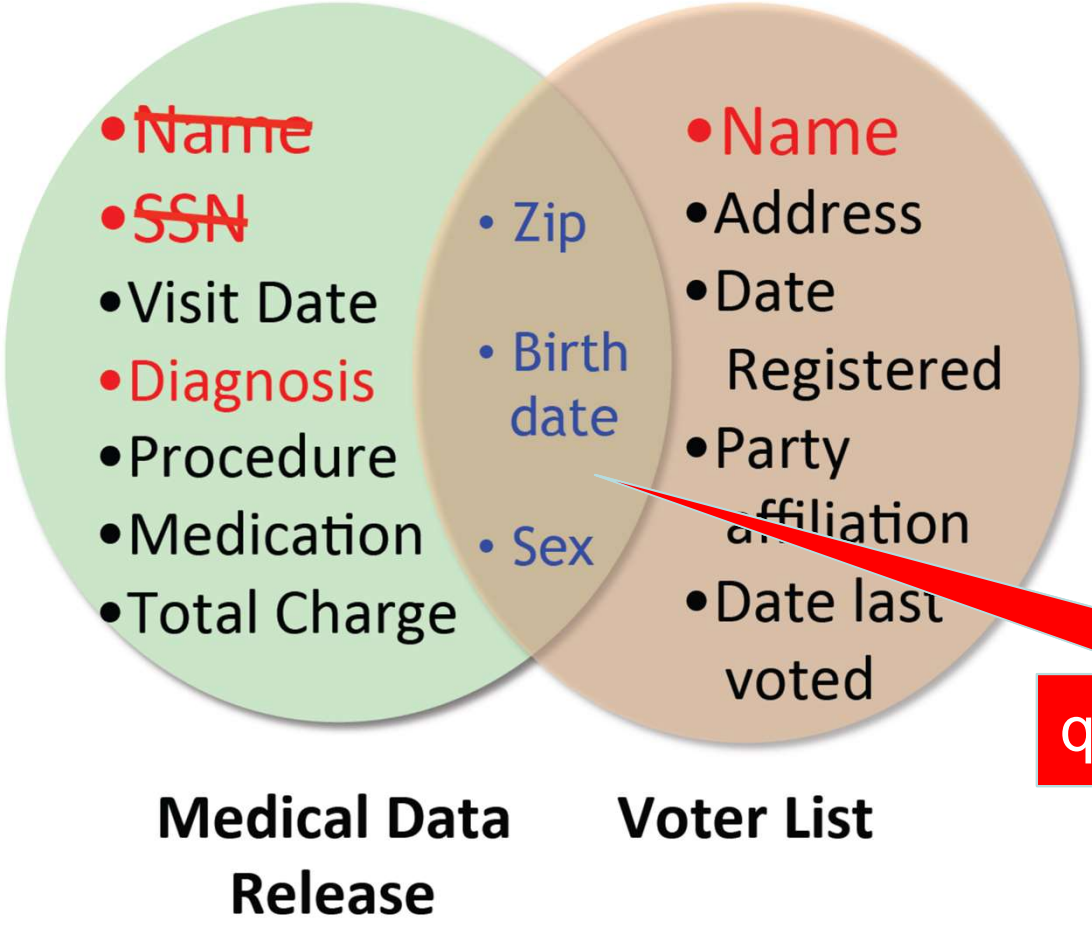
[Gymrek McGuire Golan Halperin Erlich '13]

The Massachusetts case



- Governor of MA uniquely identified using Zip Code, Birth Date, and Sex
- Name linked to Diagnosis!

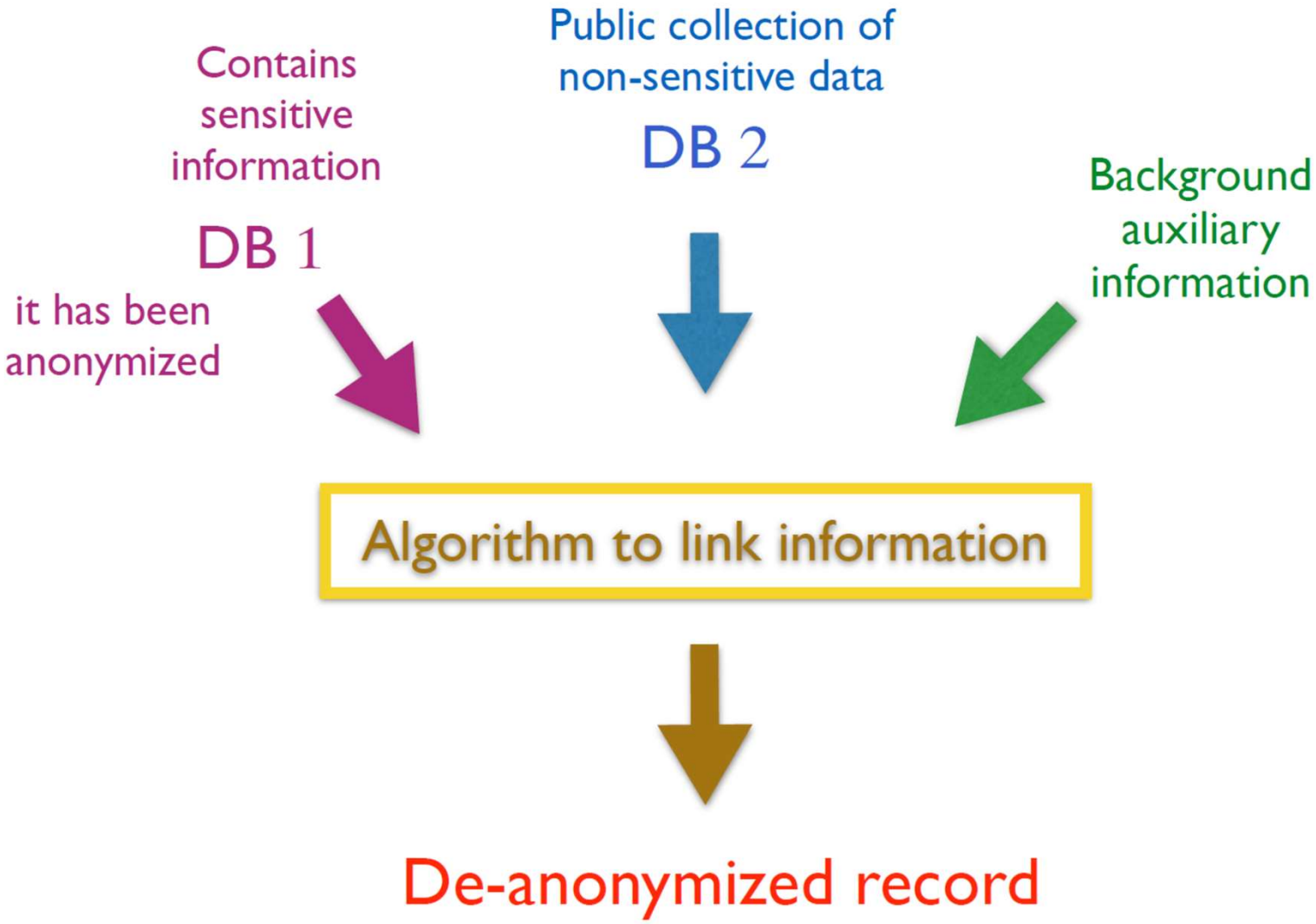
The Massachusetts case



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

quasi identifier attributes

De-anonymization Attack by Linking



k-Anonymity

- Every row should look like k-1 other rows based on the quasi identifier attributes

Zip	Age	Nationality	Disease
13053	28	Russian	Heart
13068	29	American	Heart
13068	21	Japanese	Cancer
13053	23	American	Cancer
14853	50	Indian	Cancer
14853	55	Russian	Heart
14850	47	American	Flu
14850	59	American	Flu
13053	31	American	Cancer
13053	37	Indian	Cancer
13068	36	Japanese	Cancer
13068	32	American	Cancer

Zip	Age	Nationality	Disease
130**	<30	*	Heart
130**	<30	*	Heart
130**	<30	*	Cancer
130**	<30	*	Cancer
1485*	>40	*	Cancer
1485*	>40	*	Heart
1485*	>40	*	Flu
1485*	>40	*	Flu
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer
130**	30-40	*	Cancer

Problem: background knowledge

- Adversary knows prior knowledge about Alice: 35 years old, zip code 13012
- Adversary learns Alice has cancer

	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<30	*	AIDS
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	130**	≥40	*	Cancer
6	130**	≥40	*	Heart Disease
7	130**	≥40	*	Viral Infection
8	130**	≥40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Problem: composition

- Adversary knows that: Alice/28 years old/zip code 13012; Alice's records appears in both tables

	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<30	*	AIDS
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	130**	≥40	*	Cancer
6	130**	≥40	*	Heart Disease
7	130**	≥40	*	Viral Infectic
8	130**	≥40	*	Viral Infectic
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

- Adversary learns Alice has AIDS

	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<35	*	AIDS
2	130**	<35	*	Tuberculosis
3	130**	<35	*	Flu
4	130**	<35	*	Tuberculosis
5	130**	<35	*	Cancer
6	130**	<35	*	Cancer
7	130**	≥35	*	Cancer
8	130**	≥35	*	Cancer
9	130**	≥35	*	Cancer
10	130**	≥35	*	Tuberculosis
11	130**	≥35	*	Viral Infection
12	130**	≥35	*	Viral Infection

Data reconstruction attack

Age	Gender	Employed?	Count
<18	M	Yes	x1
<18	M	No	x2
<18	F	Yes	x3
<18	F	No	x4
≥18	M	Yes	x5
≥18	M	No	x6
≥18	F	Yes	x7
≥18	F	No	x8

Age	Employed?	Count
<18	Yes	-
<18	No	2
≥18	Yes	3
≥18	No	-

Marginal 1

Age	Gender	Count
<18	M	3
<18	F	-
≥18	M	-
≥18	F	2

Marginal 2

$$\begin{aligned} & minimize\ \epsilon \\ subject\ to\quad & 2-\epsilon \le x_2 + x_4 \le 2+\epsilon \\ & 3-\epsilon \le x_5 + x_7 \le 3+\epsilon \\ & \dots \\ & x_i, \epsilon \ge 0 \end{aligned}$$

Data reconstruction attack

Age	Gender	Employed?	Count
<18	M	Yes	x1
<18	M	No	x2
<18	F	Yes	x3
<18	F	No	x4
≥18	M	Yes	x5
≥18	M	No	x6
≥18	F	Yes	x7
≥18	F	No	x8

Age	Employed?	Count
<18	Yes	-
<18	No	2
≥18	Yes	3
≥18	No	-

Marginal 1

Age	Gender	Count
<18	M	3
<18	F	-
≥18	M	-
≥18	F	2

Marginal 2

- 这样的重构结果有多准确?
- [Dinur and Nissim, 2003]中的结论:
  - 哪怕每个统计数据都有着相当程度的噪声，只要有足够多的统计数据，我们总能重构出源数据中的大部分元组

Data reconstruction attack

Age	Gender	Employed?	Count
<18	M	Yes	x1
<18	M	No	x2
<18	F	Yes	x3
<18	F	No	x4
≥18	M	Yes	x5
≥18	M	No	x6
≥18	F	Yes	x7
≥18	F	No	x8

Age	Employed?	Count
<18	Yes	-
<18	No	2
≥18	Yes	3
≥18	No	-

Marginal 1

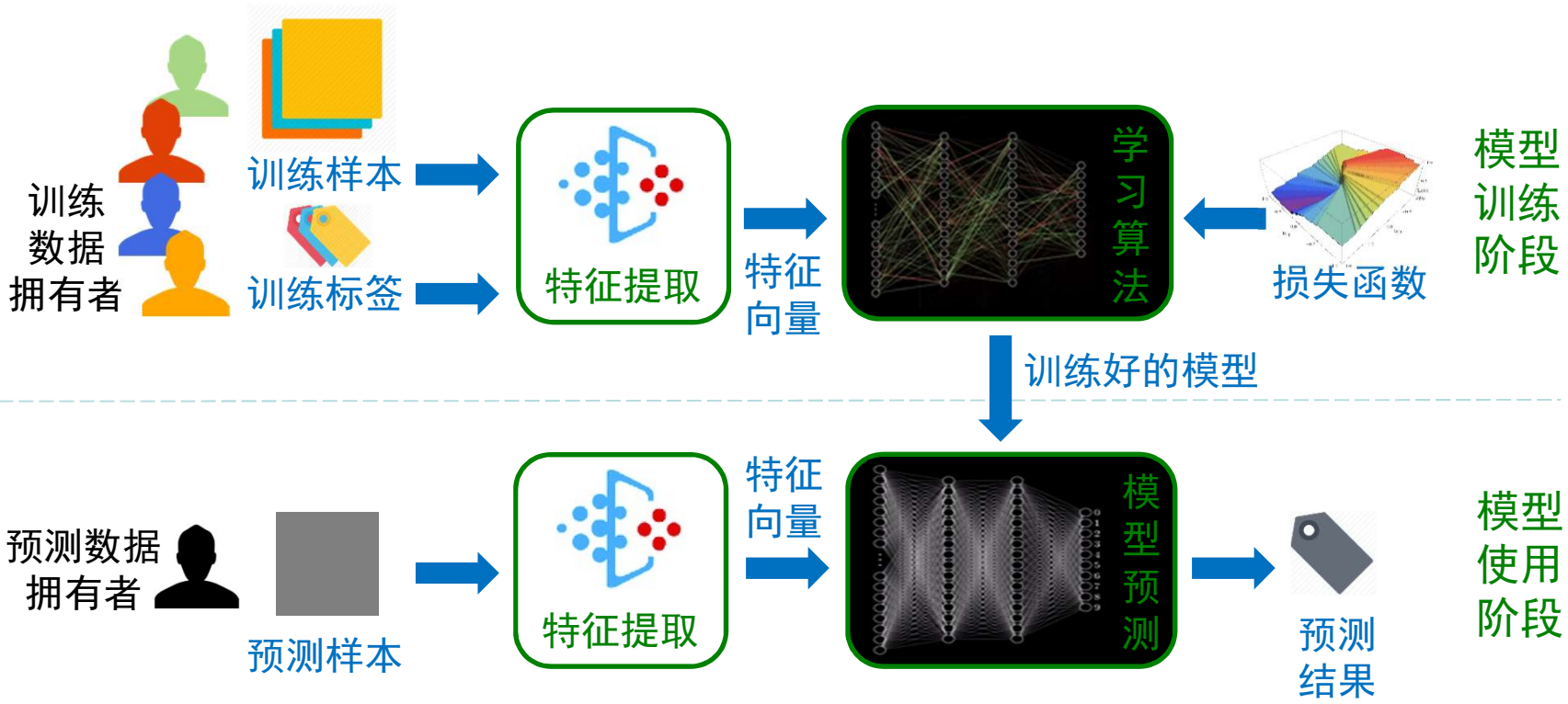
Age	Gender	Count
<18	M	3
<18	F	-
≥18	M	-
≥18	F	2

