Theory and Practice of Homomorphic Encryption

Kristin Lauter

WAM: Uhlenbeck Lectures #3

May 24, 2018

Problem:

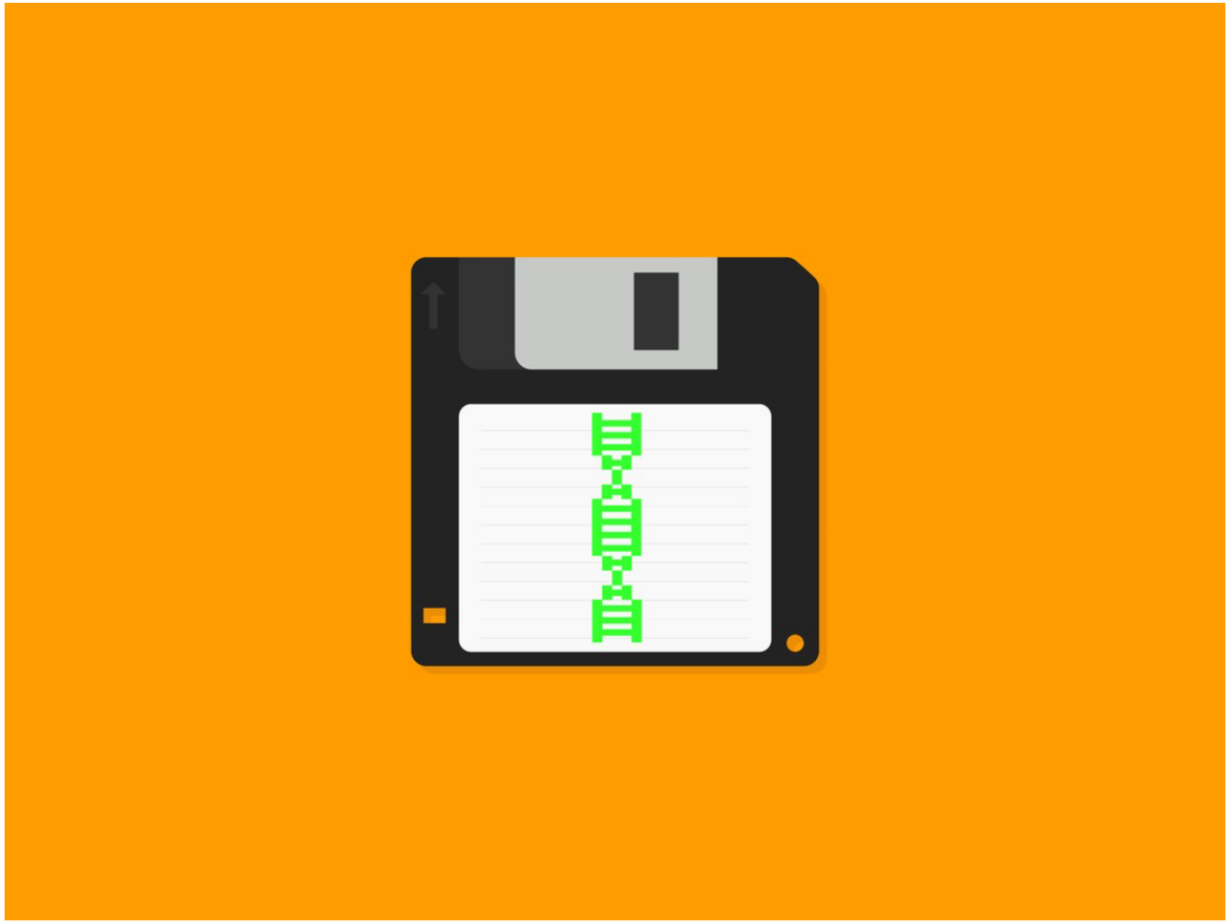
How to provide secure and private cloud storage and computation for consumers and enterprise customers?

