TRUSTED COMPUTING IN "THE CLOUD"

Lect 9

HOMOMORPHIC ENCRYPTION

These slides only introduce HE
¨more on HE in web security course.

Overview

- What is homomorphic encryption
  - Definition
  - HE over Boolean circuits and integers
  - Probabilistic encryption
  - Full Homomorphic Encryption
  - Gentry's result and next
- HE and cloud computing
- Processing on encrypted data without HE

Example

- Consider RSA encryption $Enc(m) = m^e \bmod n$
- Then $Enc(a) \times Enc(b) \bmod n = Enc(a \times b) \bmod n$
- This means we can compute the multiplication (mod n) of two clear text messages by operating on their encrypted versions

Semantic security

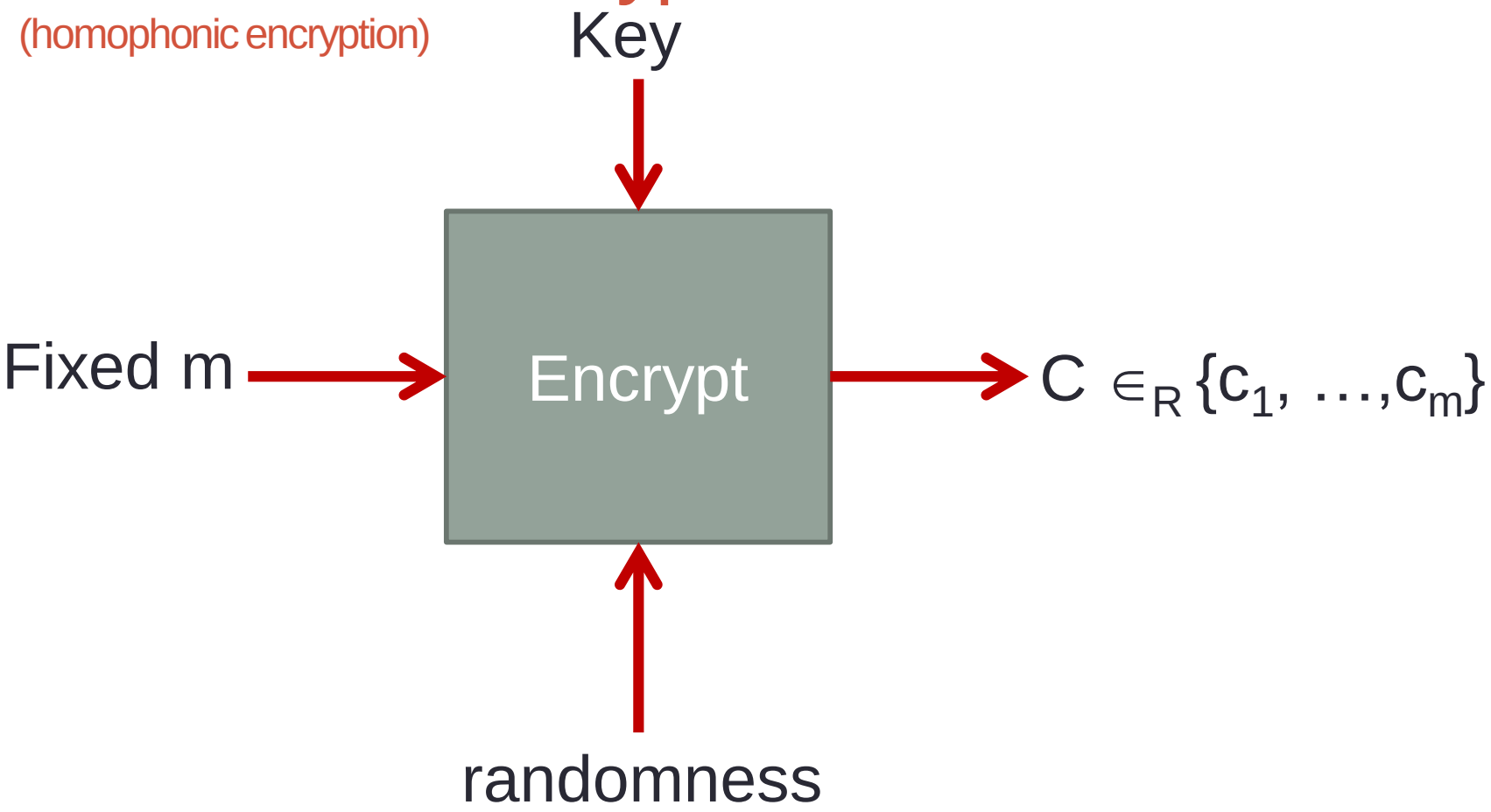
- A cryptosystem is **Semantically Secure (SS)** if any probabilistic, polynomial-time algorithm (PPTA) that is given the ciphertext of a certain message m and the message's length, **cannot determine any partial information on the message with probability non-negligibly higher than all other PPTA**'s that only have access to the message length (i.e. not the ciphertext).

- SS is the computational complexity counter part of Shannon's concept of perfect secrecy. Perfect secrecy means that the ciphertext reveals no information at all about the plaintext, whereas semantic security implies that any information revealed cannot be feasibly extracted.
- Recall discussion on message authentication codes!

- SS was first put forward by Goldwasser and Micali. Goldwasser/Micali later showed that SS is equivalent to **ciphertext indistinguishability**. This security definition works better when proving the security of practical cryptosystems.

Probabilistic encryption

(homophonic encryption)



What is HE?

In general

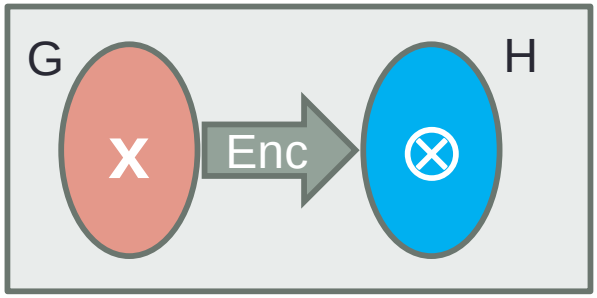
Suppose we have an encryption operation Enc with the properties

- **Inputs** to Enc are elements from a group G with operation x
- **Outputs** from a group H with operation ⊗

  and Enc(a x b) = Enc(a) ⊗ Enc(b)

  (thus Enc is a homomorphism, which explains the naming)

- Then we can perform x operations in G by operating on their encrypted counter parts using ⊗



What is HE? (cont'd)

- Somewhat Homomorphic Encryption
  - Additive
  - Multiplicative

- Fully Homomorphic Encryption

  - On bits: XOR, AND
  - On Integers: add, multiply

(+,x)-Homomorphic Encryption

It will be really nice to have…

1. Plaintext space Z_2 (with ops +,x)
2. Ciphertext space some ring $\mathcal{R}$ (ops +,x)
3. Homomorphic for both + and x, i.e.,
   - Enc(x_1) + Enc(x_2) in $\mathcal{R}$ = Enc(x_1+ x_2 mod 2)
   - Enc(x_1) x Enc(x_2) in $\mathcal{R}$ = Enc(x_1 x x_2 mod 2)

- Then we can compute any function on the encryptions
  - Since every binary function is a polynomial
    (We won't get exactly this, but it gives an idea)

Public-key Encryption - revisited

- We have three procedures: KeyGen, Enc, Dec
  1. (sk,pk) ← KeyGen(random)
     Generate random public/secret key-pair
  2. c ← $Enc_{pk}(m)$
     Encrypt a message with the public key
  3. m ← $Dec_{sk}(c)$
     Decrypt a ciphertext with the secret key

e.g., RSA: $c \leftarrow m^e \bmod N$, $m \leftarrow c^d \bmod N$

- (N,e) public key, d secret key

Fully Homomorphic Encryption

Input: Encrypted x, Program P → Encrypted P(x)
Is there a function Eval that fulfills the below?

Definition of HE scheme: (KeyGen, Enc, Dec, **Eval**)

(regular public/private-key encryption)

"it works"

- **Correctness of Eval:** For every input x, program P
  - If **c = Enc(PK, x)** and **c′ = Eval (PK, c, P),**

    then **Dec (SK, c′) = P(x).** **(so c' was the encrypted P(x))**

"properties"

- **Compactness:** Length of c′ independent of size of P
- **Security = Semantic Security**

Early History (1978-start)

First Defined: “Privacy homomorphism” [RAD’78]

“Can we search in encrypted data?”

[Rivest-Adleman-Dertouzos 78]

Early History (1978-2009)

Additively Homomorphic

[GM'82,CF'85,AD'97,Pai'99,Reg'05,DJ'05…]

Example

- Goldwasser-Micali
- Paillier's Cryptosystem

Goldwasser-Micali'82

Public key: N, y non-square mod N

Secret key: factorization of N

Enc(0): r^2 mod N,
Enc(1): y * r^2 mod N
r = random (unit) mod N

Enc(a)=y^a * r^2 mod N

Enc(a) x Enc(b) = Enc(a xor b)

(Additively) homomorphic over Z_2

Why does GM work?

- Role of randomness in the encryption

  What happens if r is fixed in $\text{Enc}(a)=y^a * r^2 \bmod N$?

- We need some additional basic understanding of so-called quadratic residues mod N using number theory

Why does GM work

- N = p x q, p and q large primes (like in RSA)
- Nonzero a in $\mathbb{Z}_N^*$ non-square mod N?
  - Means there is no b in $\mathbb{Z}_N^*$ such that $b^2 = a \bmod N$

All elements coprime N

- If we know the factorization on N it is easy to determine if a is square or not by computing the Jacobi symbol using

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

where $\left(\frac{a}{p}\right)$ is the Legendre symbol defined for **odd primes**

$$\left(\frac{a}{p}\right) = \begin{cases} 0, if\ p\ divides\ a \\ 1, if\ a\ \in Q_p \\ -1, if\ a\ \in \overline{Q_p} \end{cases}$$

Set of squares mod p

Set of non-squares mod p

$\mathbb{Z}_N^*$

With $\mathbb{Z}_N^*$ we denote the set of elements in $\mathbb{Z}$ that are relatively prime to N .