Marginal 2

- 美国普查局用他们2010年所发布的一组统计数据试验了数据重构攻击
- 结果表明，他们能重构17%美国人口的数据

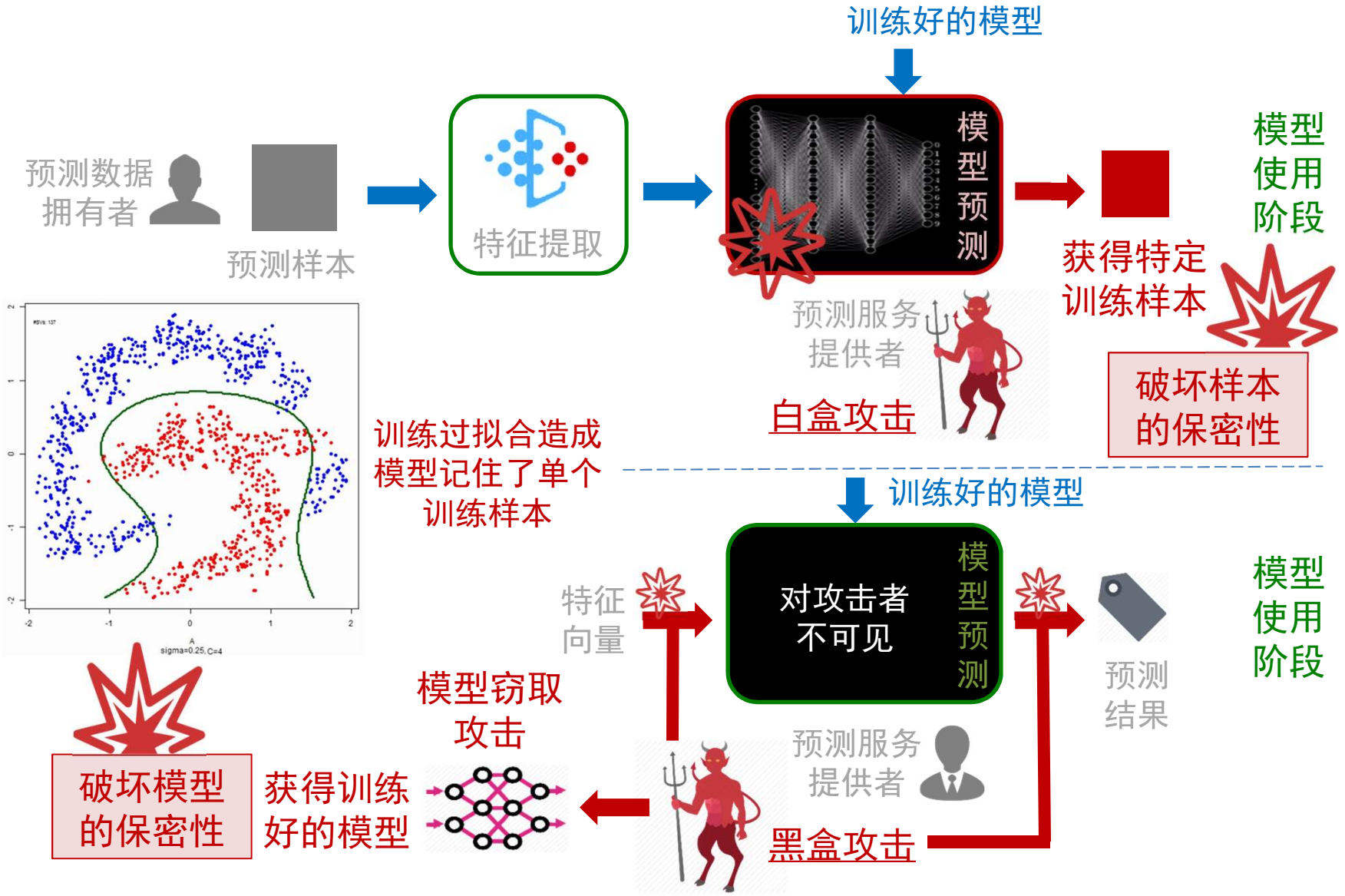
https://www.census.gov/content/dam/Census/newsroom/press-kits/2019/jsm/presentation-deploying-differential-privacy-for-the-2020-census-of-pop-and-housing.pdf

Does ML also leak?



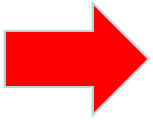
- 只公开训练好的模型可否？
- 是否就不会泄露提供训练样本的数据拥有者的隐私？

Does ML also leak? Very likely



A long journey for privacy …

- *k*-anonymity
- *l*-diversity
- *t*-closeness
- *delta*-presence
- *m*-invariance
- (*alpha*, *k*)-anonymity
- …

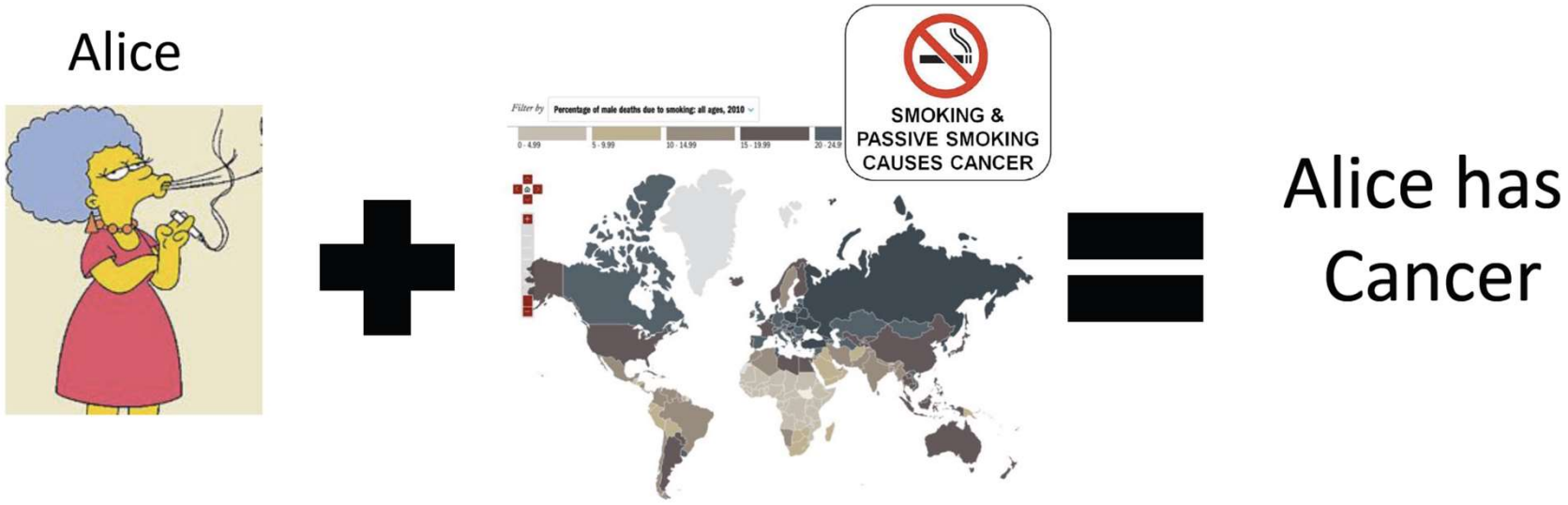


- A provable privacy guarantee
- No assumption on background knowledge
- From syntactic privacy to semantic privacy

Semantic privacy: first attempt

- Access to the results of the query should not enable one to learn anything about any individual that one would not learn without access to the results [Dalenius-77].
- Can be formalized, meets the two requirements, and maps well onto semantic security, crypto's gold standard of message secrecy [Goldwasser-82].
- Problem: this privacy definition is NOT achievable for the statistical data publishing setting (but it is for message secrecy setting!) [Dwork-10].
  - Intuitively, the goal of statistical analysis is to teach us things about people, but this definition prevents us from learning anything about anyone!

Semantic privacy: first attempt

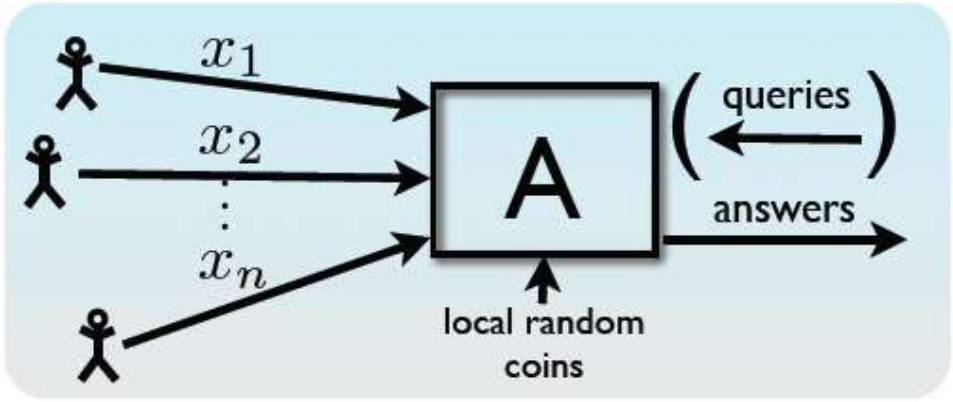


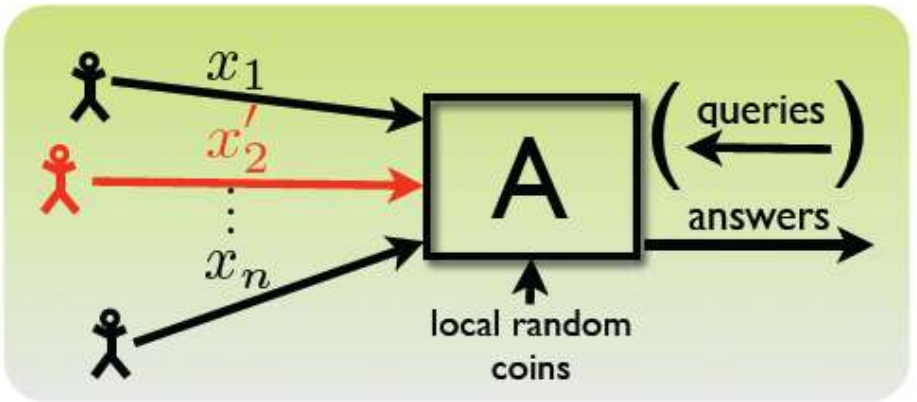
- Is this a privacy breach? NO! [Dwork-10].
- The goal of privacy-preserving statistical analysis
  - can learn things about people
  - but cannot reveal any info that is specific to any individual

Semantic privacy: second attempt

- How should we amend the definition to be more in line with our goals?
- Weaker than Semantic Security:
  - Access to the results of the query should not enable one to learn anything new confidently about any individual in the dataset that one would not learn if the individual were not in the dataset.
- But what do "anything new" and "confidently" mean?
- Differential privacy formalizes this.

Differential privacy intuition



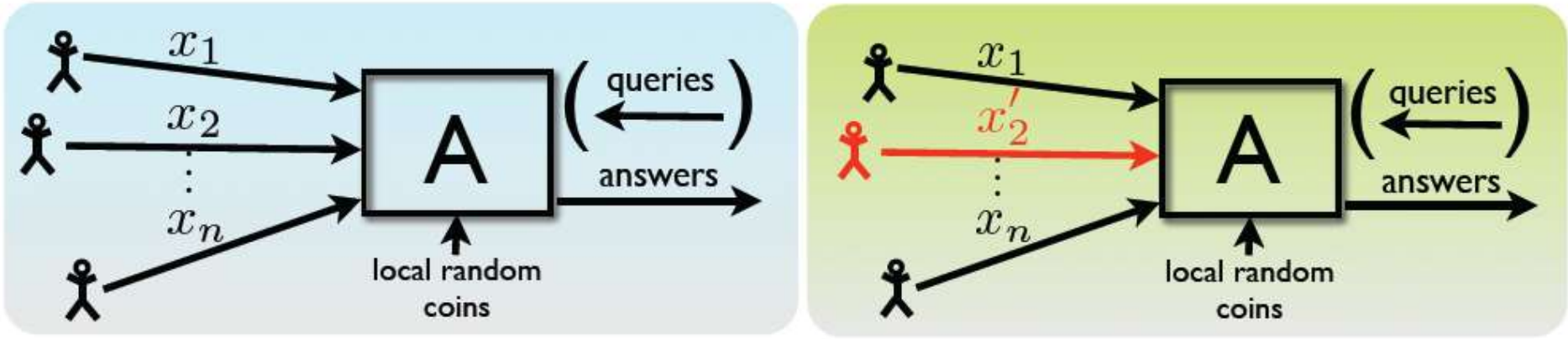


x' is a neighbor of x
if they differ in one row

From the released statistics, it is hard to tell which case it is.