Examples:

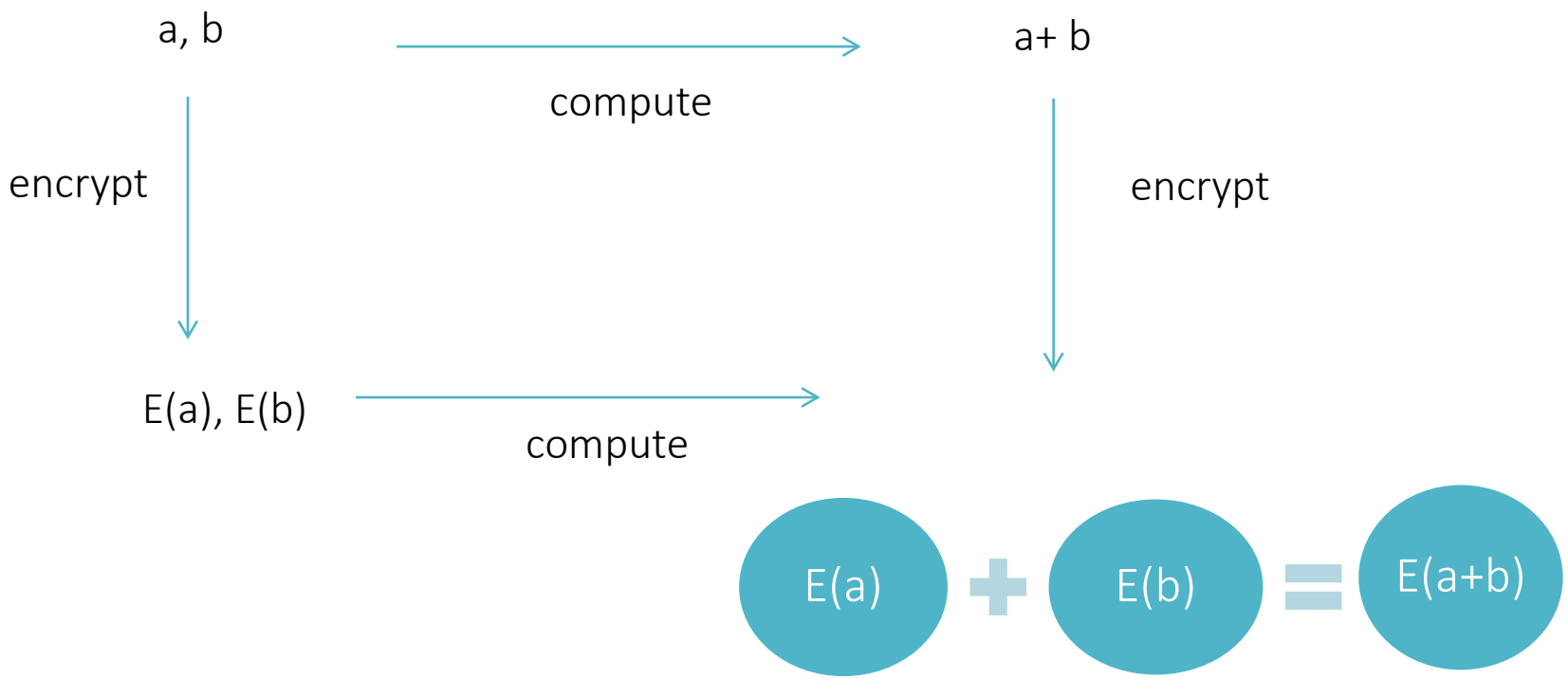
- medical data from hospitals
- financial data from banks
- personal genomic data