- This means these elements have a multiplicative inverse mod N, i. e. ,

If a in $\mathbb{Z}_N^*$ then there is a b $\in \mathbb{Z}_N^*$ such that a b = 1 mod N

We call these elements also the unit of $\mathbb{Z}_N$

Squares mod p, e.g. Mod 7

a	1	2	3	4	5	6
a^2 mod 7	1	4	2	2	4	1
Legendre(a; 7)	1	1	-1	1	-1	-1

Legendre(a;p) = $\left(\frac{a}{p}\right)$

Squares (mod 7)= Q_7: 1,2,4

Non-squares (mod 7) = $\overline{Q_7}$: 3,5,6

Mod 15

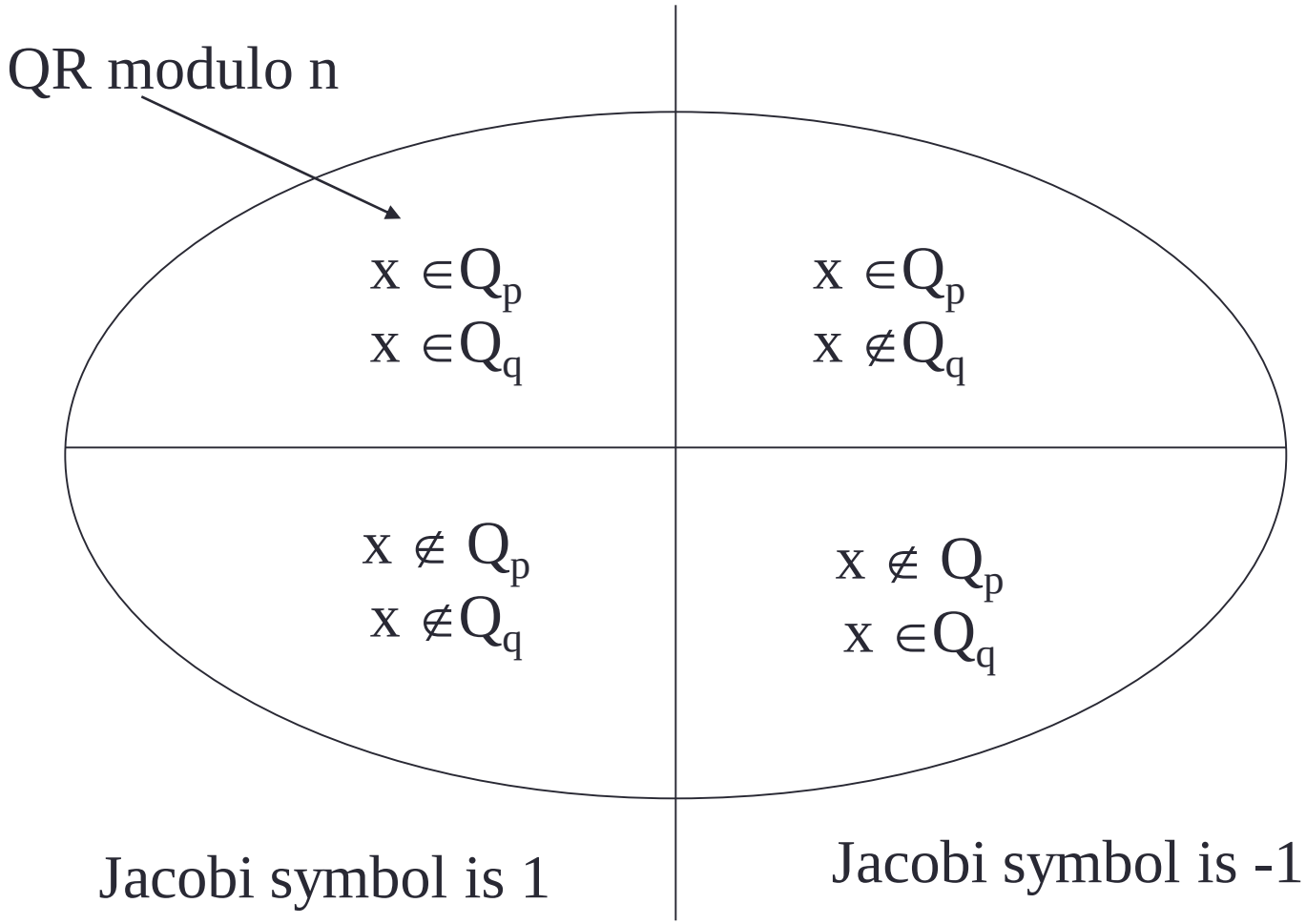
a	1	2	3	4	5	6	7	8	9	10	11	12	13	14
a^2 mod 15	1	4	9	1	10	6	4	4	6	10	1	9	4	1
J(a;15)	1	-1	0	1	0	0	-1	-1	0	0	-1	0	-1	-1
$\mathbb{Z}_{15}^*$	x	x		x			x	x			x		x	x

Squares (mod 15): 1,4,6,9,10

Non-squares (mod 15): 2,3,5,7,8,11,12,13,14

Note: In the above we also listed elements that do not belong to $\mathbb{Z}_N^*$ (in fact 3, 5, 6, 9, 10, and 12) as they have a common divisor with N. This shows that things get more complicated here when computing mod N instead of modulo a prime. In the standard definition for squares and non-squares we only consider elements in $\mathbb{Z}_N^*$. As the Jacobi symbol shows we then have only two squares 1 and 4 and 6 non-squares

Integers in Z_n^*



Why does GM work cont'd

Now

- Enc(0): r^2 mod N is a square
- Enc(1): y * r^2 mod N is a non-square

Because

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p}\right)\left(\frac{a}{q}\right)$$

this tells us that the product of square and a non-square gives a non-square!

Computing Square Roots is as hard as Factoring

Given an algorithm A that can compute one square root of a number a modulo n,

One can use A to factor n as follows

- randomly pick x, compute $z = x^2 \bmod n$
- ask A to compute the square root of z, A returns y
- if y=x or y=n–x, then try again, otherwise, compute gcd(x+y,n) which gives us a prime factor of n
- as A has no way to tell which x we've picked, with prob. ½, A returns a square root that allows us to factor n

Early History (1978-2009)

- Additively Homomorphic
  [GM'82,CF'85,AD'97,Pai'99,Reg'05,DJ'05

- Multiplicatively Homomorphic [EIG'85,…]

- Add + **One** Multipl [BGN'05,GH**V**'09]

- **A Negative Result** [Boneh-Lipton'97,DHI'03]
  Any **deterministic** FHE can be broken in sub-exponential (or, quantum poly) time.
  So we need a probabilistic encryption

Paillier – system

- Set n = p×q, p and q are primes
- λ(n) = φ(n)=(p-1)(q-1) – Euler's phi function
- Select $g = 1 + n \in \mathbb{Z}^*_{n^2}$ All elements that are units mod n^2
- Let $\mu = \frac{1}{\varphi(n)} mod\ n$ and $L(u) = \frac{u-1}{n}$

Public key: n, g

Encrypt: $c = E(m, r) = r^n g^m\ mod\ n$

Private key: λ, μ

Decrypt: $m = D(c) = L\left(c^{\lambda(n)}\ mod\ n^2\right)\mu\ mod\ n$

Paillier's Cryptosystem: additive scheme

Now consider

$E(m_1, r_1) = {r_1}^n g^{m_1}$ $E(m_2, r_2) = {r_2}^n g^{m_2}$ mod n^2,
random r's

Then

$$E(m_1, r_1) \times E(m_2, r_2)$$
$$= {r_1}^n g^{m_1} \times {r_2}^n g^{m_2}$$
$$= (r_1 r_2)^n g^{m_1 + m_2}$$
$$= E(m_1 + m_2, r_1 r_2)$$

Thus the product of encryptions of two messages is *an* encryption of the **sum** of the two messages.

Why decryption works - 1

Note: $(1+n)^x = \sum_{k=0}^{x} \binom{n}{k} n^k = 1 + nx + \binom{n}{2} n^2 + \binom{n}{3} n^3 + \ldots.$

Thus

$$(1+n)^x = 1 + nx \ (mod\ n^2)$$

But then with $u = \ (1+n)^x \ (mod\ n^2)$

we have

$$x = \frac{u-1}{n} \ (mod\ n)$$

$L\left((1+n)^x \ mod\ n^2\right) = x \ (mod\ n)$ with $L(u) = \frac{u-1}{n}$

We have computed a logarithm $mod\ n$

Why decryption works - 2

So with

$$L\left((1+n)^x \bmod n^2\right) = x \pmod n$$

and by Carmichael's theorem (compare with Fermat's little theorem)

$$\omega^{n\lambda(n)} \equiv 1 \pmod{n^2} \text{ for all } \omega \in \mathbb{Z}^*_{n^2}$$

All elements coprime n^2

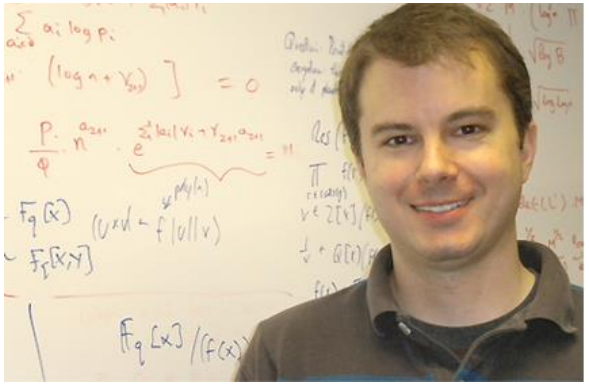
we get that with $c = r^n g^m \bmod n$

$$\begin{aligned} & L\left(c^{\lambda(n)} \bmod n^2\right)\mu \pmod n \\ &= L\left(r^{n\lambda(n)} g^{m\lambda(n)} \bmod n^2\right)\mu \pmod n \\ &= L\left((1+n)^{m\lambda(n)} \bmod n^2\right)\mu \pmod n \\ &= m\lambda(n)\mu \pmod n \\ &= \frac{m\lambda(n)}{\lambda(n)} \pmod n \\ &= m \end{aligned}$$

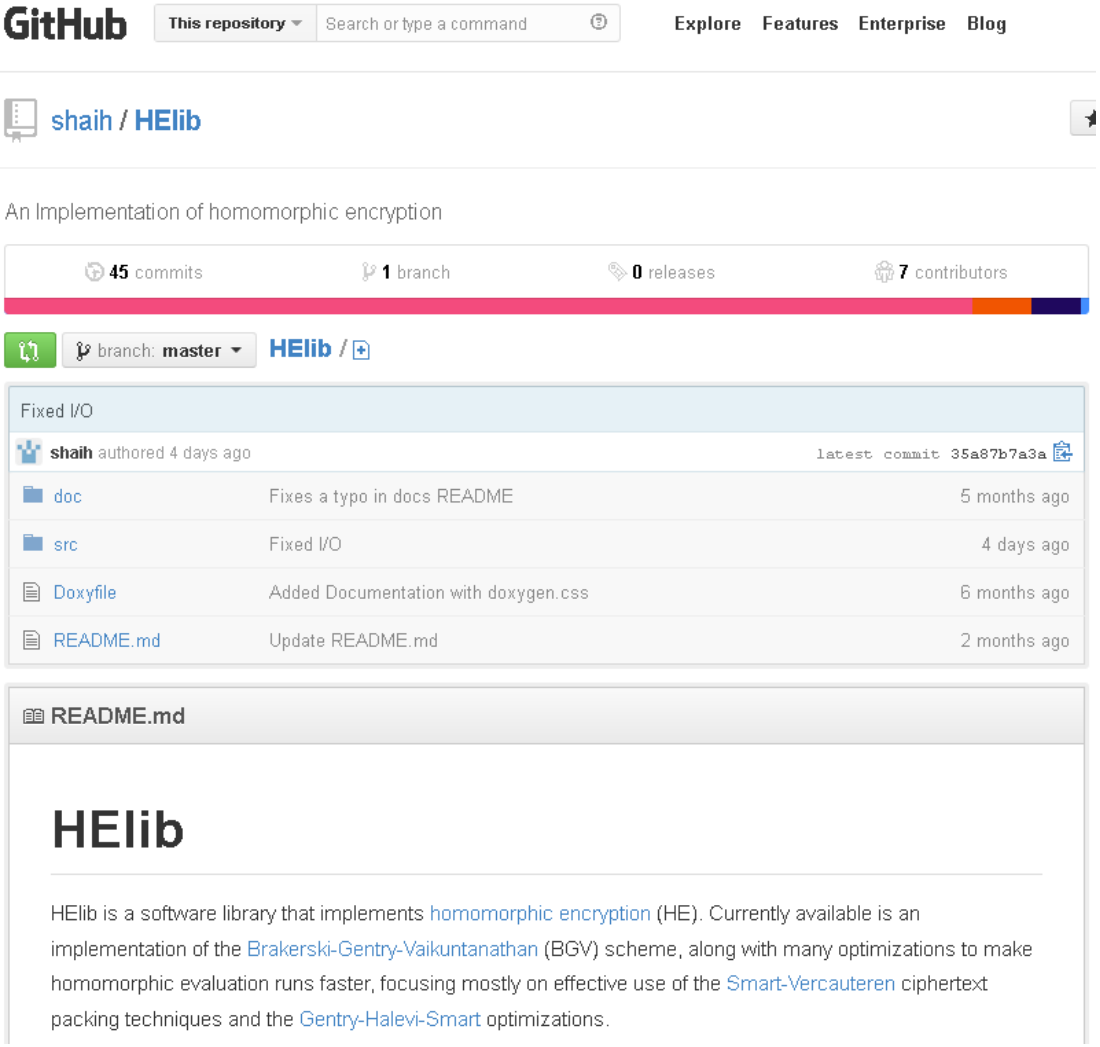
Gentry (2009)

FIRST Fully Homomorphic Encryption!