Cynthia Dwork
@ Microsoft

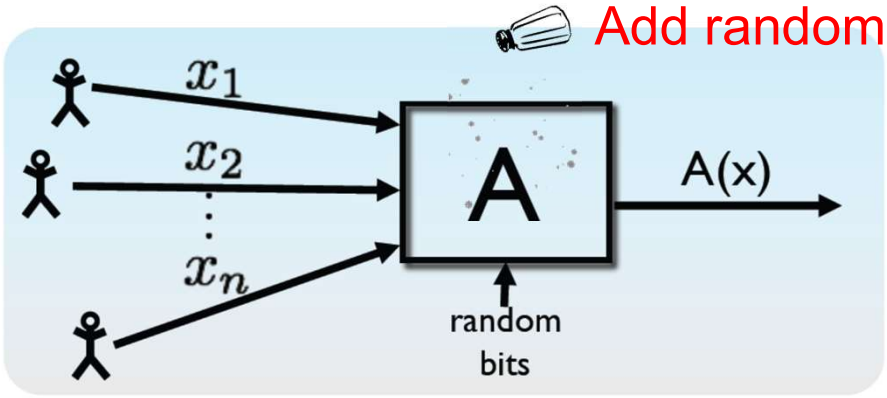
Differential privacy definition



x' is a neighbor of x
if they differ in one row

- For all neighboring databases x and x'
- For all subsets of outputs:
- $\Pr[A(x) \in S] \leq e^{\varepsilon} \Pr[A(x') \in S]$

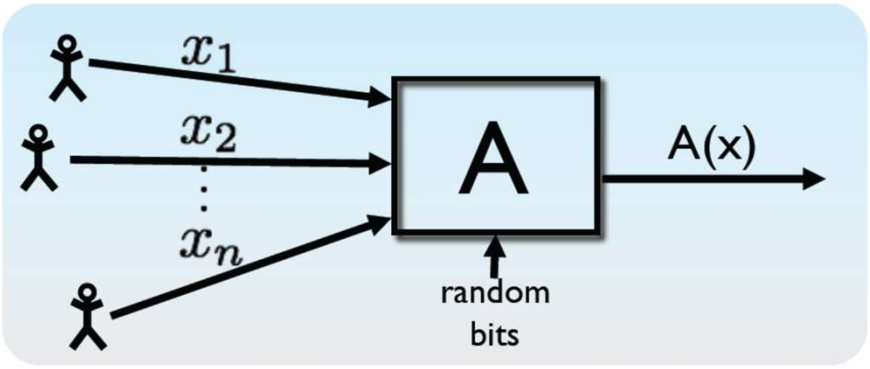
Understanding differential privacy

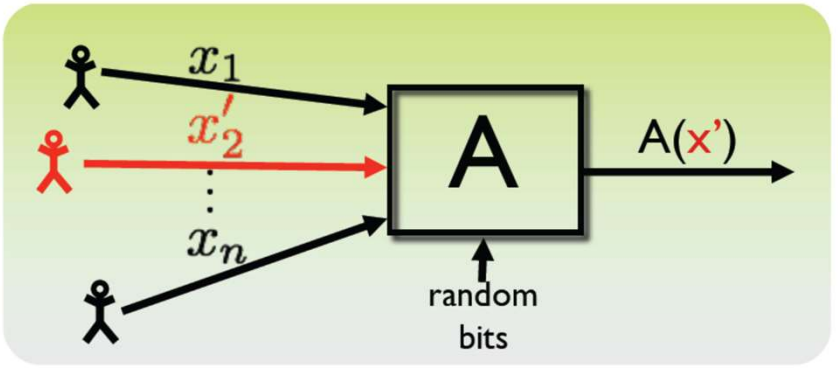


- Can deterministic algorithms satisfy DP?
- NO!

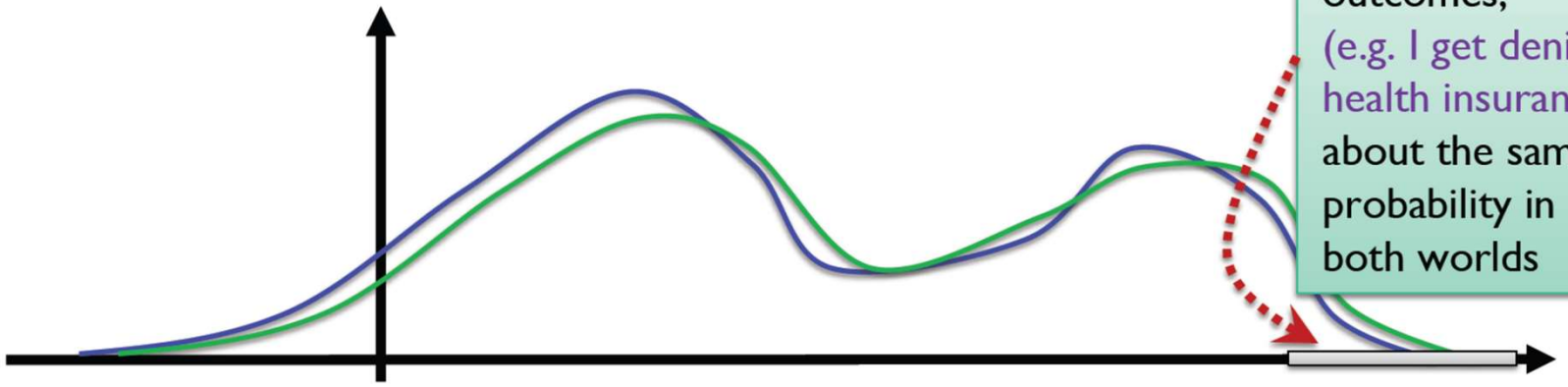
- Data set $x = (x_1, \ldots, x_n) \in \mathcal{X}$
  - Domain $\mathcal{X}$ can be numbers, categories, tax forms
  - Think of x as **fixed** (not random)
- A = **probabilistic** procedure
  - $A(x)$ is a random variable
  - Randomness might come from adding noise, resampling, etc.

Understanding differential privacy





- A thought experiment
  - Change one person's data (or add or remove them)
  - Will the **probabilities of outcomes** change?



For any set of outcomes, (e.g. I get denied health insurance) about the same probability in both worlds

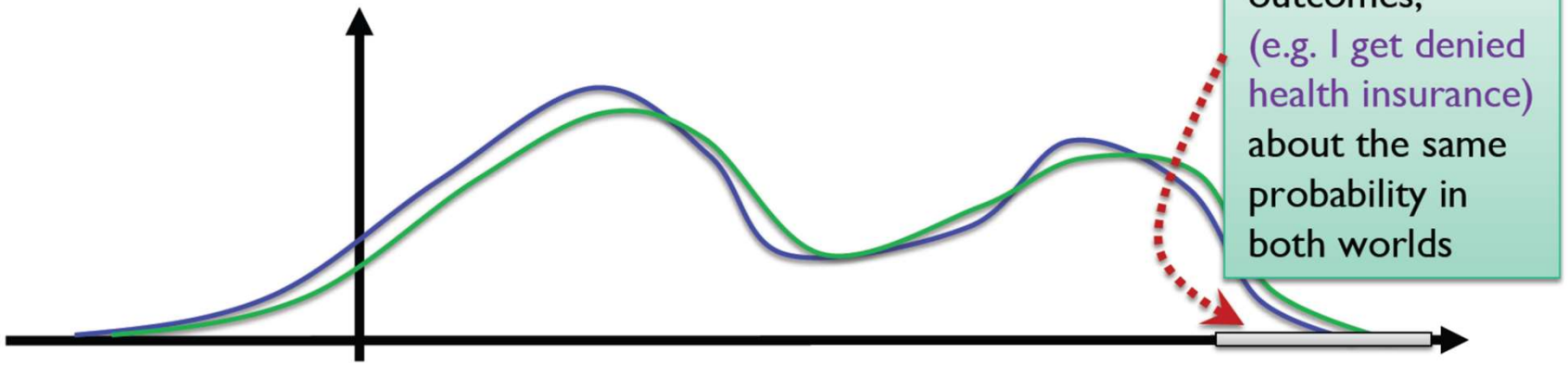
The Bayesian interpretation

$$\text{Distribution of } C(D,q_1,\ldots,q_t) \approx_{\varepsilon} \text{Distribution of } C(D',q_1,\ldots,q_t)$$

- The parameter epsilon impacts the bound on the statistical distance between the prior and the posterior distributions.
- In particular, if the adversary's prior included **all the information** about X except for the *i*'th row (the data of individual *i*), then his posterior on X_i would have been close to his prior on X_i.
- In that sense, the adversary does not learn "anything new" about *i*, i.e., anything that he couldn't have learned from the rest of the database.
- This does not mean that the adversary cannot learn anything about you from the output of a DP computation; indeed, learning about the population implies learning about individuals.

Understanding differential privacy

- A thought experiment
  - Change one person's data (or add or remove them)
  - Will the **probabilities of outcomes** change?



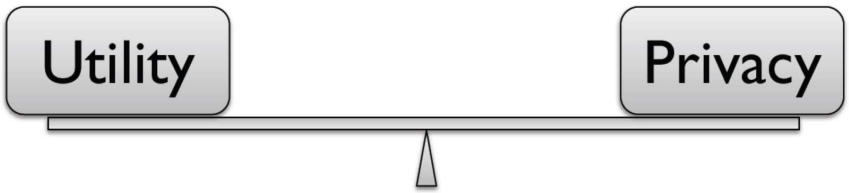
- $\varepsilon \geq 0$: small but not cryptographically small (e.g., $\varepsilon = 0.1$).
- For small ε, $e^{\varepsilon} \cong \varepsilon + 1$, so the guarantee is defined as bounding the *fractional increase* in the probability of any output from f:

$$\Pr(f(x) \in S) \leq \Pr(f(x') \in S) + \varepsilon \Pr(f(x') \in S)$$

The epsilon parameter

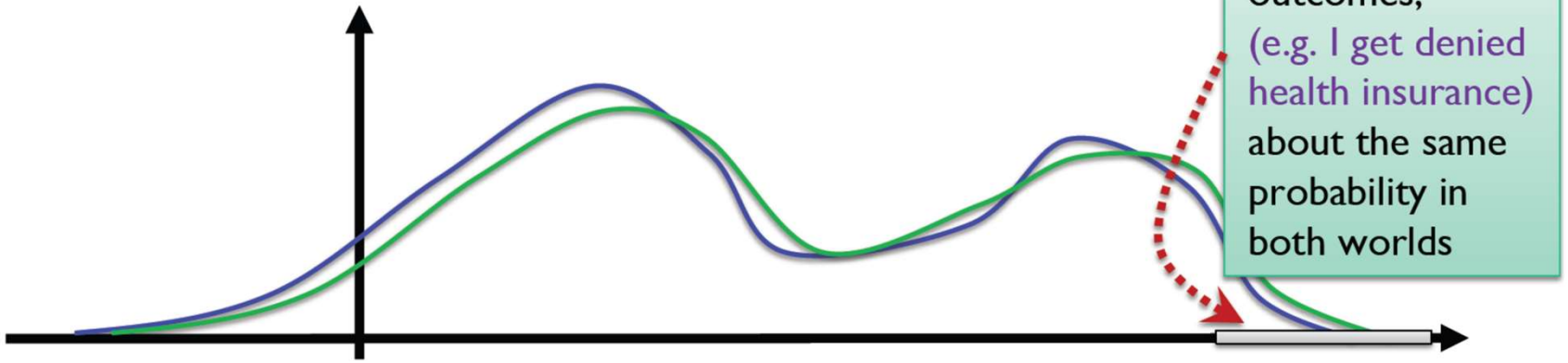
- All preceding interpretation of the protection guarantee of DP assume “low” epsilon.
- There is debate about what that should be, and concern that one would have to figure that out on a case by case basis.
- Generally speaking, smaller values mean better privacy. But they affect accuracy.
  - For example, we’ll see that for many statistical analyses, epsilon < 1/n doesn’t make sense because DP error accumulates as o(1/n).
- On the other hand, minute distinctions (low->lower) don’t seem that meaningful.
  - Consider the analogy with hypothesis tests: low p-values (generally) mean high-confidence results; but differentiating between low and even lower p-values isn’t meaningful.

  - Utility: release aggregate statistics
  - Privacy: individual information stays hidden



Understanding differential privacy

- A thought experiment
  - Change one person's data (or add or remove them)
  - Will the **probabilities of outcomes** change?



- DP stipulates that the distance between the probability distributions over the outputs shall be small when the distance between the input datasets is small.
- But why those particular choices of distance functions: hamming for inputs, multiplicative measure of distance between output probability distros?

One (bad) alternative

- What if we changed the multiplicative measure to statistical distance (total variation distance) between two distributions:

A randomized computation $f : X \to Y$ is δ-alternative-DP if $\forall x, x' \in X . d(x, x') \leq 1, \forall S \subseteq Y :$

$$\Pr(f(x) \in S) \leq \Pr(f(x') \in S) + \delta.$$

- Not a well-behaved definition: depending on delta, it either does not permit useful computations or does not provide sufficient privacy.

- Interestingly, a variation of this (bad) alternative does give us a well-behaved and valuable definition of privacy!

Approximate DP (ADP)

A randomized computation $f : X \to Y$ is (ε, δ)-DP if
$\forall x, x' \in X . d(x, x') \leq 1, \forall S \subseteq Y :$

$$\Pr(f(x) \in S) \leq e^{\varepsilon} \Pr(f(x') \in S) + \delta.$$

The probabilities are taken over the randomness in f.

- $\varepsilon \geq 0$: small but not cryptographically small (e.g., $\varepsilon = 0.1$).
- $\delta \in [0,1]$: aim for cryptographically negligible, because it loosely corresponds to the probability that f fails to be ε-DP (i.e., leaks more than e^{ε}).
- When δ>0, this definition is called *approximate DP*. When δ=0, this definition is called *pure DP* and is equivalent to the first DP definition we gave.