MEGAN MOLTENI SCIENCE 08.23.17 07:00 AM

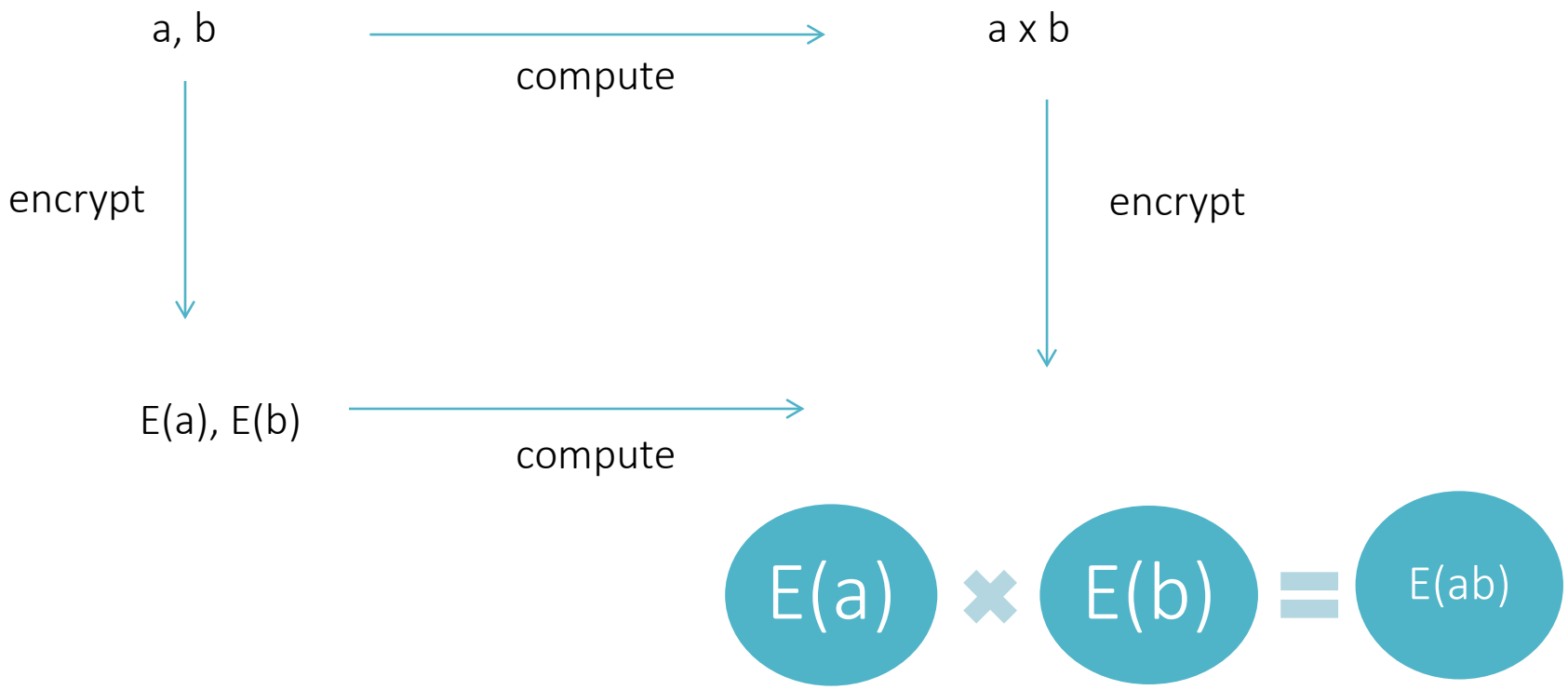
TO PROTECT GENETIC PRIVACY, ENCRYPT YOUR DNA



Homomorphic Encryption: addition



Homomorphic Encryption: multiplication



Protecting Data via Encryption:
Homomorphic encryption

1. Put your gold in a locked box.
2. Keep the key.
3. Let your jeweler work on it through a glove box.
4. Unlock the box when the jeweler is done!

4 Big Takeaways from Satya Nadella's Talk at Microsoft Build

Fortune

Jonathan Vanian
5/7/2018

© Photo credit: Microsoft Microsoft CEO Satya Nadella is focusing the company on cloud software and services.

Microsoft CEO Satya Nadella is trying to distinguish the business technology giant from its technology brethren by focusing on digital privacy.

Fortune magazine … May 7, 2018

One way Nadella is attempting to convince businesses that Microsoft msft can improve its AI technology while protecting user data is by promoting a computing technique called homomorphic encryption. Although still a research-heavy technique, homomorphic encryption would presumably let companies analyze and crunch encrypted data without needing to unscramble that information.

Nadella is pitching the technique as a way for companies to "learn, train on encrypted data." The executive didn't explain how far along Microsoft is on advancing the encryption technique, but the fact that he mentioned the wonky terms shows that the company is touting user privacy as a selling point for its Azure cloud business.

Mathematics of Homomorphic Encryption

New hard problems proposed (2009-2013), related to well-known hard lattice problems

Small Principal Ideal Problem, Approximate GCD, Learning With Errors (LWE), Ring-Learning With Errors (RLWE)

Lattice-based Cryptography:

Compare to other public key systems: RSA (1975), ECC (1985), Pairings (2000)

Proposed by Hoffstein, Pipher, and Silverman in 1996 (NTRU), Aijtai-Dwork

Hard Lattice Problems:

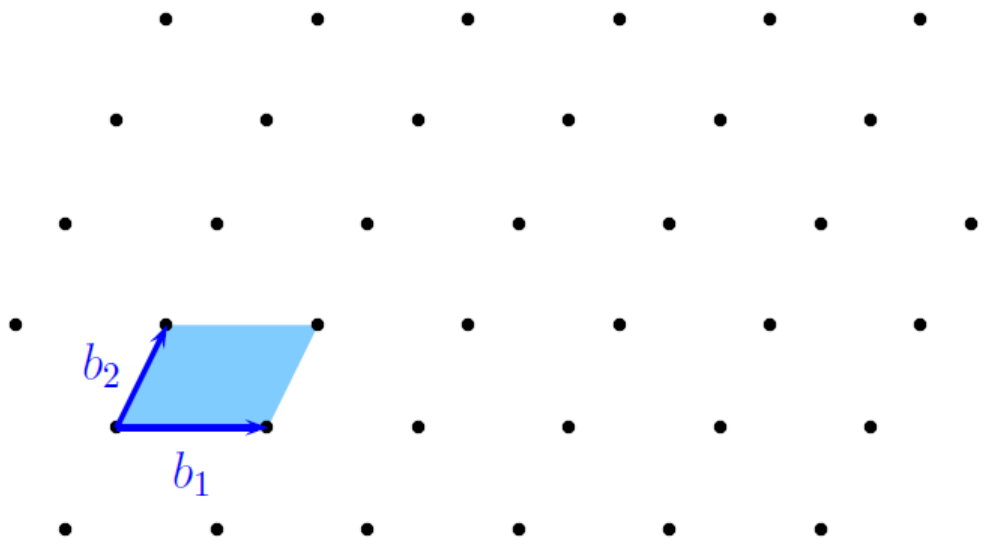
approximate Shortest Vector Problem, Bounded Distance Decoding

SECURITY:

best attacks take exponential time

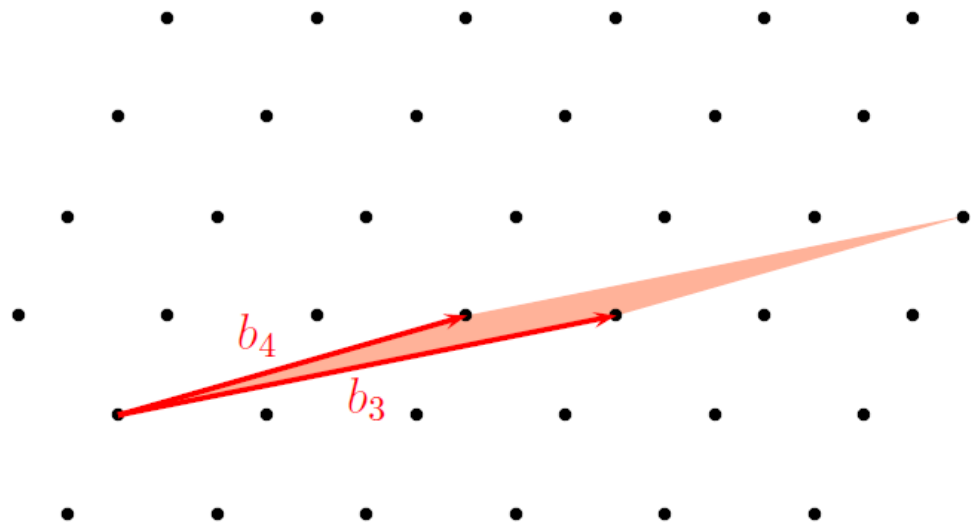
secure against quantum attacks (so far…)