Forbes.com
U.S. EUROPE ASIA
Home Page for the World's Busines
Home Business Investing Technology Entrepre
CIO Network Games Gear Green Tech Innovation & Science
Breakthroughs
IBM's Blindfolded Calculator
Andy Greenberg, 06.24.09, 06:00 PM EDT
Forbes Magazine dated July 13, 2009
A researcher's algorithm could teach computers a new privacy trick.
© Evan Kafka
Encryption master: Craig Gentry unlocks secrets with



HElib (2013)


GitHub
This repository
Search or type a command
Explore
Features
Enterprise
Blog
shaih / HElib
An Implementation of homomorphic encryption
45 commits
1 branch
0 releases
7 contributors
branch: master
HElib /
Fixed I/O
shaih authored 4 days ago
latest commit 35a87b7a3a
doc
Fixes a typo in docs README
5 months ago
src
Fixed I/O
4 days ago
Doxyfile
Added Documentation with doxygen.css
6 months ago
README.md
Update README.md
2 months ago
README.md
HElib
HElib is a software library that implements homomorphic encryption (HE). Currently available is an implementation of the Brakerski-Gentry-Vaikuntanathan (BGV) scheme, along with many optimizations to make homomorphic evaluation runs faster, focusing mostly on effective use of the Smart-Vercauteren ciphertext packing techniques and the Gentry-Halevi-Smart optimizations.

Since then

Gentry's initial scheme had astronomic complexity

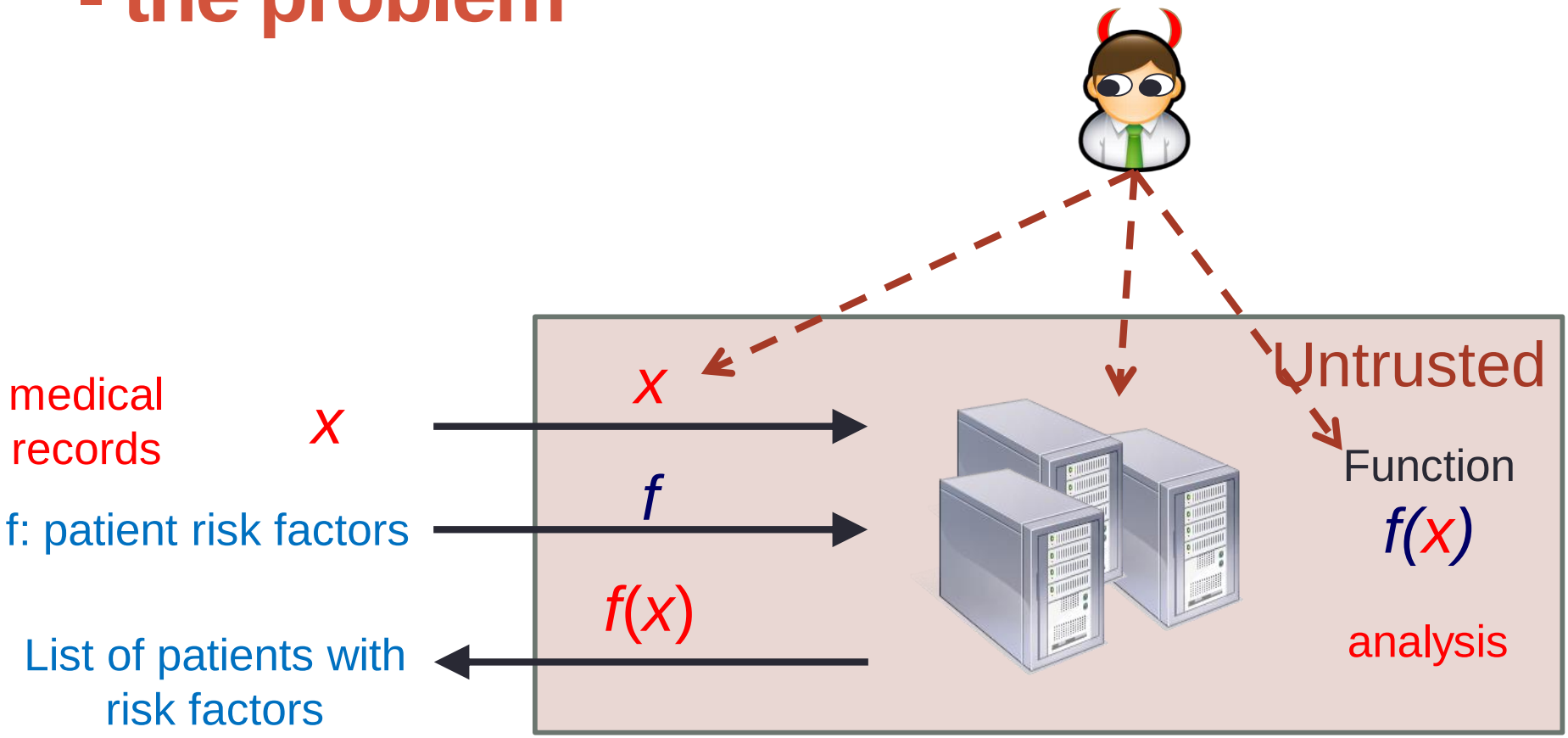
Also the original proof was quite involved and there were many assumptions.

2017 Achieved:

- Much more efficient solutions
- Shorter proofs
- Relaxing of assumptions

But this is still a very active field of cryptographic research, to find better solutions and proofs

Outsourcing Computation - the problem

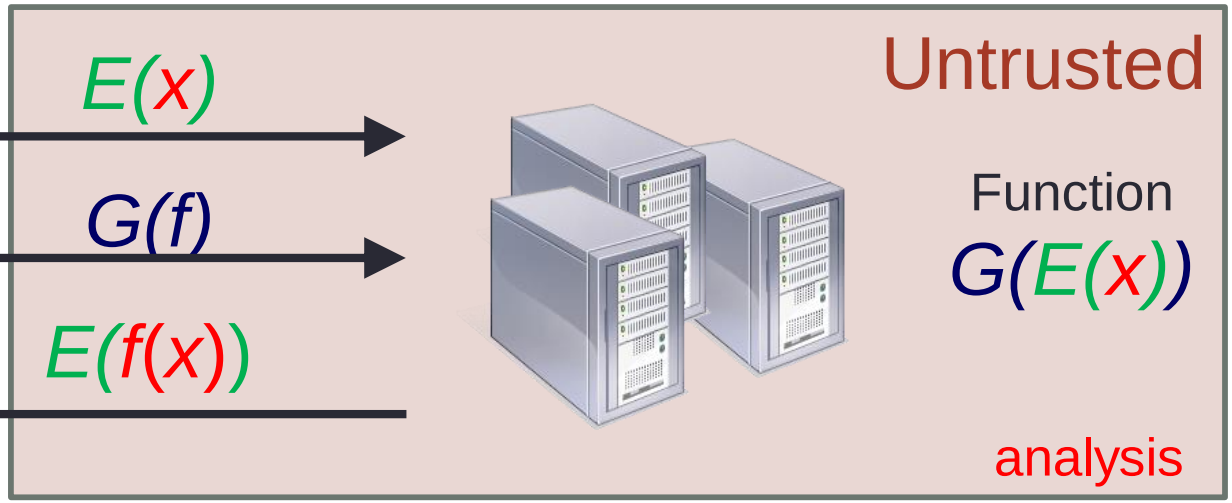


Outsourcing Computation

Encrypt data

Transform f into algorithm G

Decrypt result

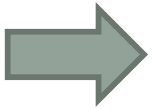


Some other uses of HE

- Secure voting schemes
- Multi-party computations

Alternatives

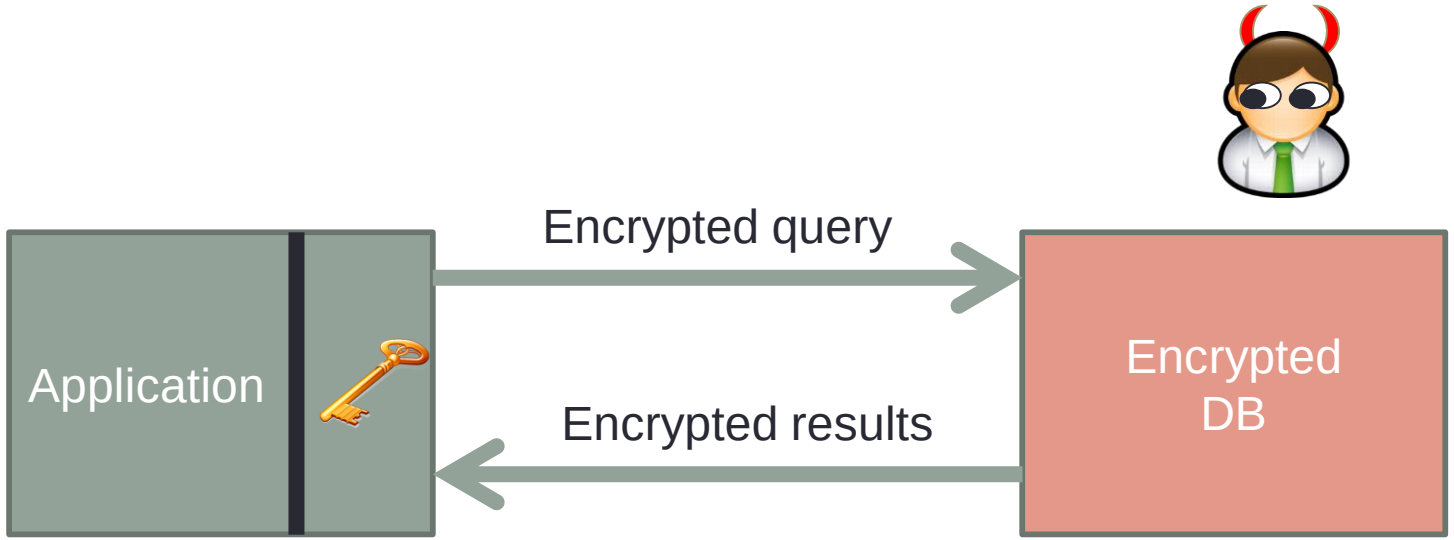
- Can we do processing on encrypted data without relying on HE if we only ask for certain types of processing



YES, we know that already but how complex processing we can do ?