Interpretation of ADP

- It can be formalized and proven that f is (ε,δ)-DP **IFF**:

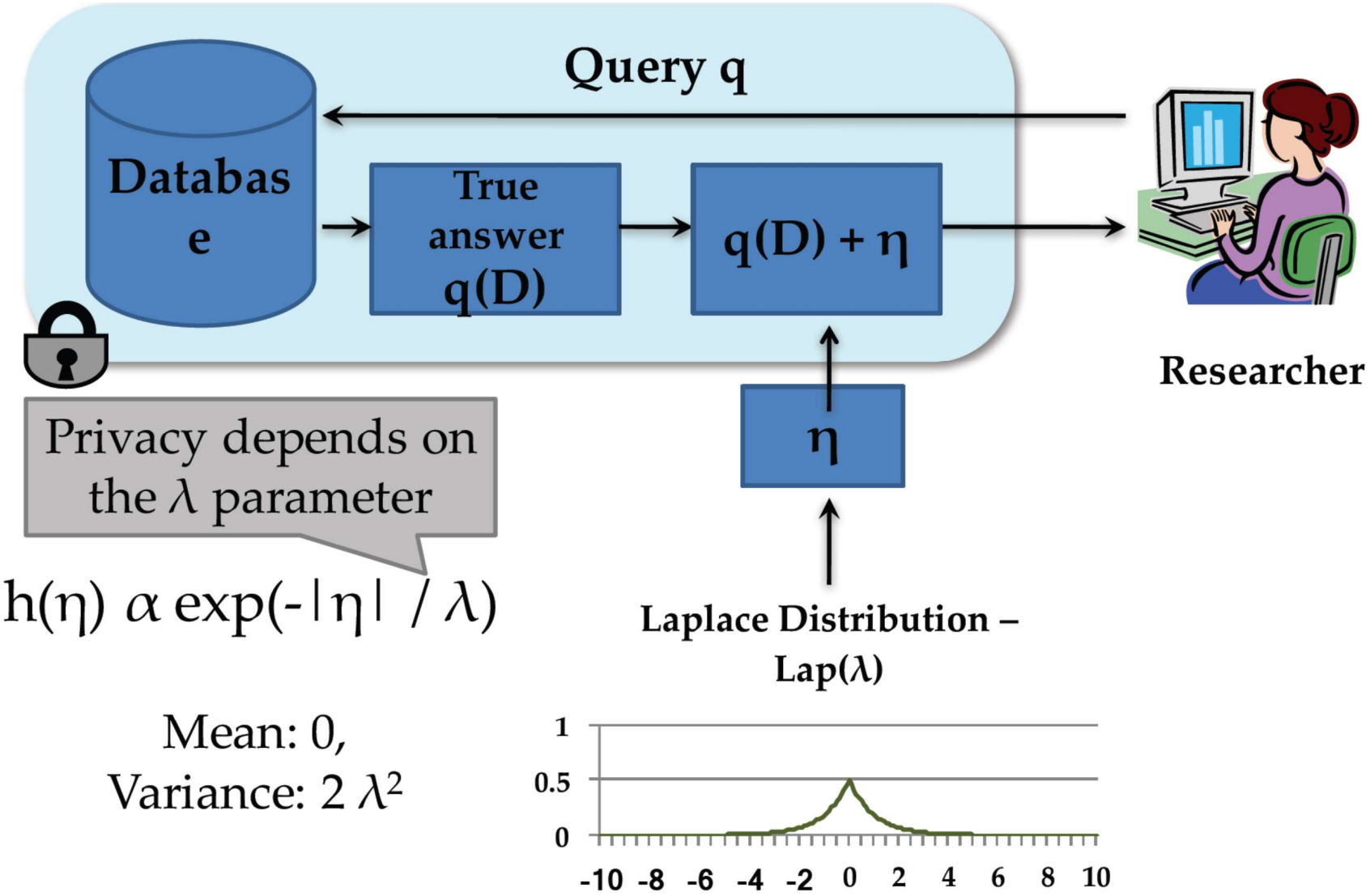
$$\forall x, x' \in X . d(x, x') \leq 1,$$

$$\exists \text{ event } BREACH . \Pr[BREACH] \leq \delta \text{ AND}$$

$$\text{apart from } BREACH, \forall S \subset Y : \Pr[f(x) \in S] \leq e^{\varepsilon} \Pr[f(x') \in S].$$

- From this, an accepted but rough interpretation of (ε,δ)-DP is as "ε-DP with probability at least (1-δ)."
- Thus, δ is the probability of an uncontrolled BREACH, and ε is the worst-case privacy loss for any individual if no BREACH.

Laplace mechanism



Laplace mechanism

Sensitivity: Consider a query q: $I \rightarrow$ R. S(q) is the smallest number s.t. for any neighboring tables D, D',

$$| q(D) - q(D') | \leq S(q)$$

Thm: If **sensitivity** of the query is **S**, then the following guarantees ε-differential privacy.

$$\lambda = S/\varepsilon$$

When Laplace doesn't make sense

- What if we have a non-numeric function?
  - "What's most common eye color in the room?"
- What if the perturbed answer isn't "almost as good as" the exact answer?
  - "Which price would bring the most money from a set of buyers?"

Exponential Mechanism

- What if L1-sensitivity is large?
  - "What's the median salary in a salary database?"

Smooth Sensitivity (and other mechanisms)

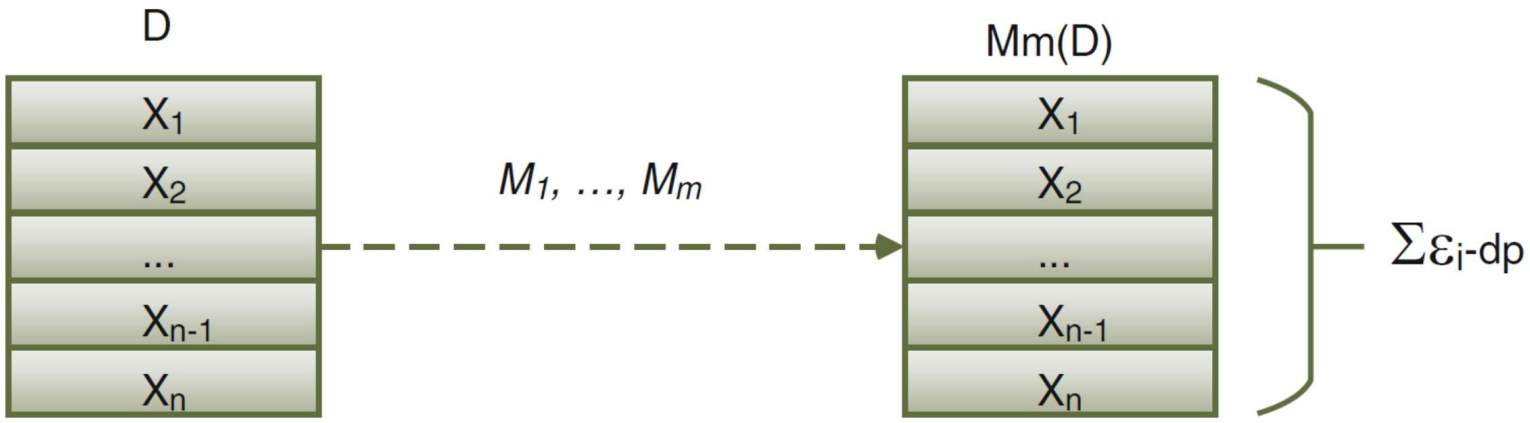
Composition properties of DP

- Why composition?
  - Reasoning about privacy of a complex algorithm is hard.
  - Helps software design
    - If building blocks are proven to be private, it about privacy of a complex algorithm built e blocks.

Composition properties of DP

- Sequential composition
  - If M_1, M_2, ..., M_k are algorithms that access a private database D such that each M_i satisfies ε_i -differential privacy,

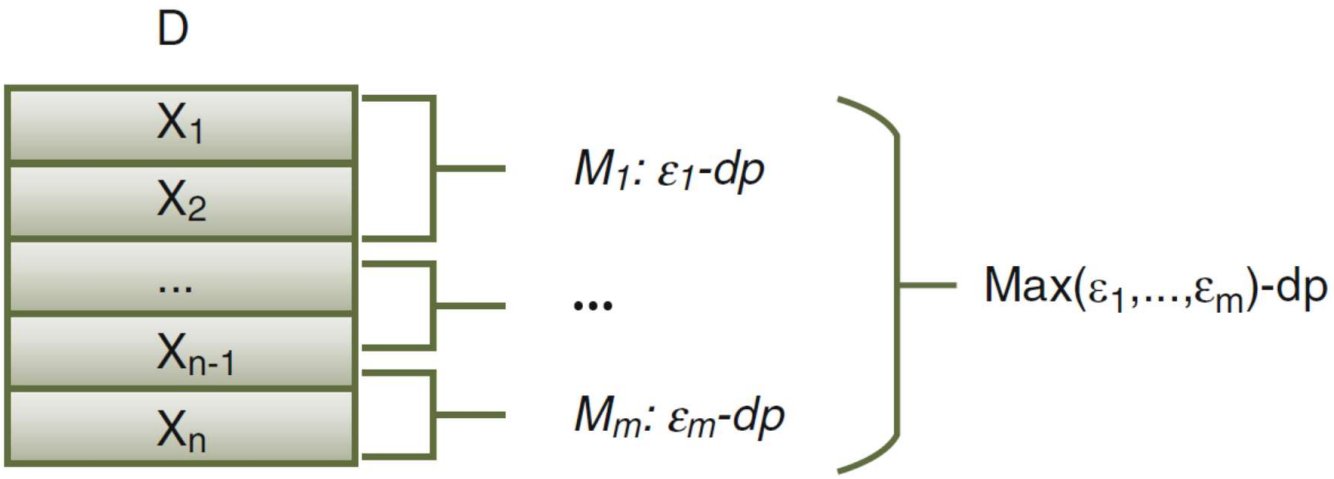
    then the combination of their outputs satisfies ε-differential privacy with $\varepsilon=\varepsilon_1+...+\varepsilon_k$



Composition properties of DP

- Parallel composition
  - If M_1, M_2, ..., M_k are algorithms that access disjoint databases D_1, D_2, ..., D_k such that each M_i satisfies ε_i -differential privacy,

    then the combination of their outputs satisfies ε-differential privacy with $\varepsilon = \max\{\varepsilon_1,...,\varepsilon_k\}$



Composition properties of DP

- Post-processing
  - If M_1 is an εdifferentially private algorithm that accesses a private database D,

    then outputting $M_2(M_1(D))$ also satisfies ε-differential privacy.

- Closure under arbitrary post-processing

A bound on the number of queries

- In order to ensure utility, a statistical database must leak some information about each individual
- We can only hope to bound the amount of disclosure

- Hence, there is a limit on number of queries that can be answered

DP: theory & practice

Theory: differential privacy research has

- many intriguing theoretical challenges
- rich connections w/other parts of CS theory & mathematics

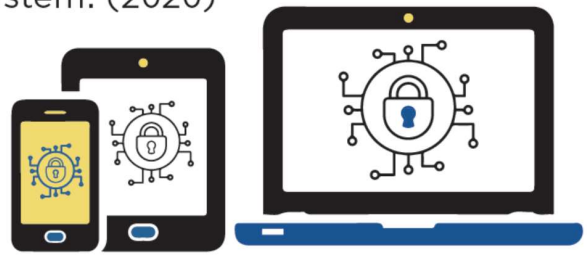
e.g. cryptography, learning theory, game theory & mechanism design, convex geometry, pseudorandomness, optimization, approximability, communication complexity, statistics, …

Practice: interest from many communities in seeing whether DP can be brought to practice

e.g. statistics, databases, medical informatics, privacy law, social science, computer security, programming languages, …

DP deployed

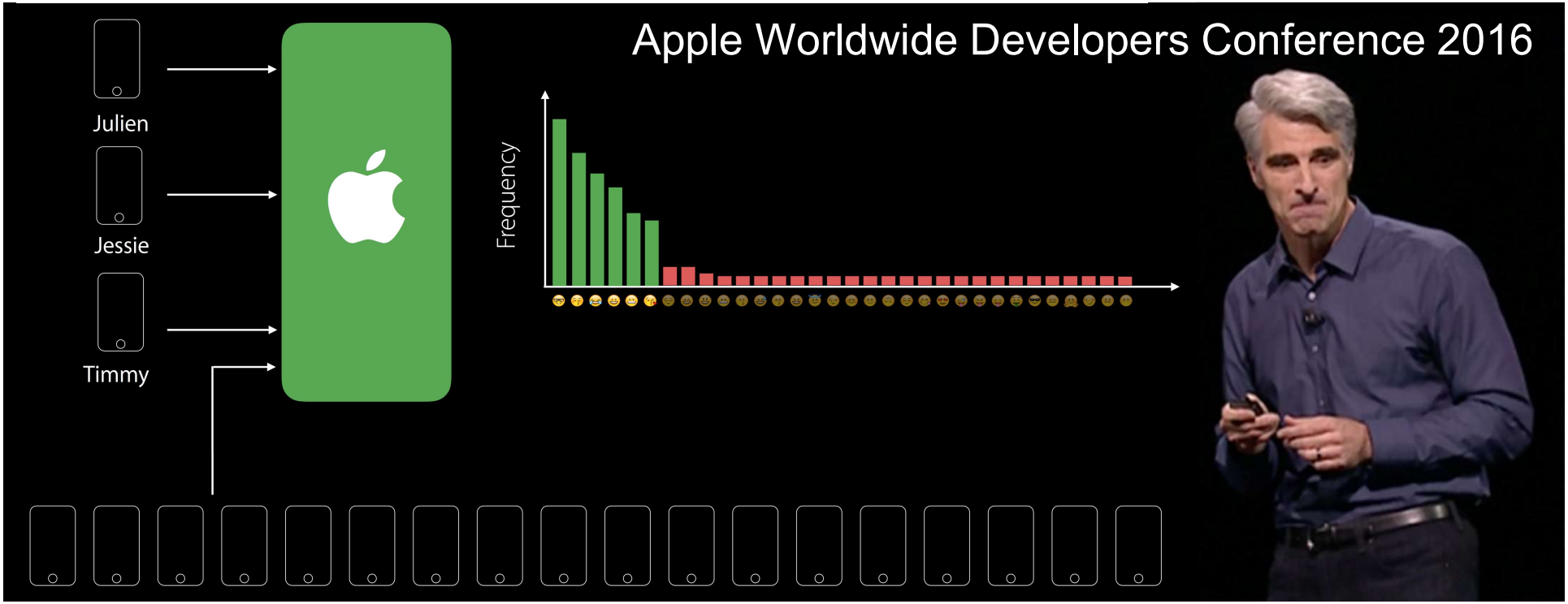
2020 Census data products will be protected using differential privacy.
World's first large-scale application of new privacy system. (2020)



United States™ Census Bureau

U.S. Department of Commerce
Economics and Statistics Administration
U.S. CENSUS BUREAU
census.gov

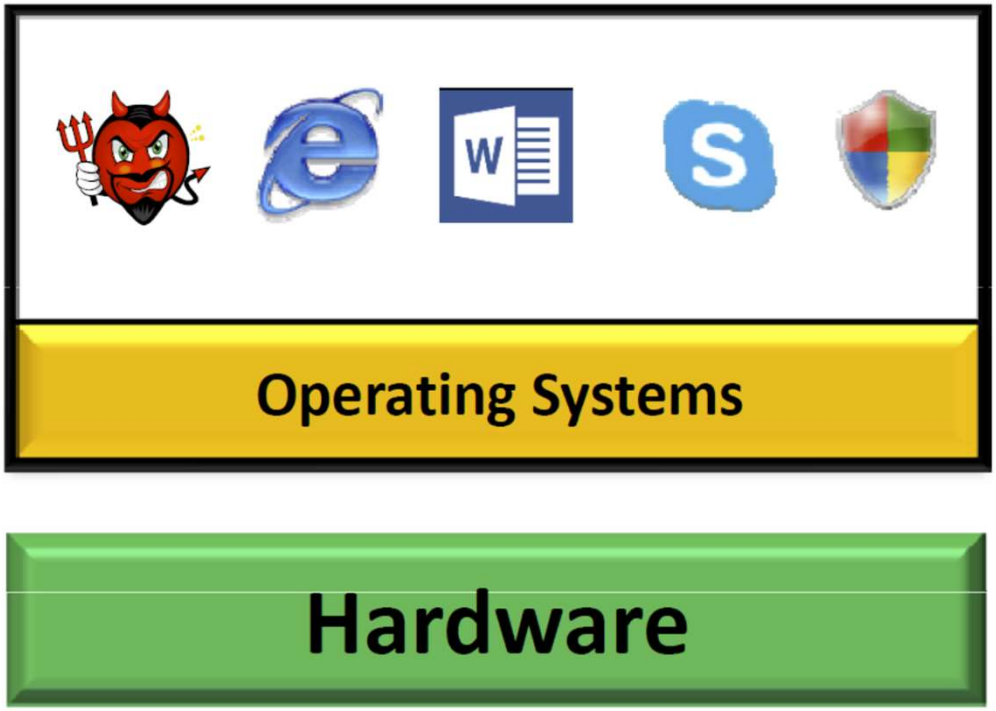
CENSUS 20/20



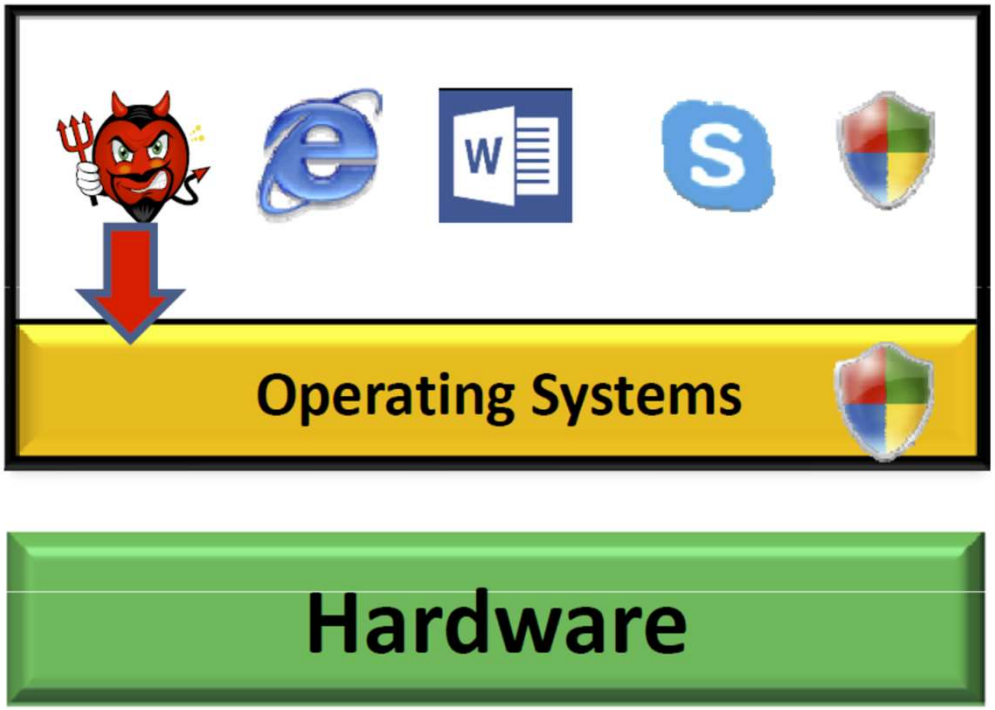
Part 7: outline

- S&P Issues for Cloud Computing
- Crypto 2.0
  - Attribute-based Encryption
  - Anonymous Credential
  - Homomorphic Encryption
- PETs
  - PIR/ORAM
  - Differential Privacy
  - Trusted Hardware-SGX

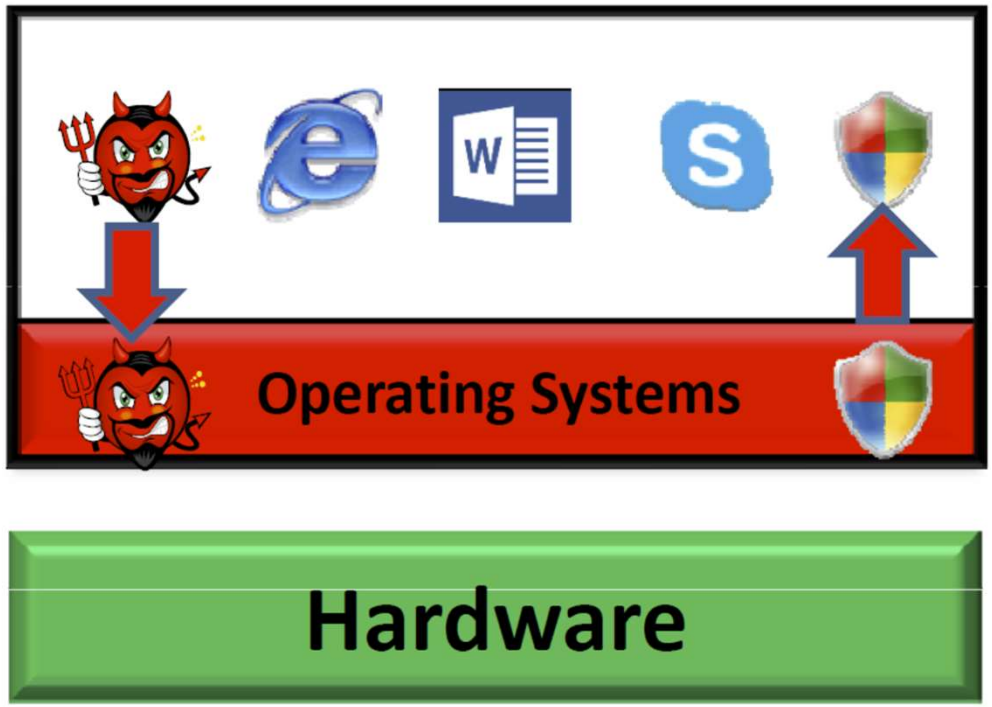
The evolutions of using isolation for malware defense



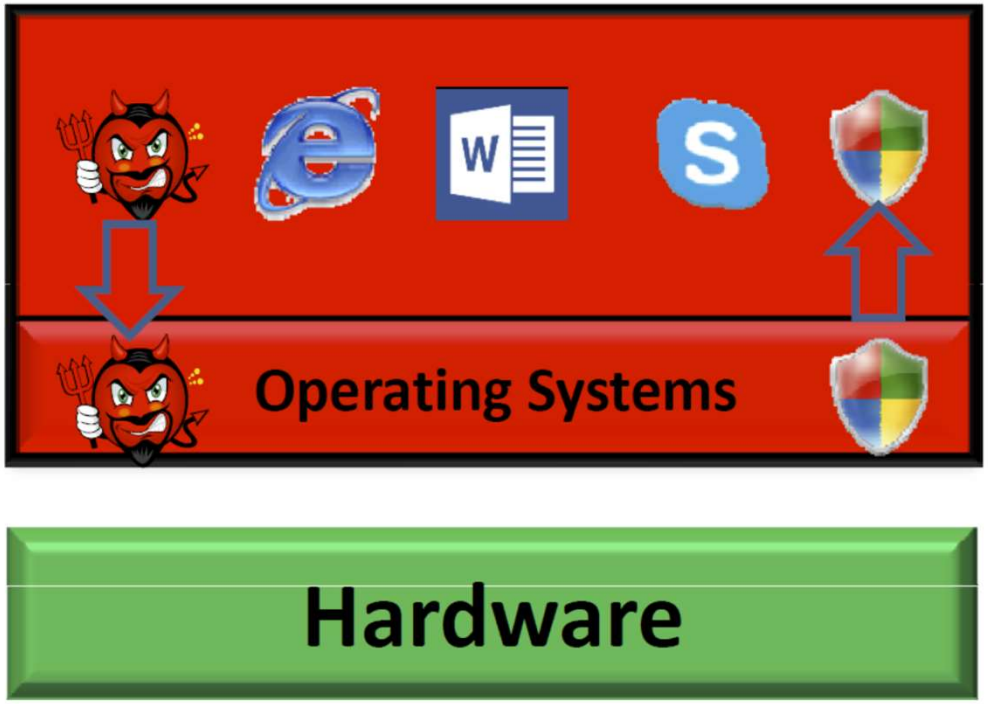
The evolutions of using isolation for malware defense



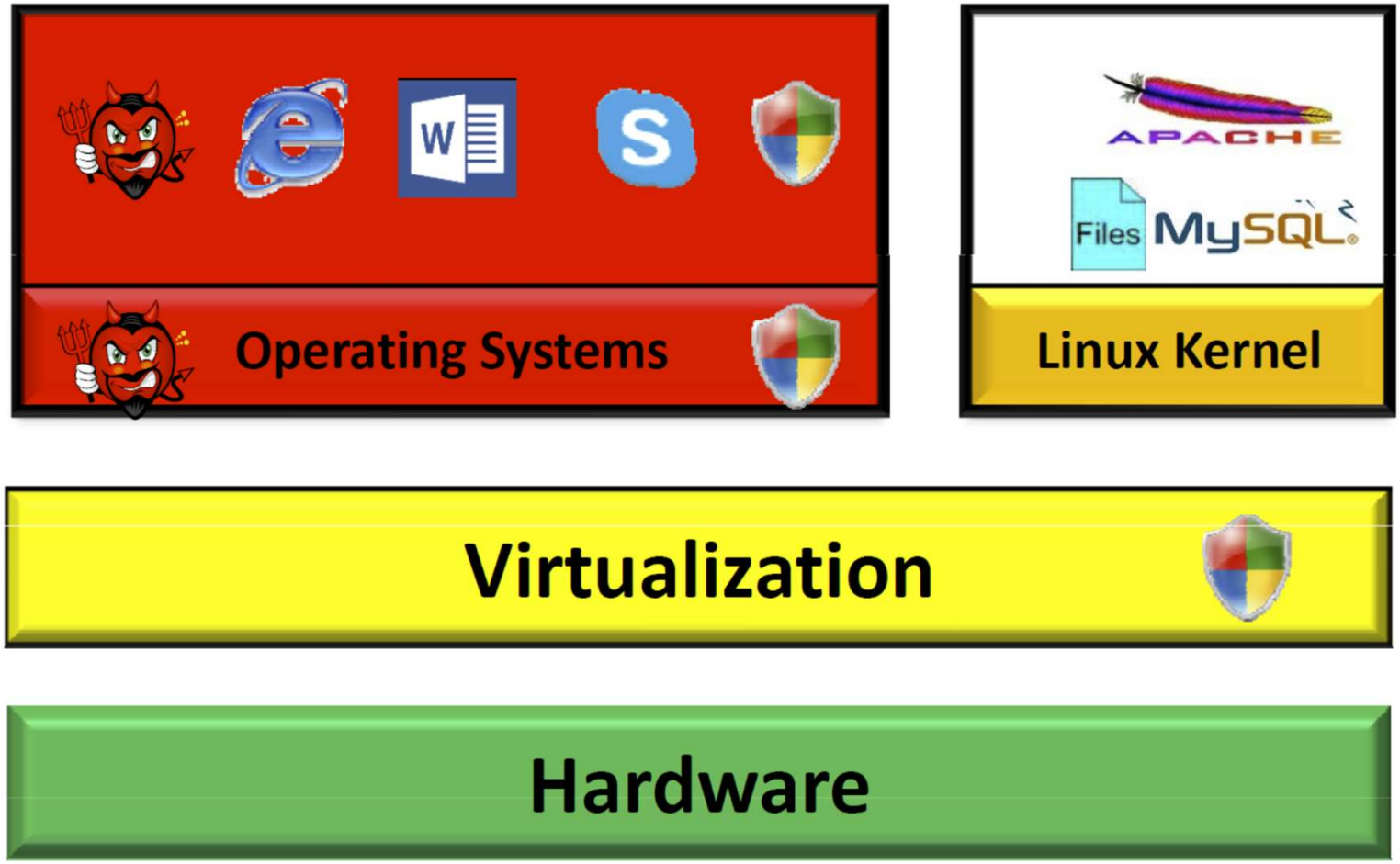
The evolutions of using isolation for malware defense