Database operations on encrypted data

- This is indeed possible (http://css.csail.mit.edu/cryptdb/)



Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. CryptDB: Protecting Confidentiality with Encrypted Query Processing. In Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP), Cascais, Portugal, October 2011.

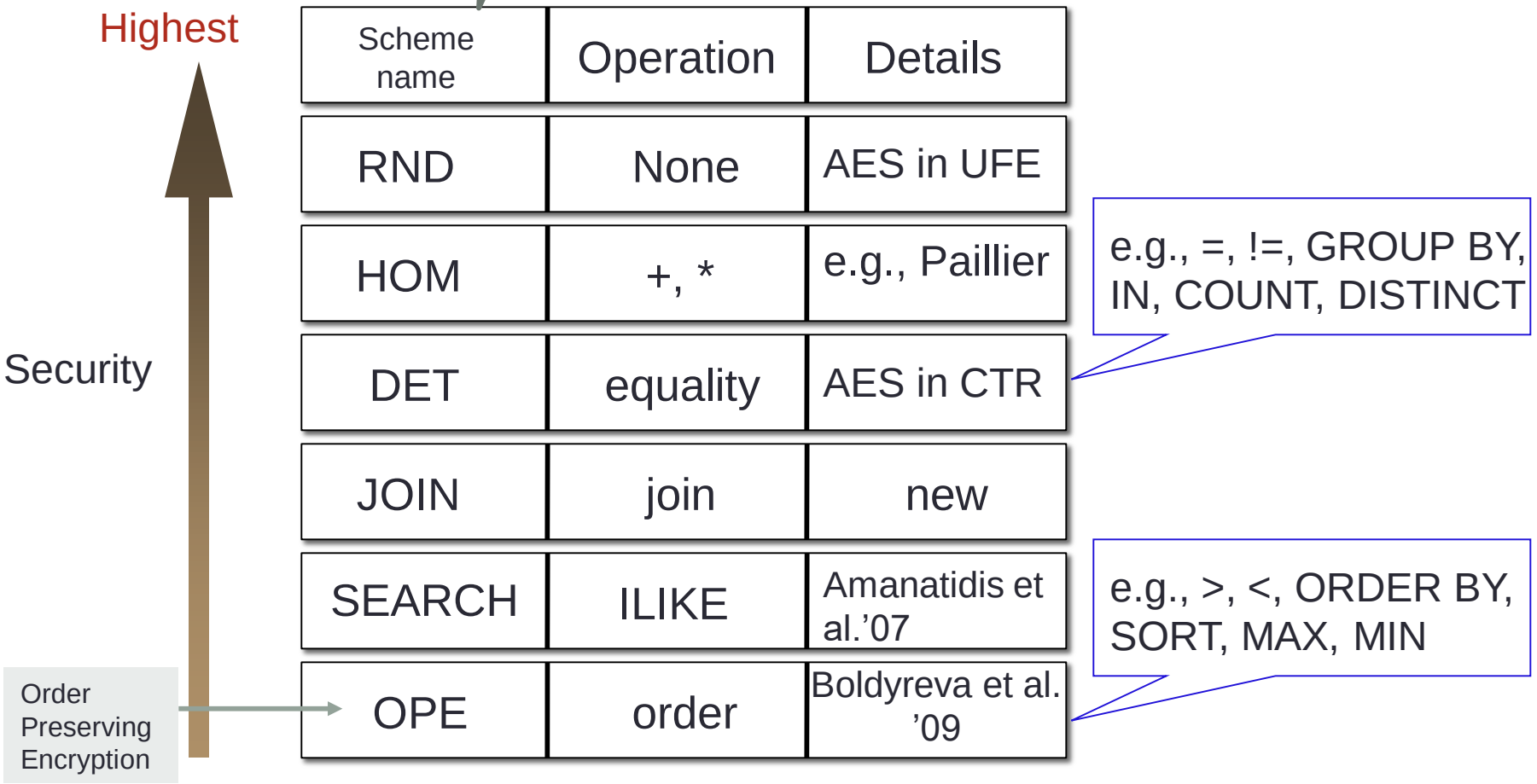
Two techniques

1. SQL-aware encryption strategy
   - *Obs.*: set of SQL operators is limited
   - Different encryption schemes provide different functionality

2. Adjustable query-based encryption
   - Adapt encryption of data based on user queries

Read paper below what these schemes are

SQL-aware encryption

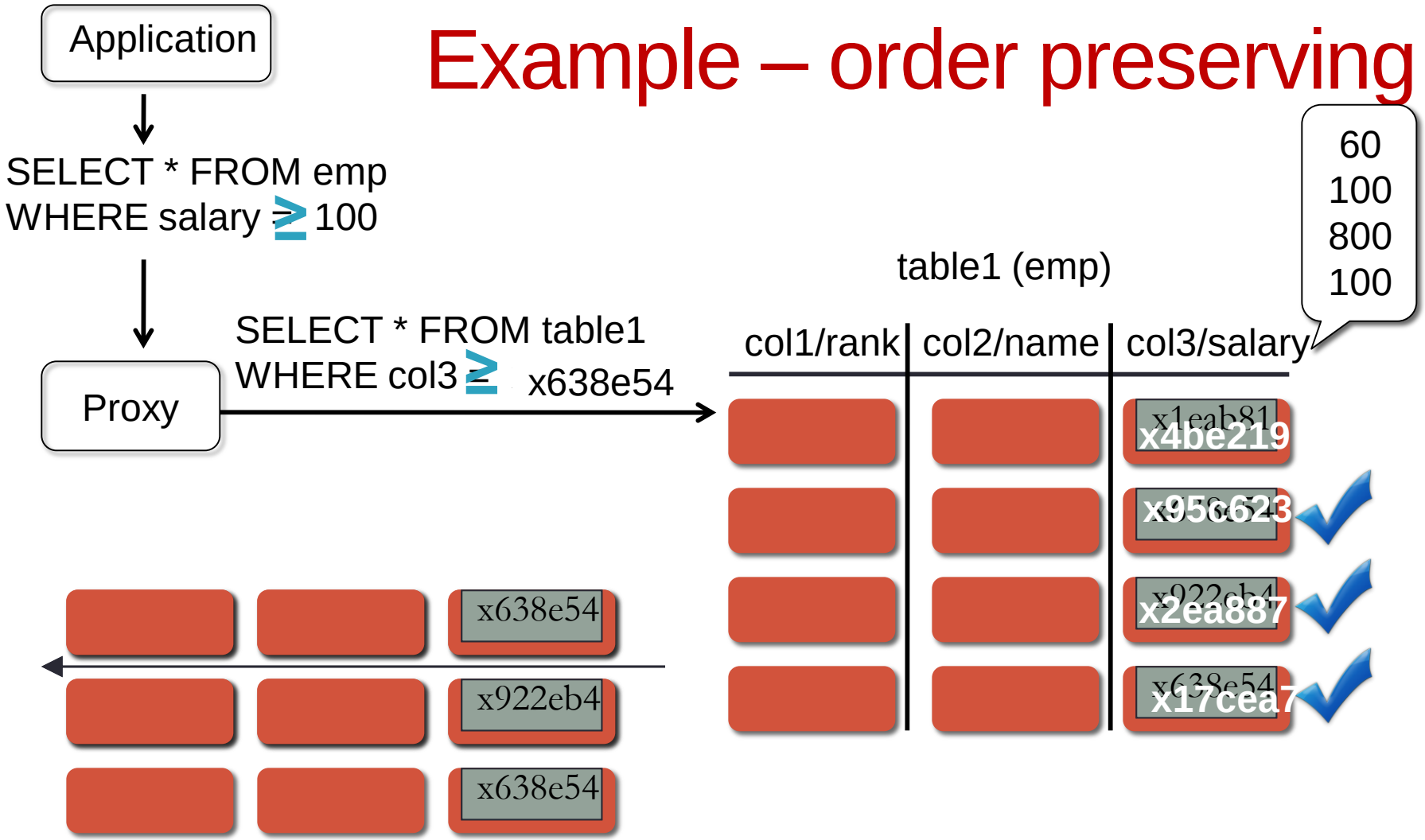


http://people.csail.mit.edu/nickolai/papers/raluca-cryptdb.pdf

Adjustable query-based encryption

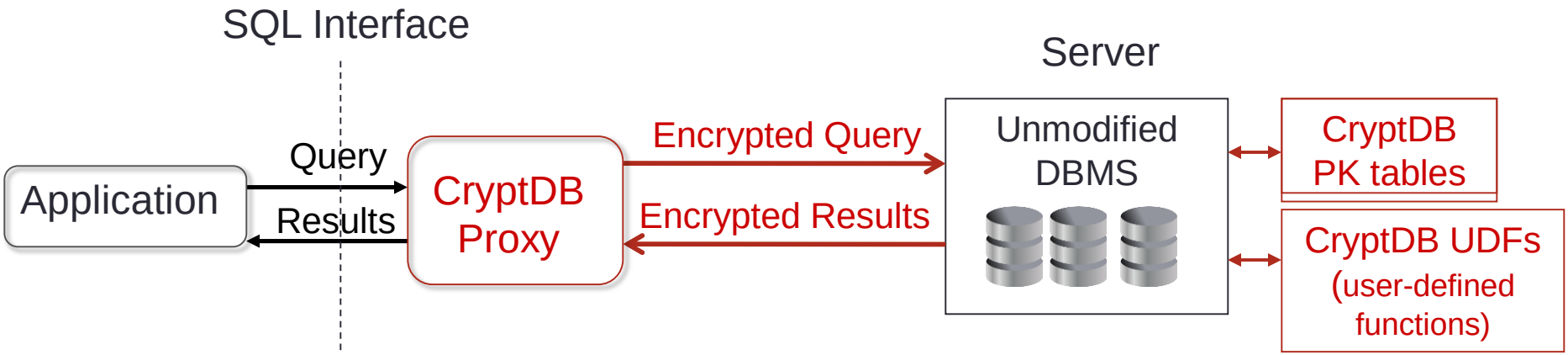
- Start out the database with the most secure encryption scheme
- Adjust encryption dynamically
  - Add more tables to support more functions