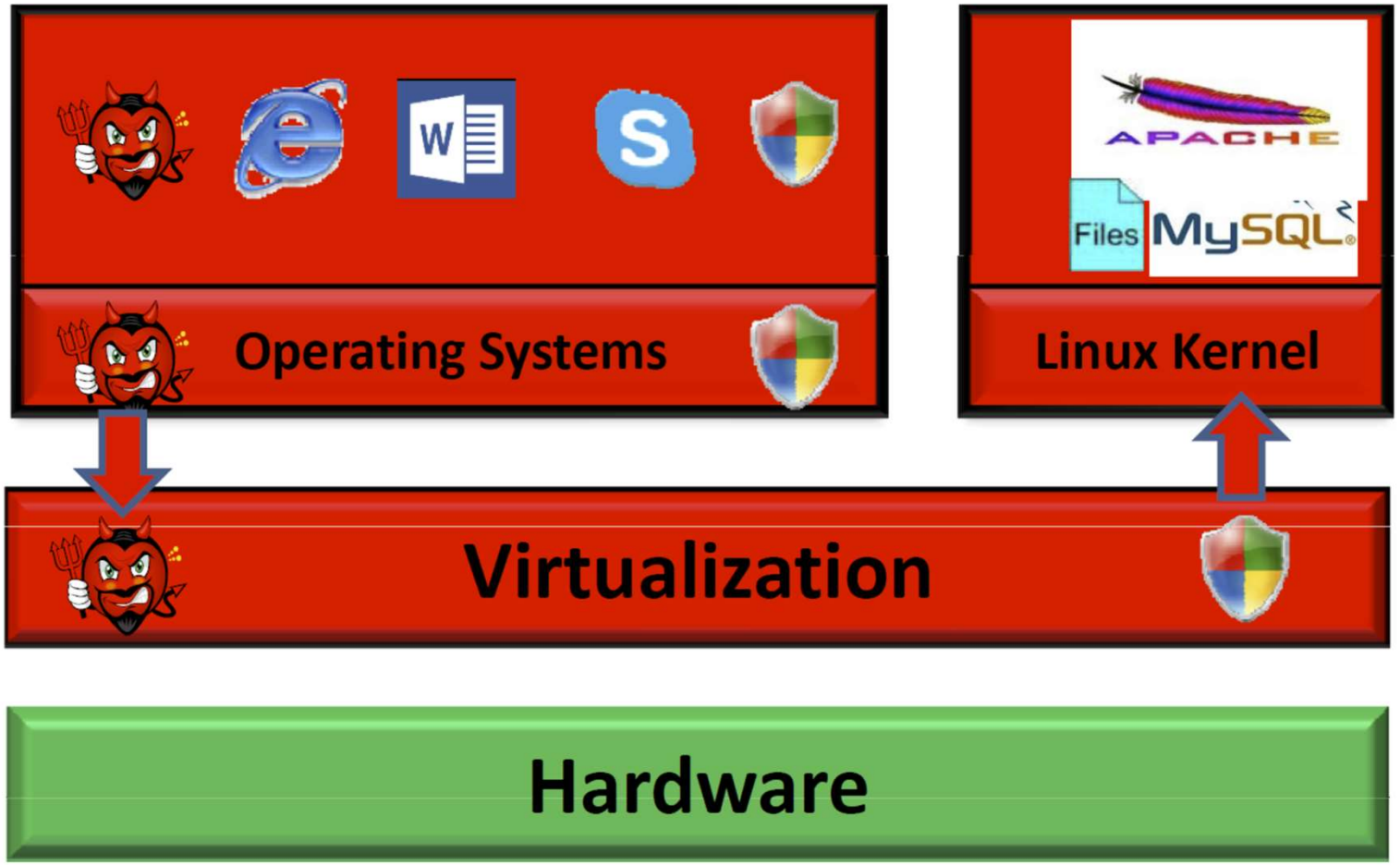
The evolutions of using isolation for malware defense



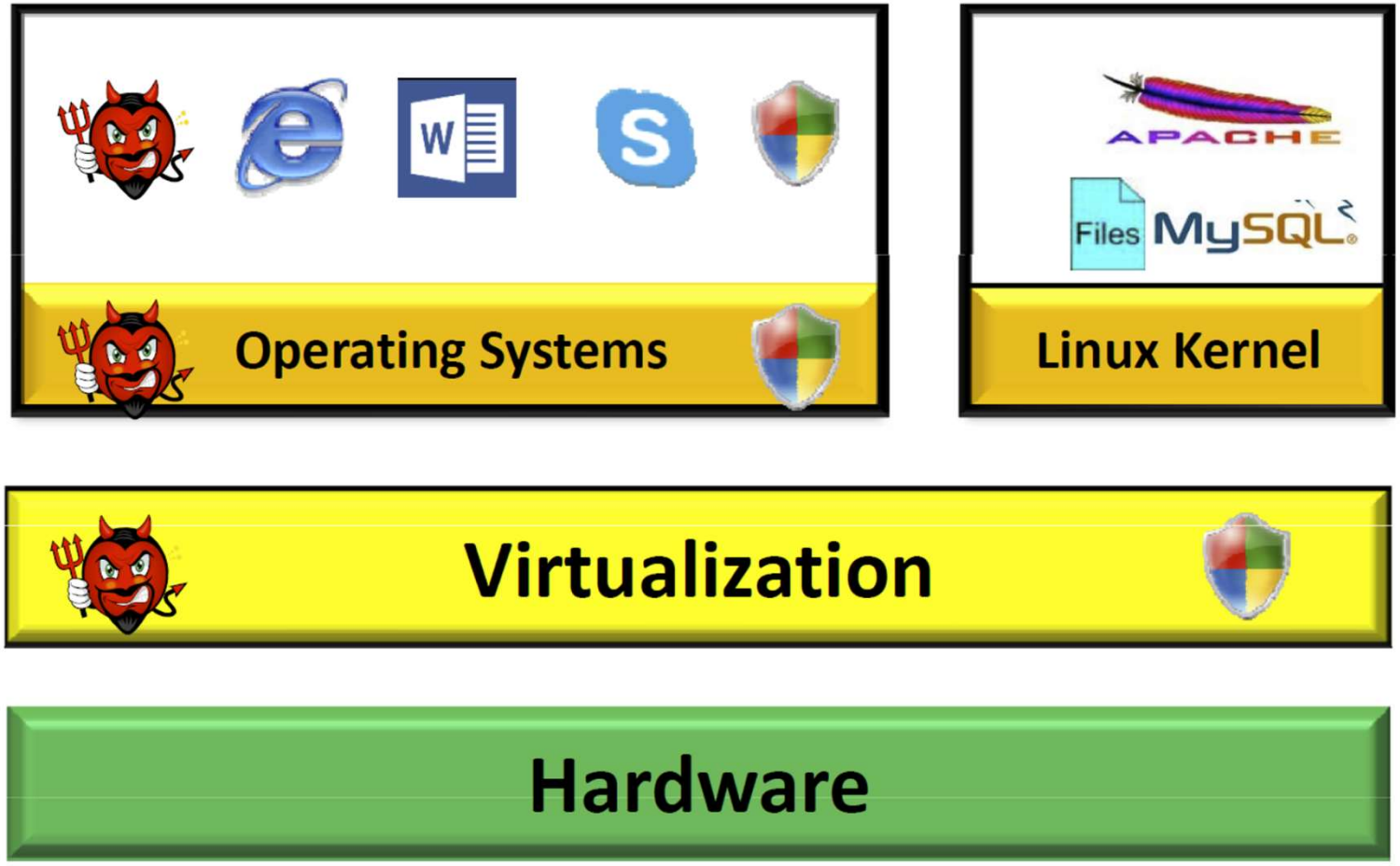
The evolutions of using isolation for malware defense



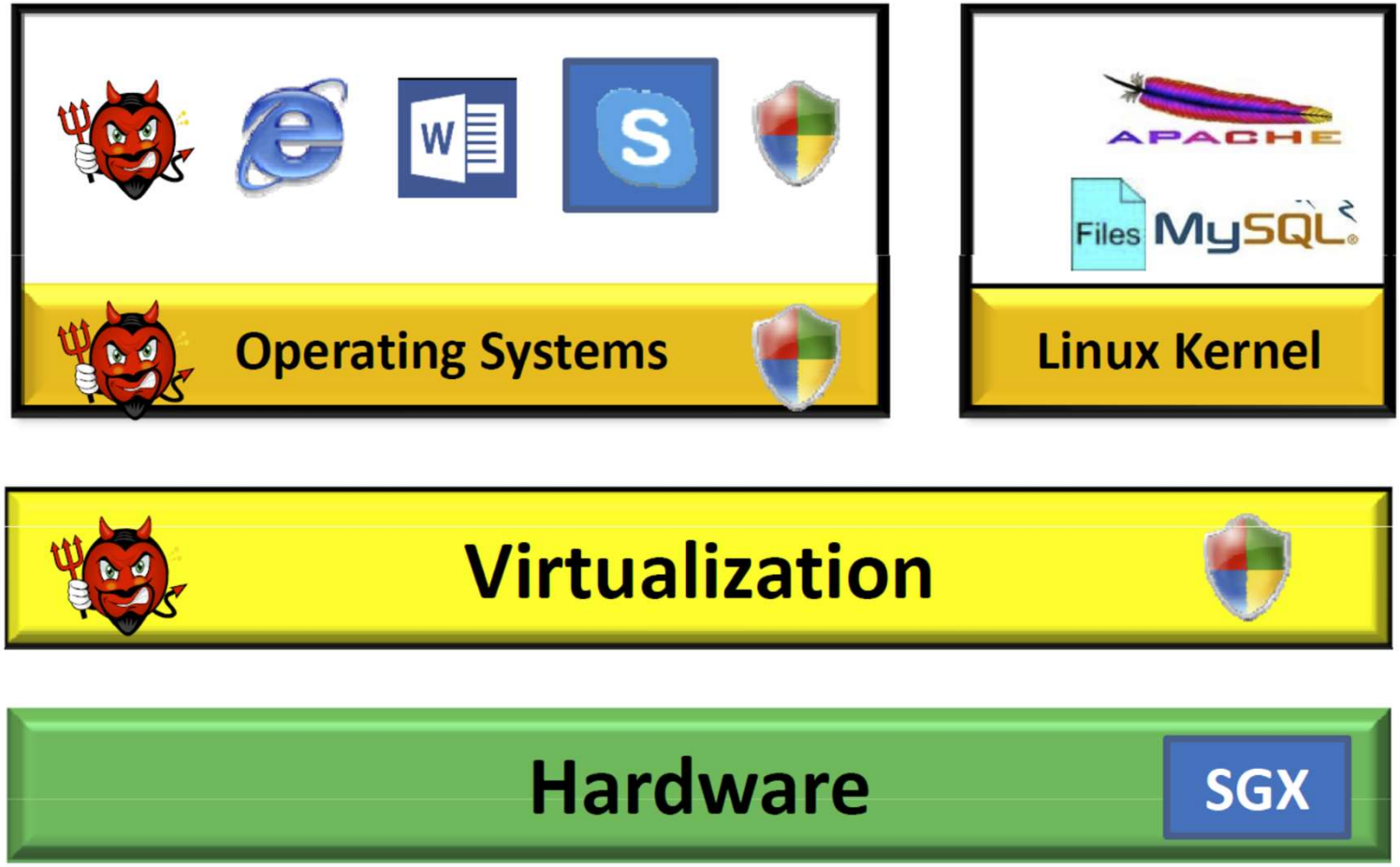
The evolutions of using isolation for malware defense



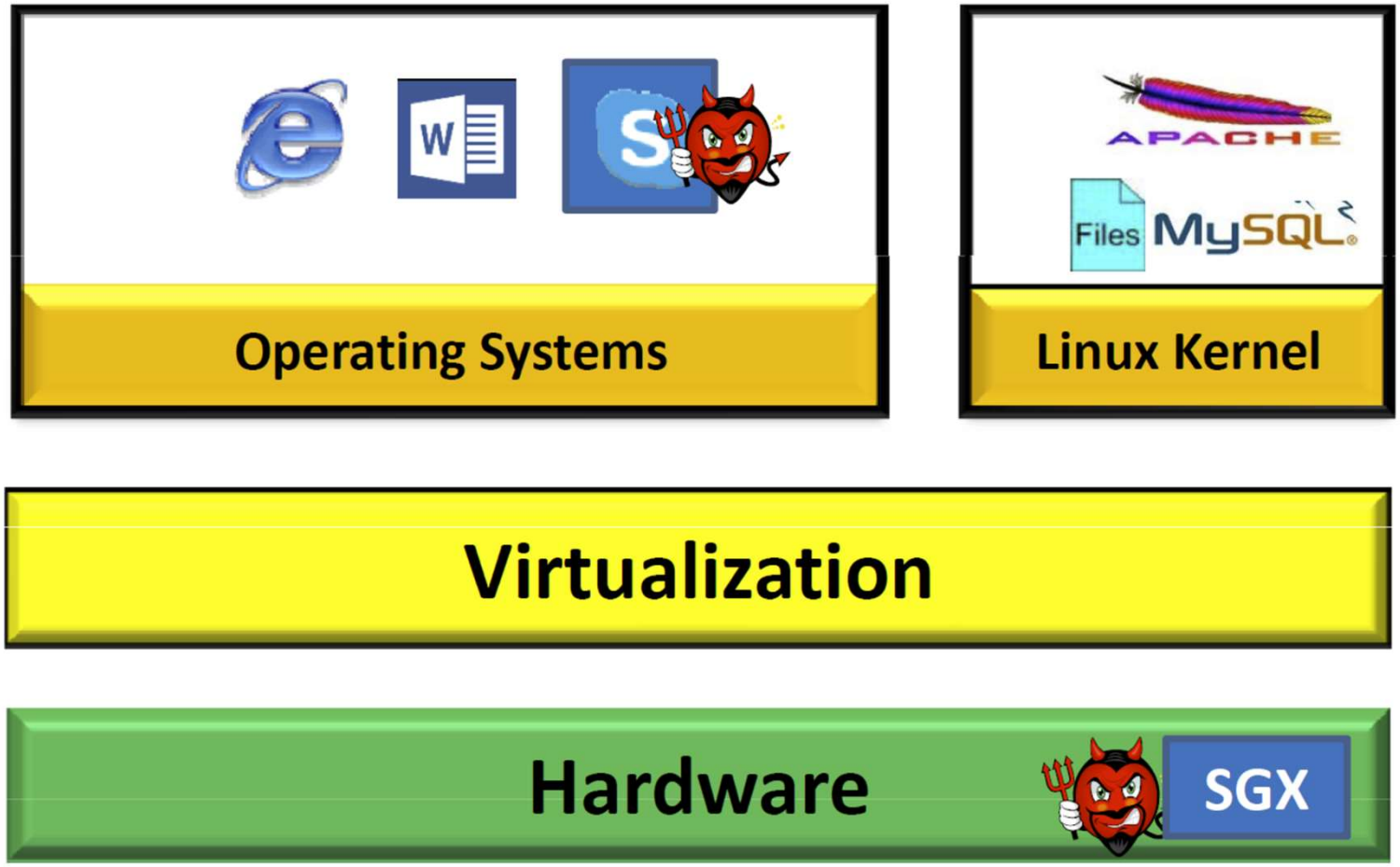
The evolutions of using isolation for malware defense



The evolutions of using isolation for malware defense



The evolutions of using isolation for malware defense

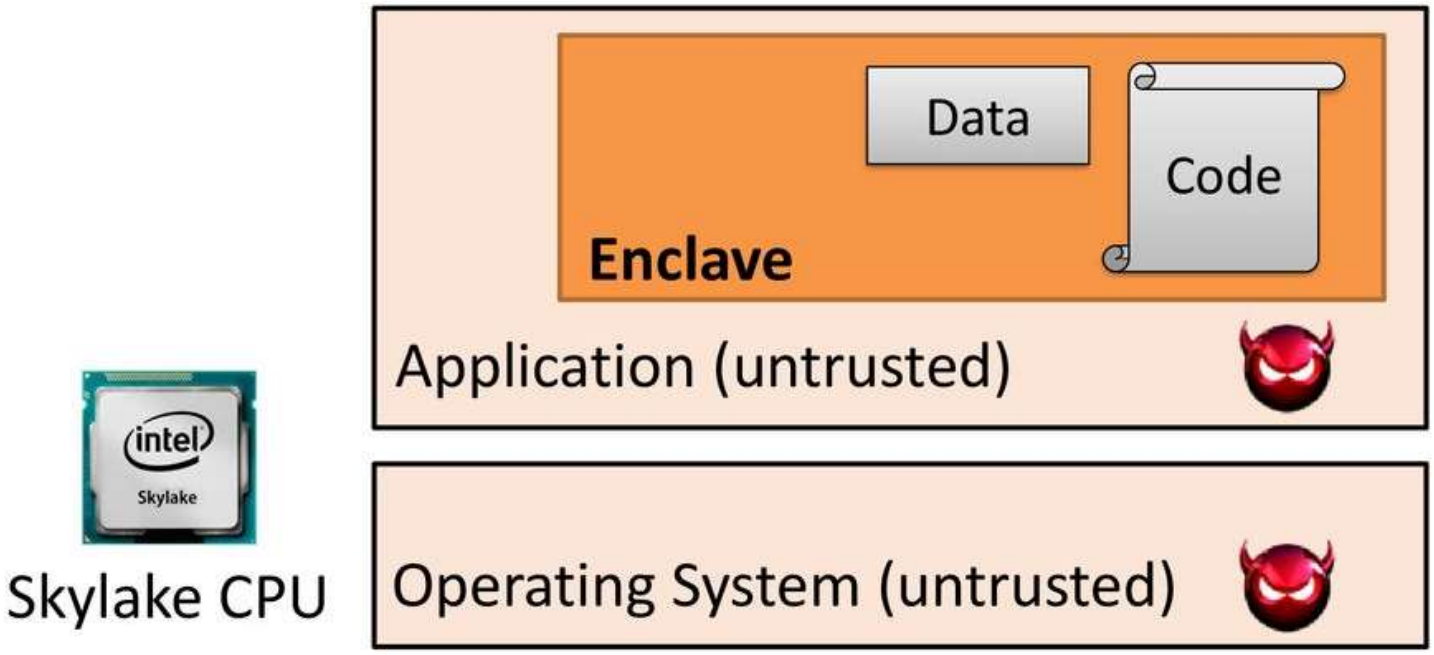


Hardware assured security

1. Secure coprocessors [Yee94]
   - IBM 4758 [SW99]
2. Aegis secure processor [SCG+03]
3. Trusted Platform Module (TPM) [TPM03]
4. Trust Zone [Alv04]
5. AMD SVM [VD06]
6. Intel Trusted Execution Technology (TXT) [FG13]

Intel Software Guard eXtension (SGX)

- An extension of x86 Instruction Set Architecture (ISA)
  - Offers native performance, Compatibility with x86
  - Application keeps its data/code inside the "**enclave**"



Intel Software Guard eXtension (SGX)

Existing Computer Systems

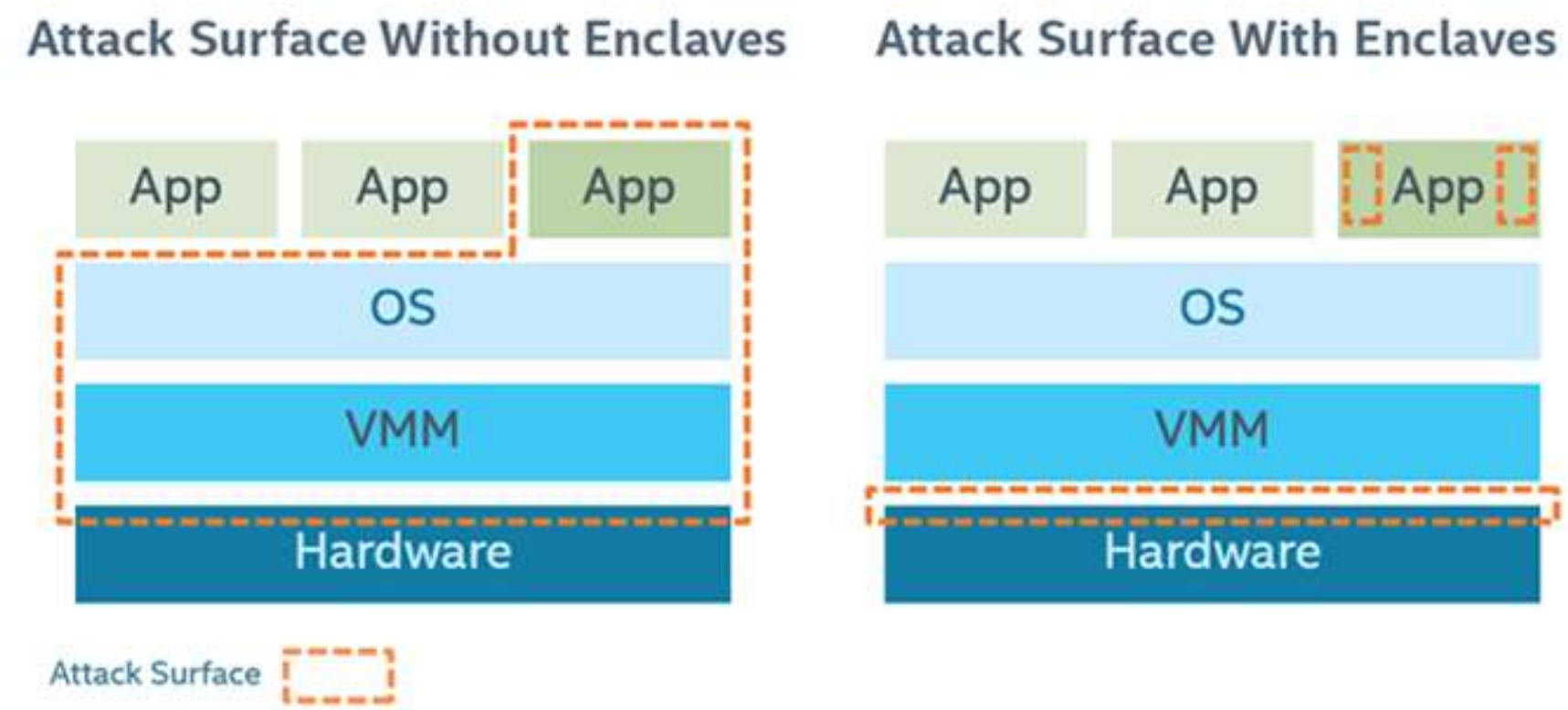
- Apps must trust
  - OS/VMM
  - BIOS, SMM
- Trust relies on software

Computer Systems w/ SGX

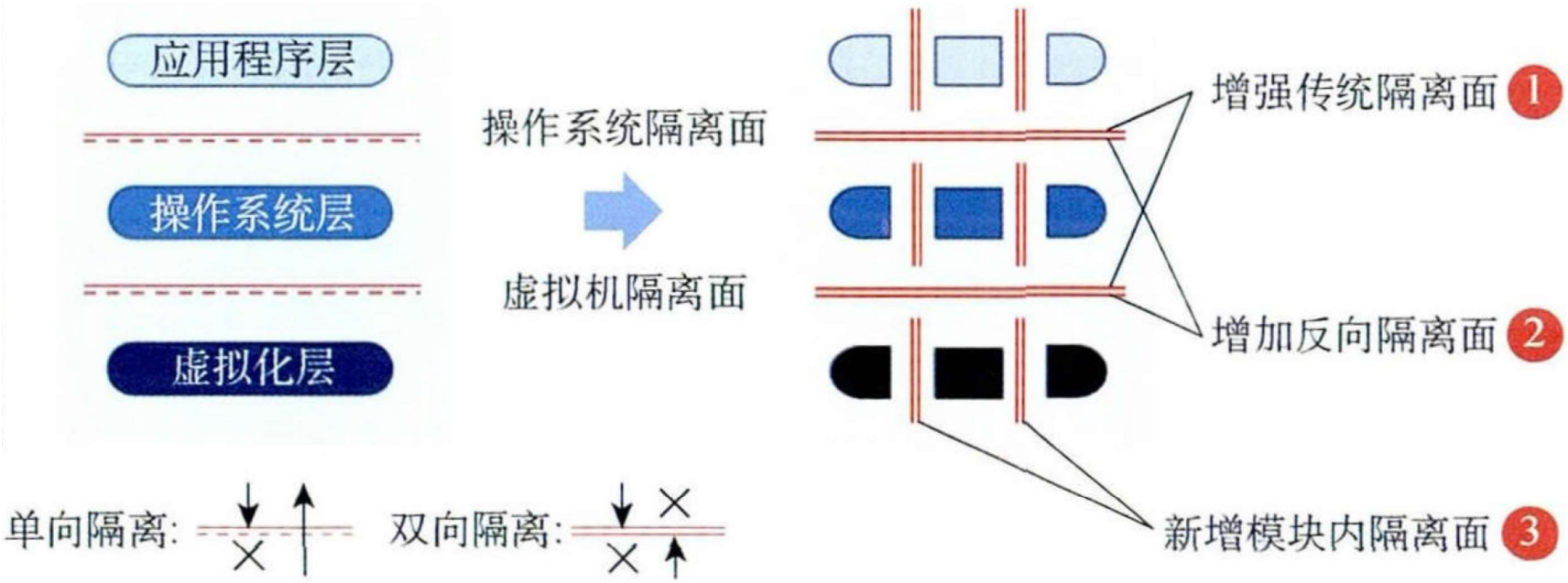
- Apps must trust
  - SGX hardware
- Trust excludes OS/VMM/BIOS/SMM

With SGX, for the first time, apps gain the ability to manage its own secret, without relying on the underlying systems software

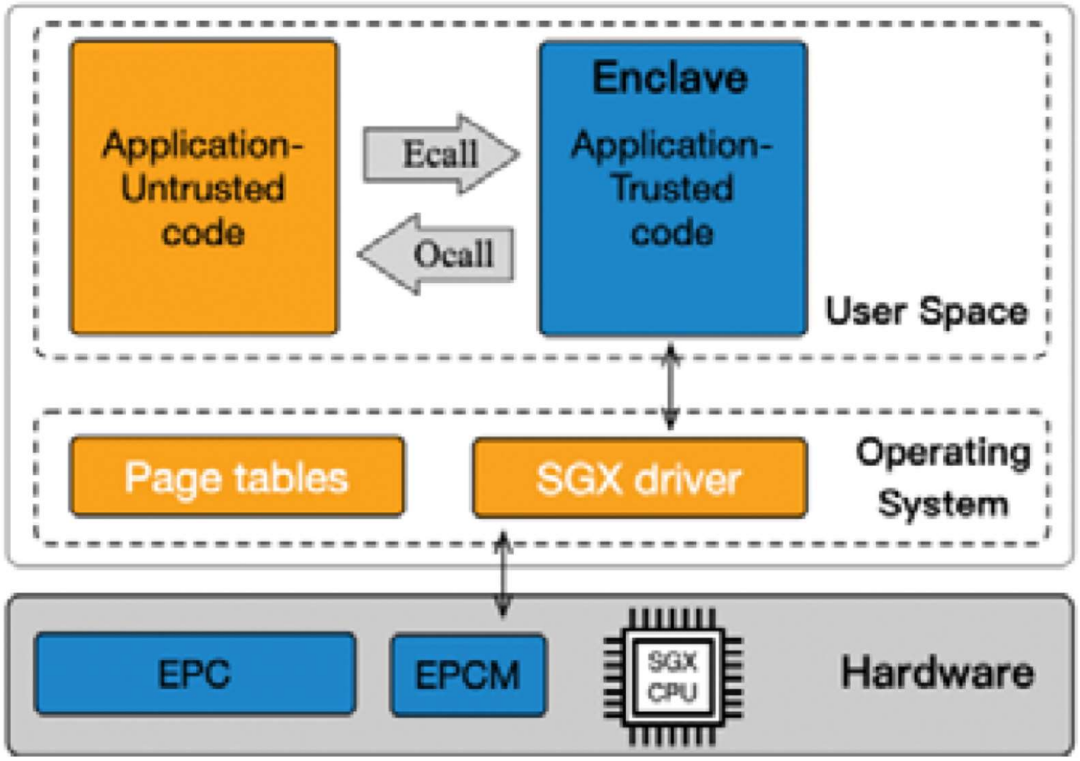
SGX can reduce the attack surface



Disaggregated isolation



Intel SGX architecture



Instructions(User)