Example – order preserving



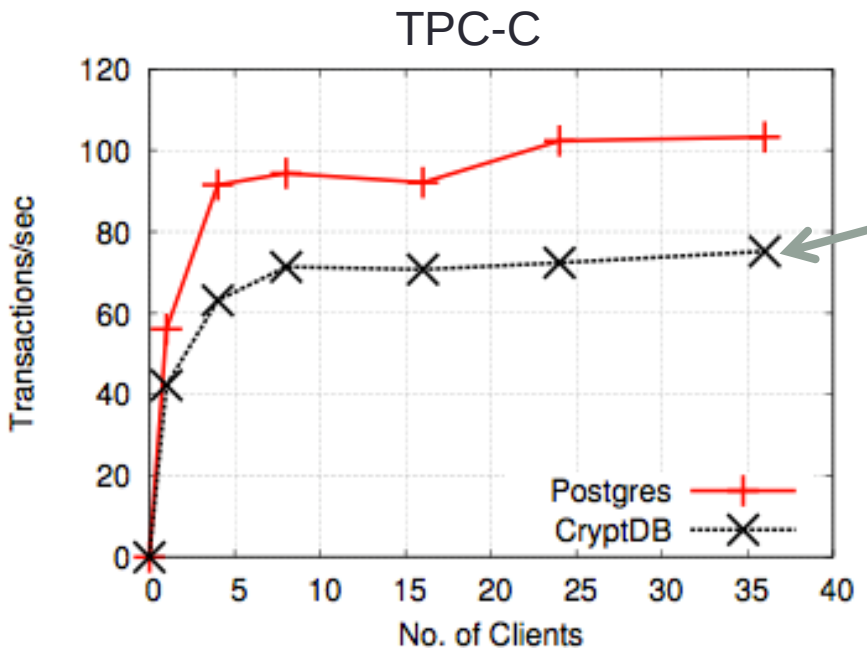
Slide from: CryptDB: Confidentiality for Database Applications with Encrypted Query Processing, Raluca Ada Popa, Catherine Redfield, Nickolai Zeldovich, and Hari Balakrishnan, MIT CSAIL

CryptoDB architecture



- No change to the DBMS
- Portable: from Postgres to MySQL with 86 lines
- One-key: no change to applications
- Multi-user keys: annotations and login/logout

Complexity



Throughput loss 27%

With phpBB application: throughput loss of 13%

Encrypted DBMS is practical

TPC-C is an on-line transaction processing benchmark

Experimental work

- Google:
  - https://code.google.com/p/encrypted-bigquery-client/

- SAP:

  http://www.fkerschbaum.org/sicherheit14.pdf

VM PROTECTION IN VIRTUALIZED COMPUTING INFRASTRUCTURES

Material from a SICS presentation

Contents

- Cloud computing
- Example: Infra cloud project
- Security problems in Cloud
- Secure Launch and Migration

What is Cloud Computing?

- It is a collection of technologies to perform remote and distributed processing

- That the term cloud computing really became widely known was heavily influenced by the Amazon introduction of the Elastic Compute Cloud in 2006.

Or simply: The Network is the Computer (Sun Microsystems, 1997)

Definition of Cloud Computing

Somewhat hard. The following aspects should somehow be involved

- Multi-tenancy - shared/pooled resources
- Massive scalability
- Elasticity – on demand, expand or shrink resources
- Self provisioning of resources

- Moveable resources
- Pay as you go (e.g. Amazon EC2)

Linux | RHEL | SLES | Windows | Windows with SQL Standard | Windows with SQL Web

On-Demand Instance Prices

Region: EU (Ireland)

	Linux/UNIX Usage
Standard On-Demand Instances	
Small (Default)	$0.065 per Hour
Medium	$0.130 per Hour
Large	$0.260 per Hour
Extra Large	$0.520 per Hour
Second Generation Standard On-Demand Instances	
Extra Large	$0.550 per Hour
Double Extra Large	$1.100 per Hour
Micro On-Demand Instances	
Micro	$0.020 per Hour
High-Memory On-Demand Instances	
Extra Large	$0.460 per Hour
Double Extra Large	$0.920 per Hour
Quadruple Extra Large	$1.840 per Hour
High-CPU On-Demand Instances	
Medium	$0.165 per Hour
Extra Large	$0.660 per Hour
Cluster Compute Instances	
Eight Extra Large	$2.700 per Hour
High-Memory Cluster On-Demand Instances	
Eight Extra Large	$3.750 per Hour
Cluster GPU Instances	
Quadruple Extra Large	$2.36 per Hour
High-I/O On-Demand Instances	
Quadruple Extra Large	$3.410 per Hour
High-Storage On-Demand Instances	
Eight Extra Large	$4.900 per Hour

A Massive Concentration of Resources

- But this gives also a massive concentration of risk
  - expected loss from a single breach can be significantly larger
  - concentration of “users” represents a concentration of threats
- “Ultimately, you can outsource responsibility but you can’t outsource accountability.”

But we could also see the cloud in a distributed fashion

Concept of cloud services

- SaaS - Software as a Service
  Application Delivery, as Office 365

- PaaS - Platform as a Service
  Platform delivery, as Google App Engine

- IaaS - Infrastructure as a Service
  Infrastructure Delivery, for example. Amazon EC2

- Multi-tenancy is an architecture in which a single instance of a software application serves multiple customers. Each customer is called a **tenant**.

Risks in the Cloud Services
Confidentiality and accuracy

Customer information is available in large parts of the infrastructure and the provider's(infrastructure owner’s) staff

- Servers
- Storage
- Network
- Backup
- Hypervisor
- Monitoring
- Technical staff
- Administrative staff

Private vs Public Cloud

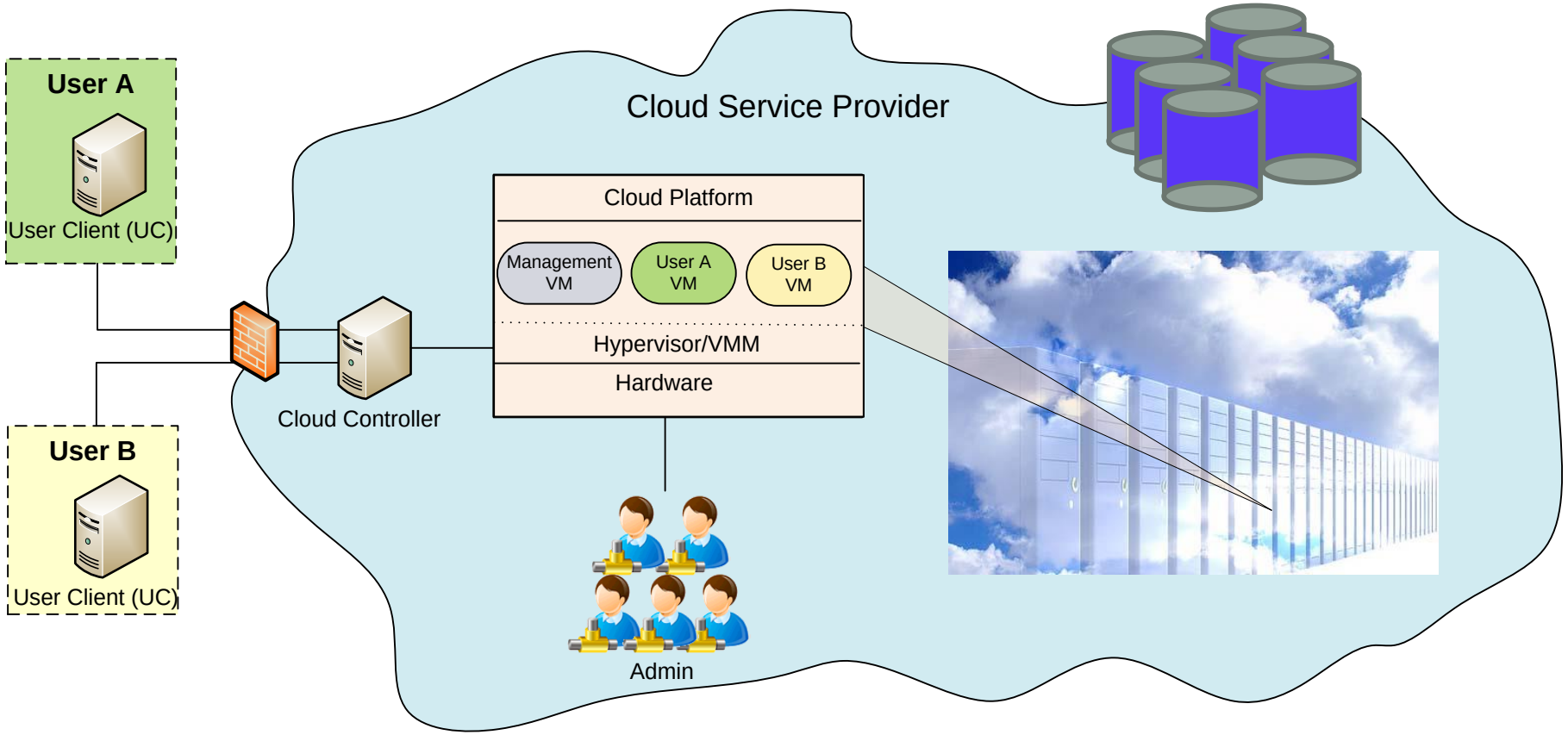
Or in between – hybrid

Private: I own the infrastructure for the cloud solution.

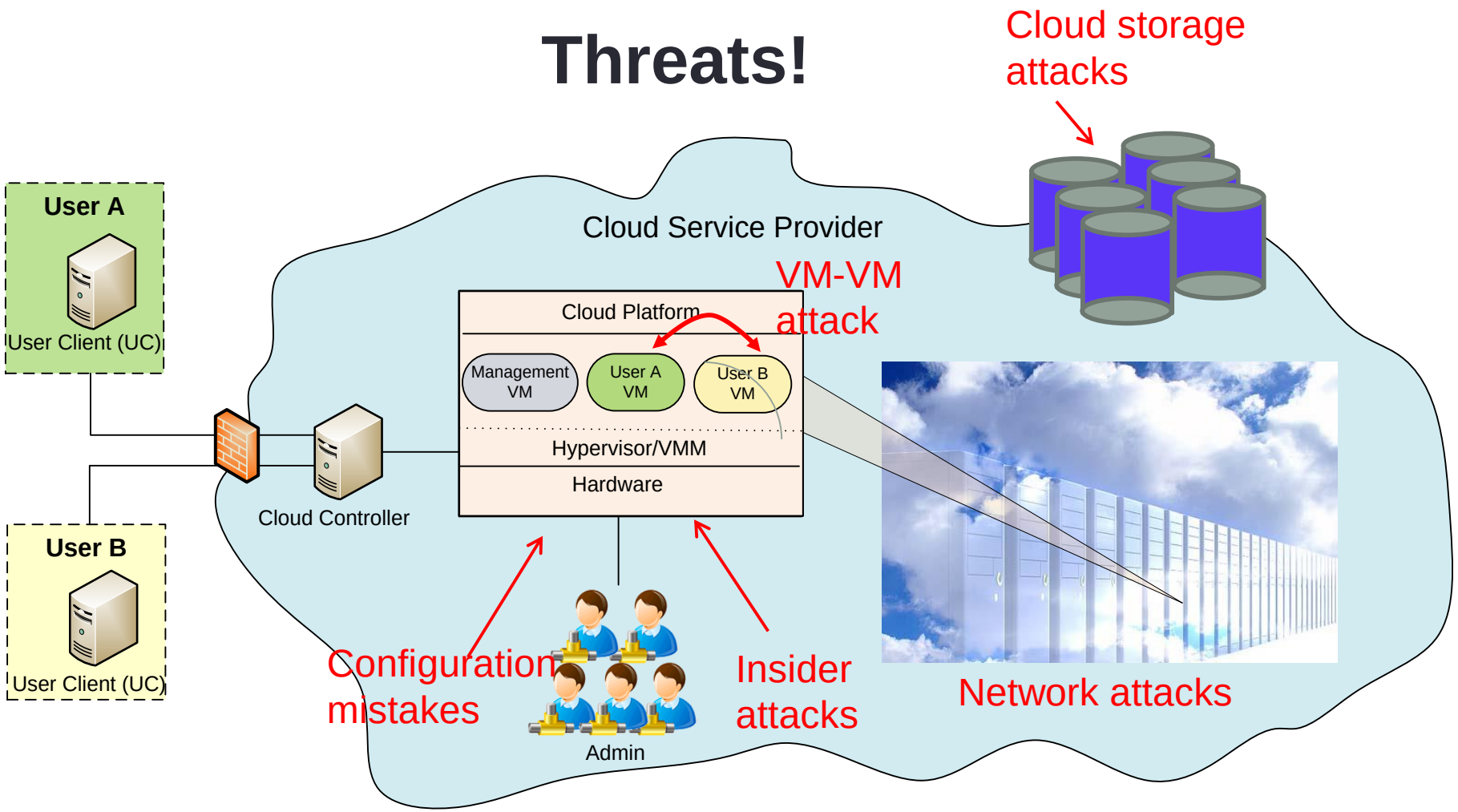
Public: I use public available cloud solution