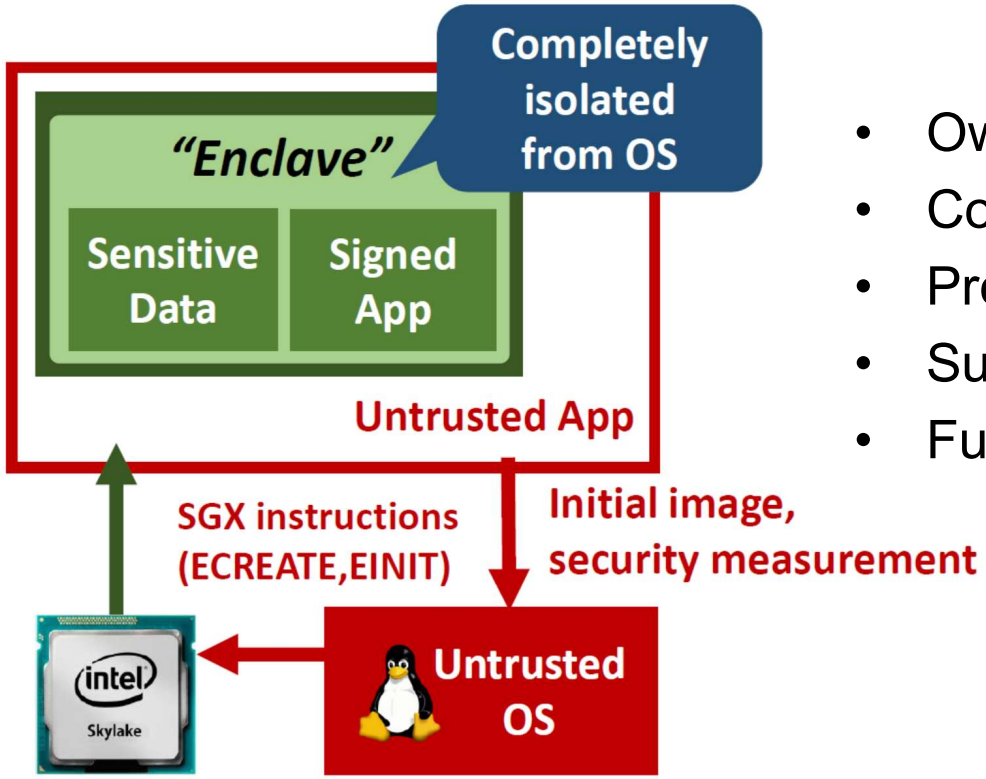
EENTER
EEXIT
ERESUME
EREPORT
EGETKEY

Instructions(System)

ECREATE	ETRACK
EADD	EWB
EEXTEND	ELD
EINIT	EPA
EBLOCK	EREMOVE

Untrusted Trusted

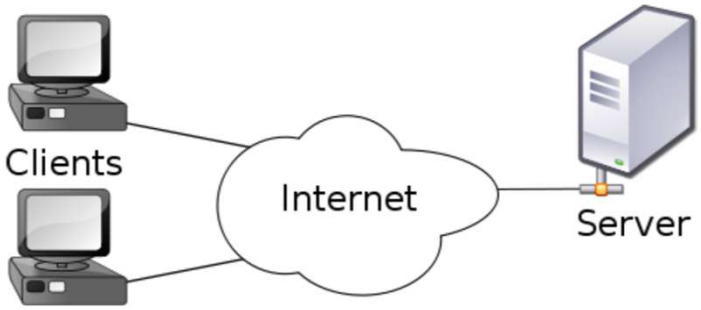
Isolated execution



- Own code & data
- Controlled entry points
- Provides confidentiality & integrity
- Supports multiple threads
- Full access to application memory

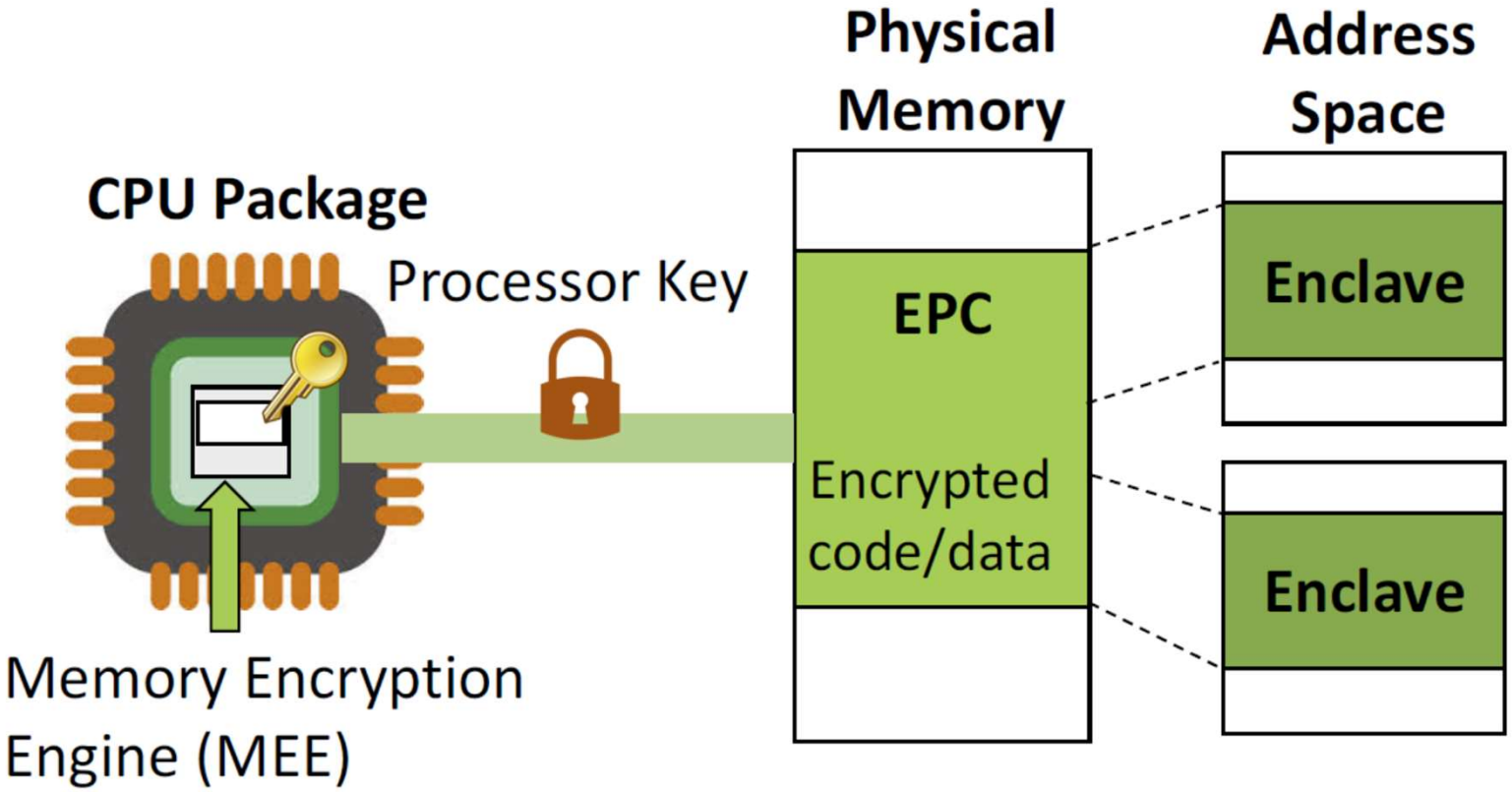
Attestation

- Is my code running on remote machine intact?
- Is code really running inside an SGX enclave?
- Local attestation
  - Prove enclave's identity (= measurement) to another enclave on same CPU
- Remote attestation
  - Prove enclave's identity to remote party
- Once attested, enclave can be trusted with secrets

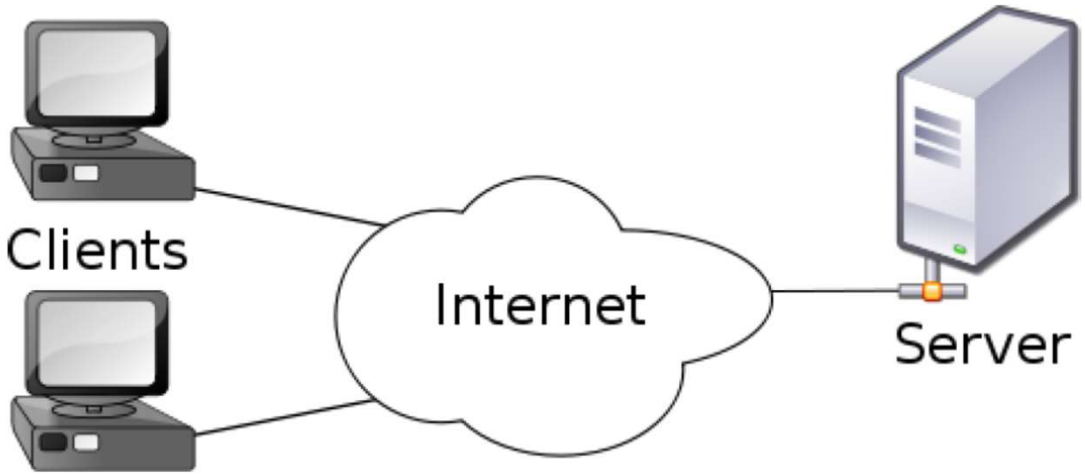


Sealing

- EPC: Enclave Page Cache



Killer APPs



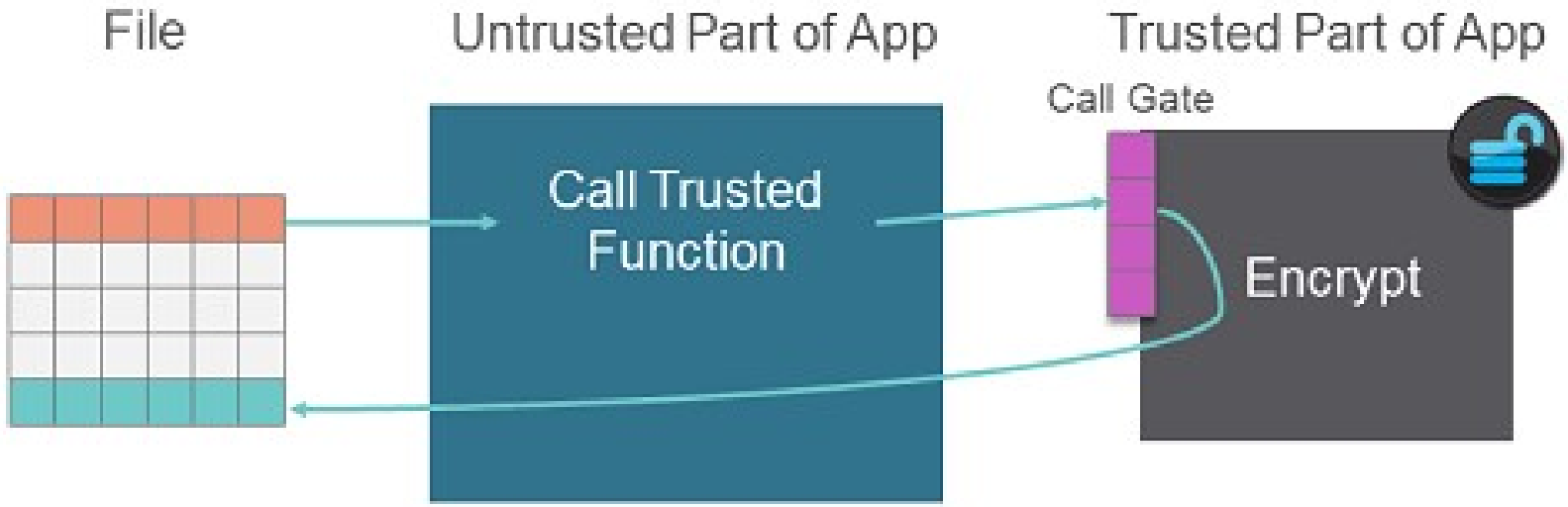
Client: End Users

- **Cloud computing**
  - (Secret-preserving) data analytics
  - Healthcare record processing

Server: Service Providers

- Computer game publishers
- Media streaming providers
- Software vendors (e.g., DRM)

Secure outsourced computation



User/Untrusted storage Untrusted server SGX@Server

Research problems for SGX

- Security issues
  - Cache attacks, side-channel attacks, denial-of-service attacks,…
- Performance bottlenecks
  - Ecall/ocall performance, encrypted page overheads, …
- Function extensions
  - To support container, to support virtual machine migration, …
- Development tools
  - SGX SDK, APP development language, libraries,…