Technology background (I)

The IaaS model



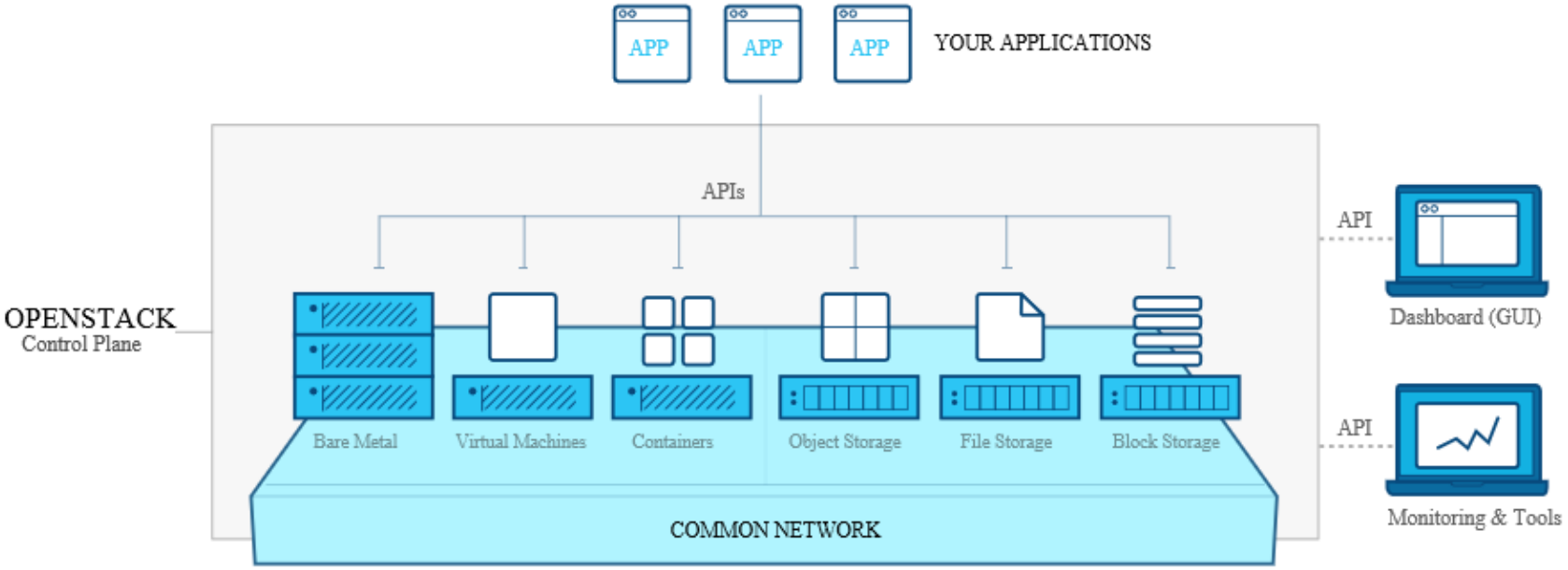
Technology background (II)



Technology background (III)

- Usage of TPM for verification of computing resources:
  - Lock information to specific platform (boot) state(s)
  - Remote verification of platform software state
  - Protection of platform keys
  - Etc.

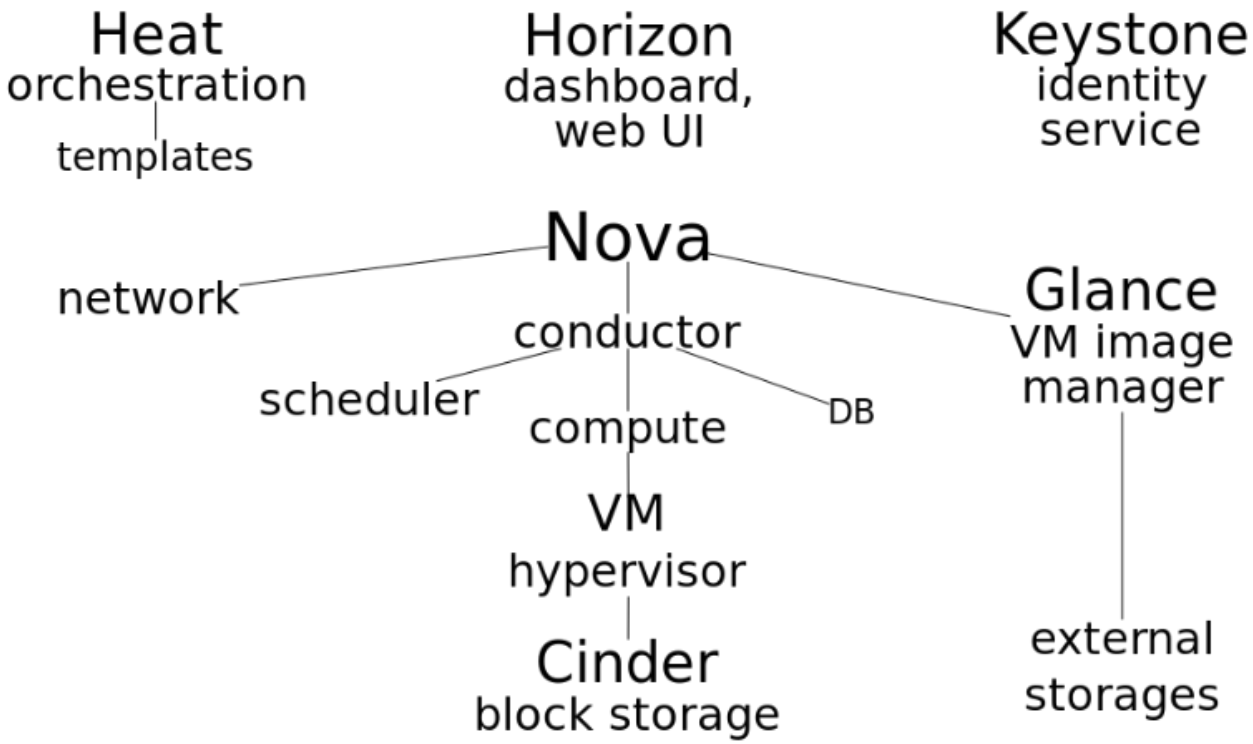
OpenStack



OpenStack is a cloud computing operating system that consists of compute pools, storage solutions, and networking.

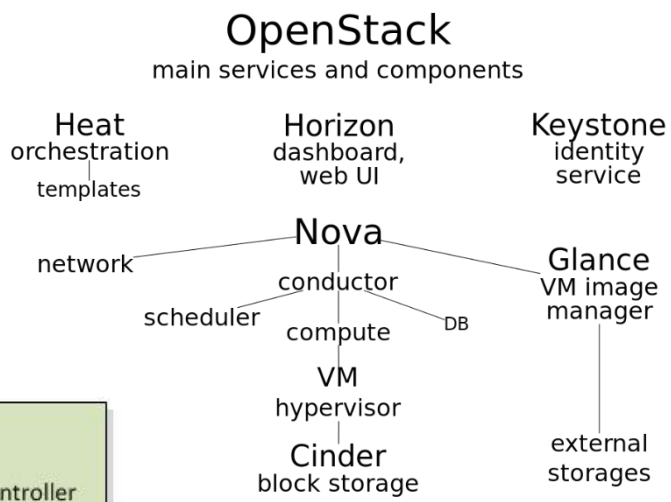
It consists of many component services, each have given names (that not are self-explanatory) such as Nova, Glance, Horizon, Keystone

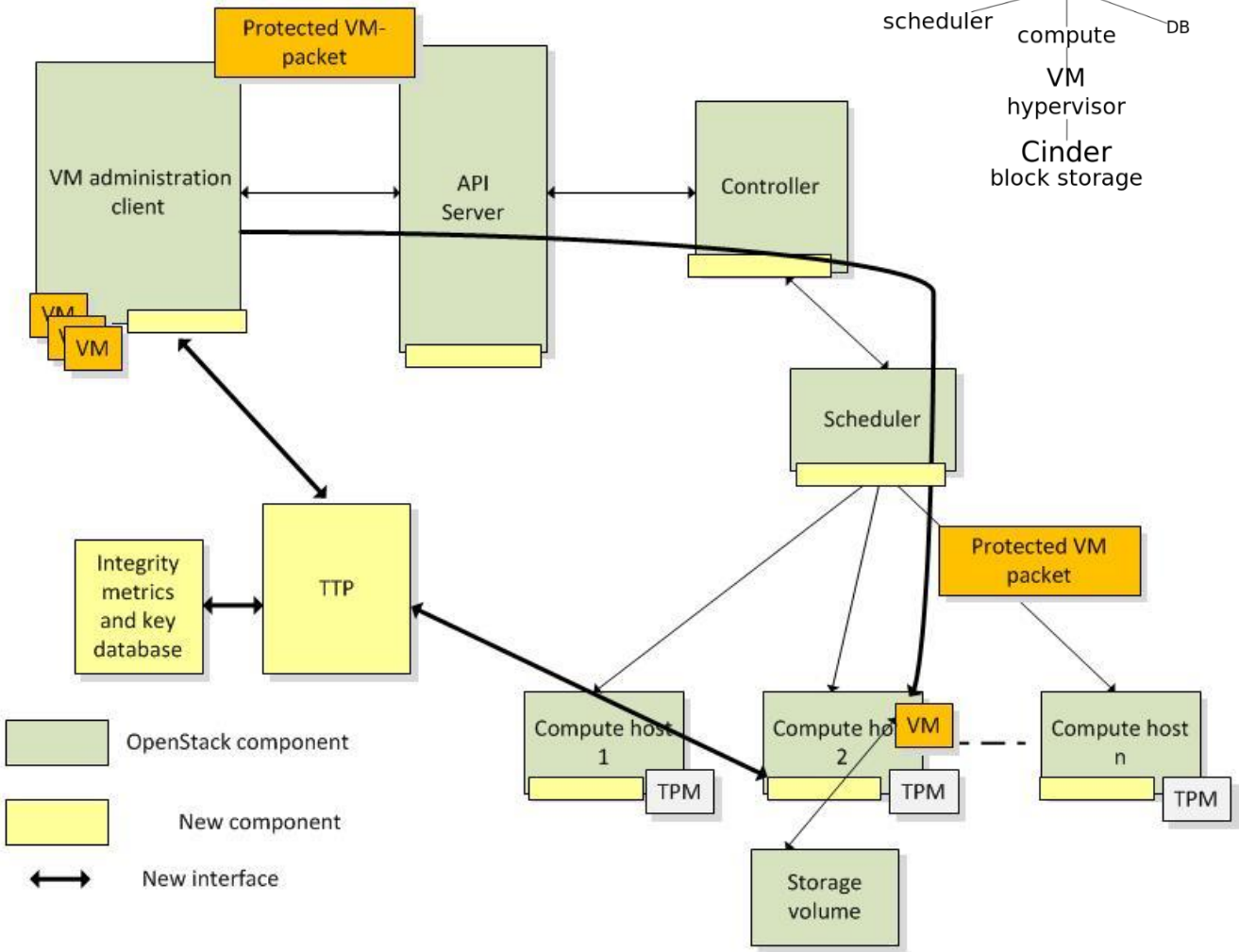
OpenStack – main services



OpenStack View of System

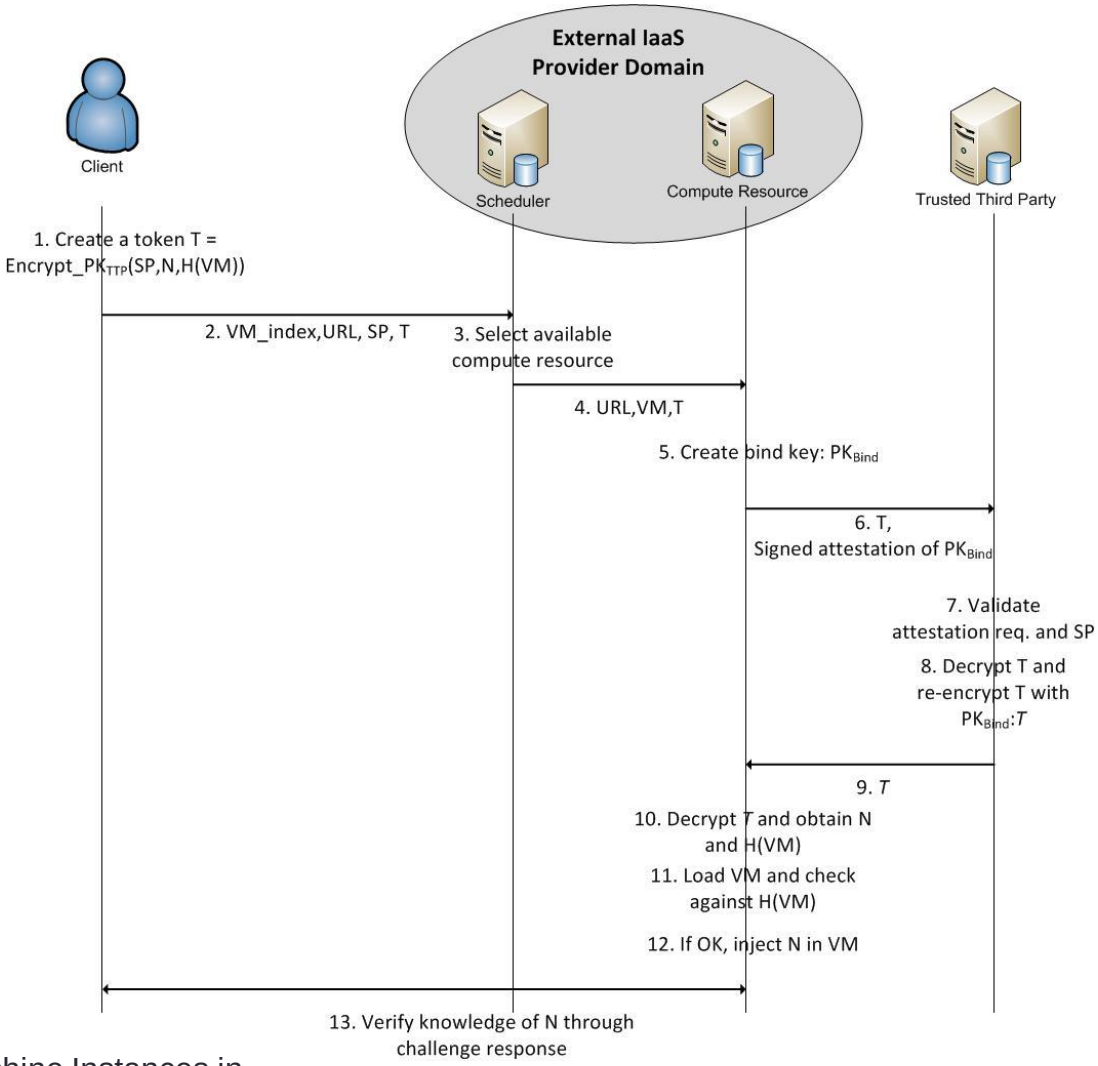
VM = Virtuell Machine
TPM = Trusted Platform Module





Secure VM Launch

(Generic VM, we can also handle encrypted customized VM launch)

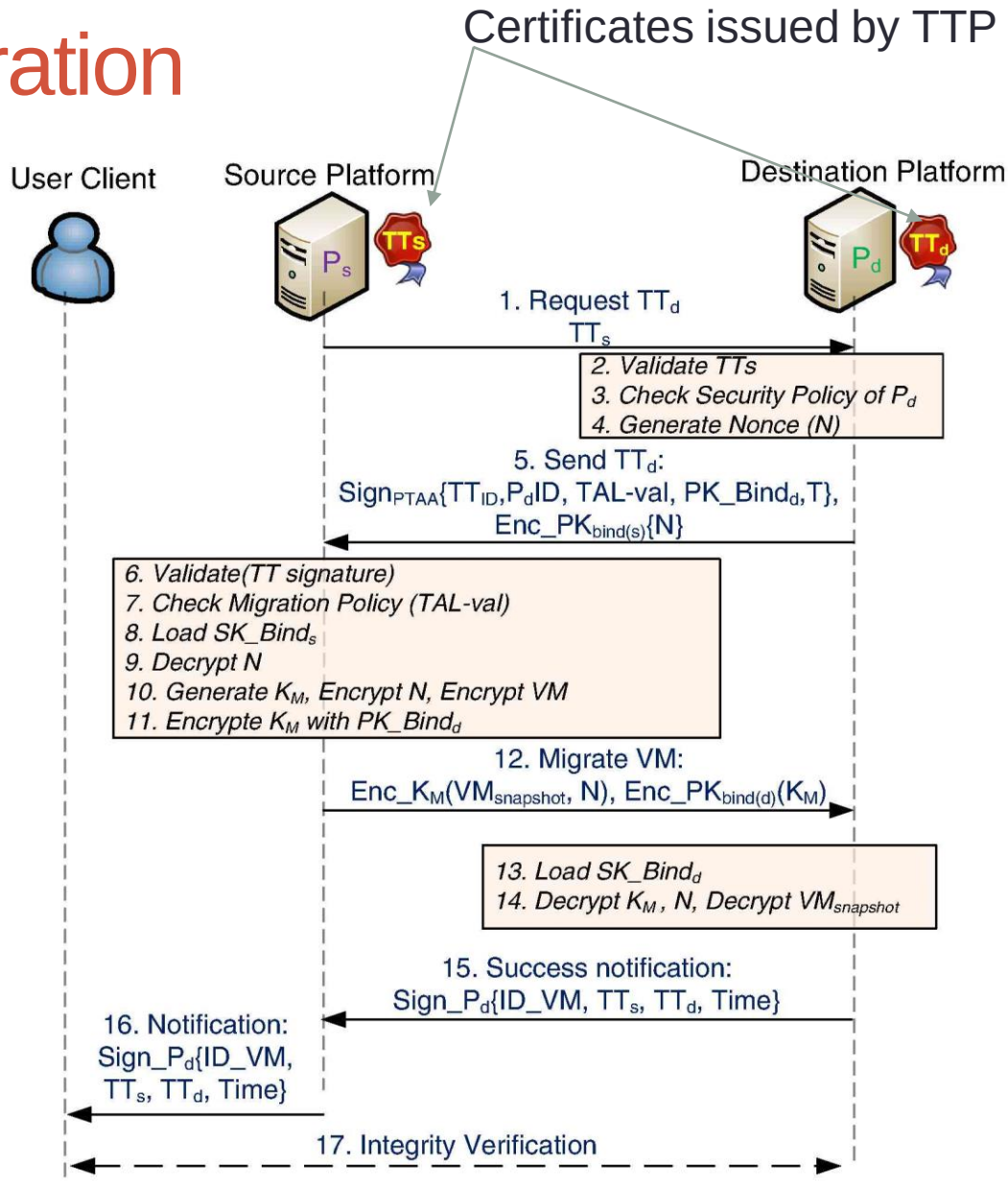


Details: see link

See article: Trusted Launch of Virtual Machine Instances in Public IaaS Environments

Secure VM Migration

Secure VM migration



Details on protocol

- See literature list or follow link

References

- Cloud Security Alliance, https://cloudsecurityalliance.org/education/white-papers-and-educational-material/
- All material from “Security Guidance for Critical Areas of Focus in Cloud Computing v2.1”, http://www.cloudsecurityalliance.org
  - All figures in this presentation taken from this paper
- NIST: Cloud Computing “Security Reference Architecture” (SP 500-299)
- NIST Cloud Model: www.csrc.nist.gov/groups/SNS/cloud-computing/index.html
- Various cloud working groups
  - Open Cloud Computing Interface Working Group, Amazon EC2 API, Sun Open Cloud API, Rackspace API, GoGrid API, DMTF Open Virtualization Format (OVF)
- Cloud Security and Privacy: Mather, Kumaraswamy and Latif, O’Reilly Publishers

IP PROTECTION

Controlling/Protecting Information in Enemy Territory

IP Topics

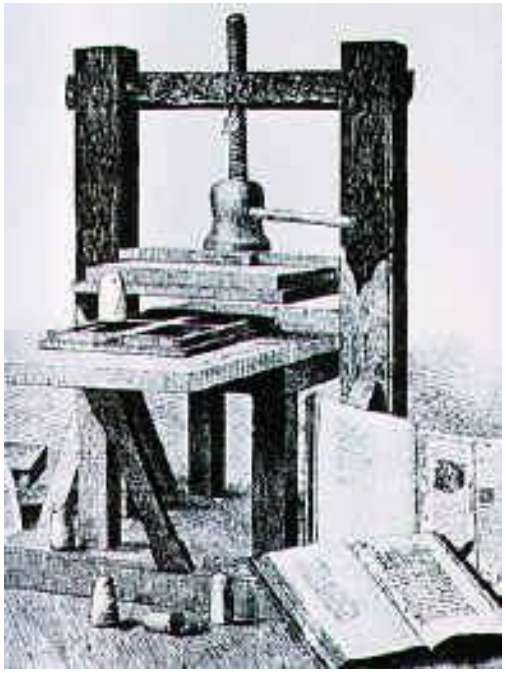
- Is there such thing as “Intellectual Property”
- Why it has become a problem
  - Information Reproduction Technology Through the ages.
- Solutions for Copyright Protection (Law, Technology, Economics).
- Specific Technologies
  - DVD, SCMS, etc.).
  - Apple
  - OMA DRM
  - Windows DRM

IP = Intellectual Property and copyright

- Where does the notion of copyrights come from?
- Why do we have copyrights?
- What does the law say?

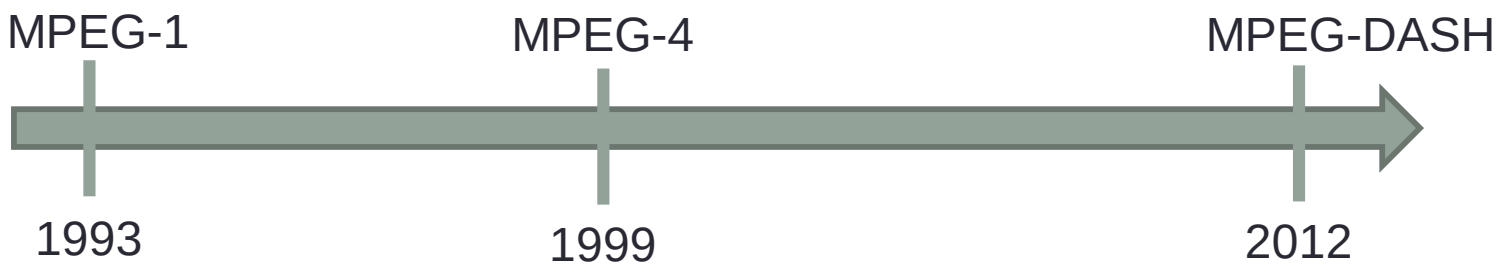
Information Reproduction

- In the “Good Old Days” information (books, music, theatre, etc. . .) was very difficult to reproduce.
- The introduction of the printing press in 1452 (the first mass digital reproduction technology) changed what was practical.



Progess in (lossy) compression

- In 1993 MPEG (Motion Picture Experts Group) standardized MPEG 1. This allowed for the first time nearly CD quality at 1/5 the size of a CD recording.
- By 1996 MPEG 1 layer 3 was becoming a popular format on Internet Newsgroups for posting MP3 files. This allowed CD quality at 10x compression.



There is progress in lossless compression too but for analog data this is not the main stream approach to storing data

The Problem

- The music industry wants to allow the user to play audio from but not copy the content
- In order to play music, the data must be read, at that point it may be digitally copied.
- In the worst case, as the audio goes to the speakers, a user may make an analog recording.

- (and the same for Video)

What Can Possibly Be Done????

Solution 1: the Law

- One solution to the reproduction problem is to use the very search engines that make it easy to find (in case of audio, for example) MP3's to find those distributing them. At that point they can be brought to trial for violating copyright law.

- Big-time offenders can be found more easily, and made example of. The industry did this to some success early with MP3.com and Napster and in recent years getting support of the legislator in many countries.

What About Cryptography?

- The problem of intellectual property protection is not one that can be solved in the usual cryptographic settings.

- In the usual cryptographic case, Alice wants to send Information to Bob without Eve learning it. In the IP protection case, Alice wants to send information to Bob so he can use it but without him being able to copy it.

The latter is much much more difficult

Solution 2: Technology, Take 1

- If we assume that we can give Bob a trusted box that will obey the rules, and Bob cannot open this box, then we can make a solution. This is called "Trusted Perimeter".

- The solution is simple: the box has a public key known to the content providers. In this way, the box and the content provider can communicate securely without Bob learning anything.
- Bob can register all of his boxes with the content provider, and all content he buys will be encrypted for these boxes.

The Problem With the Box

In the previous solution we assumed that the box could be trusted.

- This is impossible in software. Software can be decompiled and reverse engineered. Bob can always learn the private key of his “box”.
- In hardware, it is easier to assume the user cannot read the inner workings of the box, but it is still possible.
- Hardware only solutions are more expensive and less versatile than software solutions.

Solution 2: Technology, Take 2

A separate technique is the Traitor Tracing schemes. This involves using technology to attempt to trace who breaks the rules.

- Watermarking is often invoked in this context. Watermarking is the idea of putting a signal into a digital media file that includes some identifiable information.
- This information could then be used to trace the original purchaser of the media file.

How Does Watermarking Work?

- Bitmap images give a simple example of watermarking: Suppose each pixel has 16 bits of color information associated with it. Suppose the last bit of each pixel is thrown away leaving 15 bits per pixel. The quality is not significantly worse. Now, a digital signature of the file is made. This information is inserted at a rate of one bit per pixel into the file.
- The resulting image will have an imperceptible signal embedded.

General Watermarking Strategies

- In any media file, some bits will be more significant than others. Replacing insignificant bits with digital signatures is a general technique.
- A powerful attack on the above strategies is to randomize the least significant bits in a file.

Problems with Watermarking

- The trouble with Watermarking is that it must be difficult to remove, and yet not negatively affect the quality of the media file.

- So far, no schemes have been presented that are truly practical, which give impossible to remove watermarks and retain high fidelity.

Solution 3: Economics

- Some have suggested that digital media will require a total reworking of business models associated with copyrighted material.
- Subscription models are an attempt at this solution. Consumers never store music, it is delivered to special devices which play the music as it is received.
- This is like cable TV with no VCR's. This solution also relies on the Trusted Perimeter model to a degree.

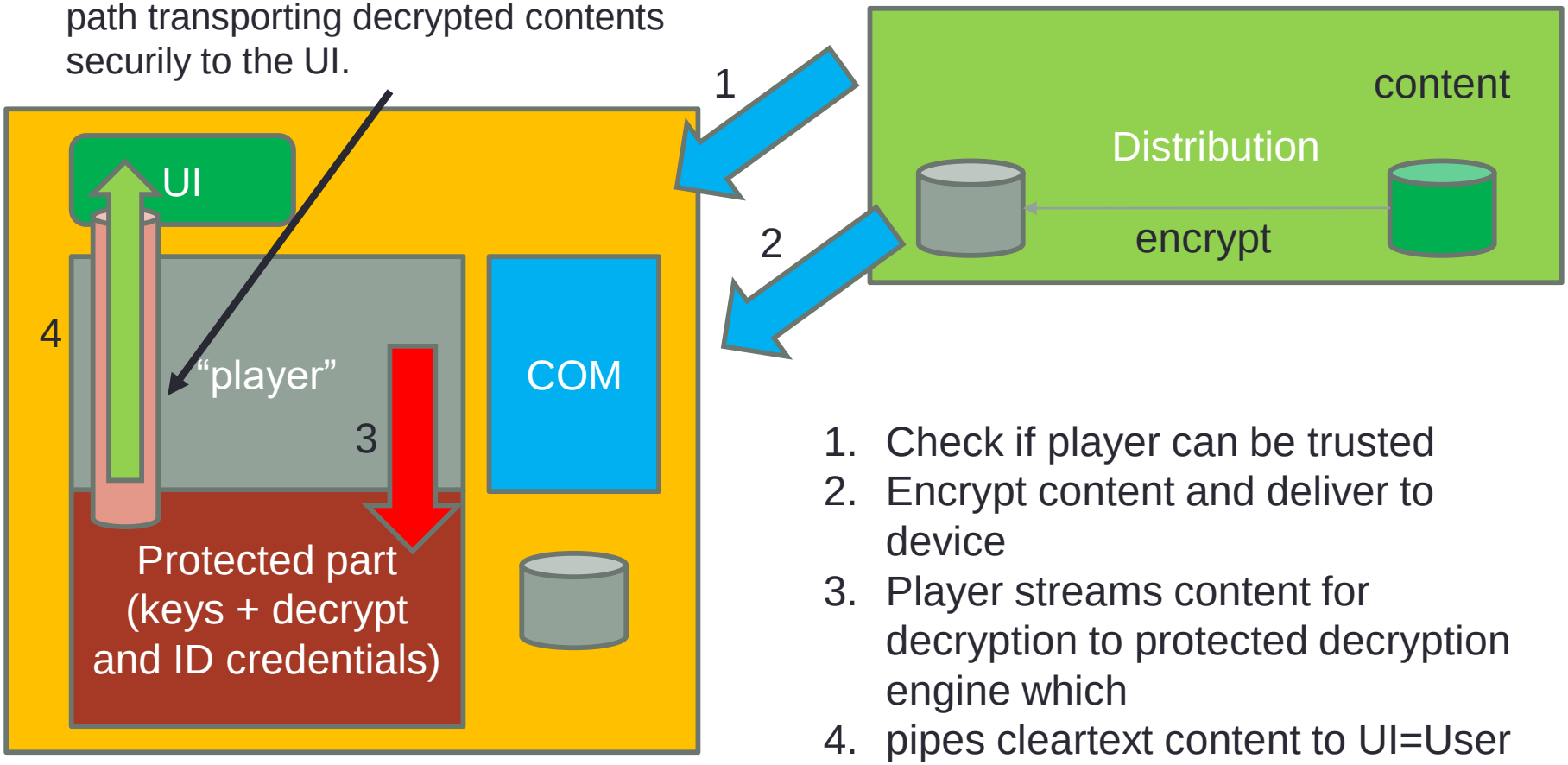
The Street Performer

- Bruce Schneier, suggested a somewhat radical solution: like street performers, content providers would collect money for their next project. When the money is sufficient, they release the content into the public domain.

- This solution by-passes the problem by the copyrights issue by making it irrelevant.

Archetype of a DRM solution

Some devices have a protected media path transporting decrypted contents securily to the UI.



1. Check if player can be trusted
2. Encrypt content and deliver to device
3. Player streams content for decryption to protected decryption engine which
4. pipes cleartext content to UI=User Interface (display, speaker, etc)

Examples of Technology for protection

- SCMS: Serial Copy Management System.
- DVD CSS: Content Scrambling System.
- DIVX Video Discs.
- Macrovision: Analog video copy protection on DVD's.
- SDMI: Secure Digital Music Initiative.
- DVD Audio/SDMI: Watermarking by Verance.
- Trusted Perimeter: Intertrust.

All have been broken

Remember Solution 1? (Law)

- In the US the publishers of the DeCSS code (which allows Windows and Linux computers to decrypt DVD's) were sued in New York and California courts.
  - The DMCA (Digital Millennium Copyright Act) was being invoked to argue that the DeCSS code is illegal.

- At present, in many countries breaking/bypassing the copyright protection is illegal

DRM - today

- DRM technology remains and is being used in mobile devices and so-called protected media path (protected perimeter) implementations

- Mostly used for premium content: e.g. 4K film, Blu-ray (?)

Personal DRM ?

- Private (Your) data protected by a DRM like system
  - Avoid giving the data away
  - But give to others only permission to perform processing on (certain) parts
  - Can it be done ?
  - Are their differences compared to DRM for content industry?
    - Risks?
    - Privacy?

Data ownership –in general

- The IoT use will lead that large amounts of data is produced that directly or indirectly relate to a person.
  - Information systems in cars
  - E-health
  - Facility management systems
- Who owns the data?
  - E.g. can sell Volvo sell the location of your care to gas stations so they can sell you messages when you are near?
  - Regulations: E.g. In China location data is not for user to decide
- Privacy related.
  - EU regulation (accepted 2016 and effective in use 2018).
    - Fines (up to 4% of turnover world wide), reporting within 72 hours, assessment of privacy impact of product/service