Lattice with a Good (short) Basis



$$L = \mathbb{Z}b_1 + \mathbb{Z}b_2$$

Lattice with a Bad Basis



$$L = \mathbb{Z}b_3 + \mathbb{Z}b_4$$

Idea of HE schemes

Lattice vectors → coefficients of polynomials

Polynomials can be added and multiplied

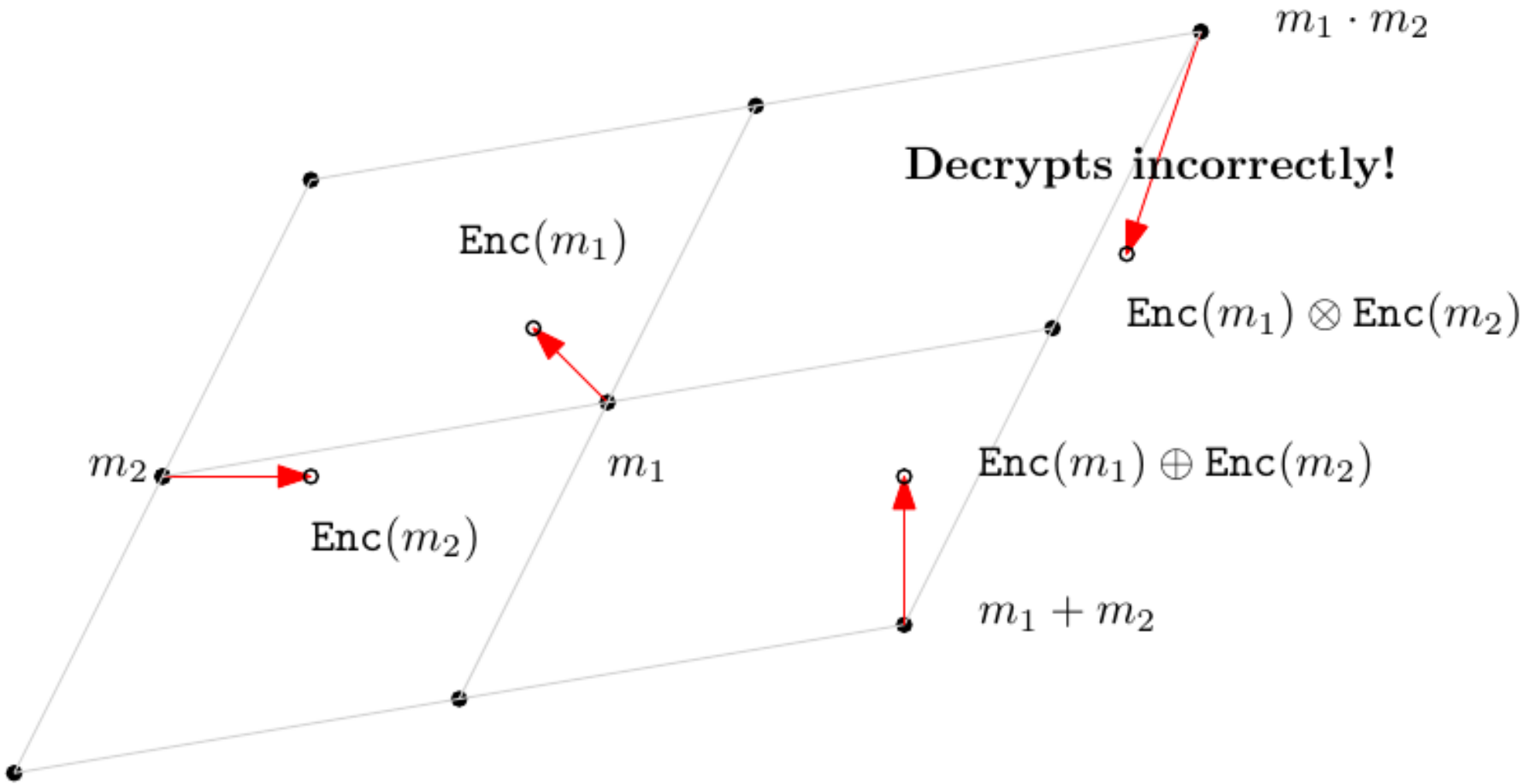
Encryption adds noise to a “secret” inner product:

message is “blinded” by secret inner product + noise

Decryption subtracts the secret and then the noise becomes easy to cancel

Hard problem is to “decode” noisy vectors

If you have a short basis, it is easy to decompose vectors



Decrypt by recovering the nearest lattice point using secret key information.

Practical Homomorphic Encryption [LNV11]

- do not need *fully* homomorphic encryption
- encode integer information as “integers”
- several orders of magnitude speed-up
- do not need deep circuits to do a single multiplication
- do not need boot-strapping
- for “logical” circuits, use ciphertext packing and tradeoff depth for ciphertext size
- need to set parameters to ensure correctness and security

•PHE=homomorphic for any fixed circuit size, with correctly chosen parameters

Ring-Learning With Errors (R-LWE)

Let $q \equiv 1 \bmod 2n$ be a prime, $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. $n=2^k$. Consider the polynomial ring

$$R_q = \mathbb{Z}_q[x]/(x^n + 1).$$

Given a secret element $s \in R_q$ and a number of pairs

$$(a_i, b_i = a_i s + e_i),$$

where $a_i \leftarrow R_q$ are chosen uniformly at random, and $e_i \leftarrow D_\sigma(R_q)$ are chosen coefficientwise according to the discrete Gaussian error distribution $D_\sigma(\mathbb{Z}_q)$.

R-LWE problem: Find the secret s (search), or distinguish whether a list of pairs (a_i, b_i) was chosen as described above or whether both $a_i, b_i \leftarrow R_q$ were chosen uniformly at random (decision).

Encryption [BV11] (symmetric key)

Sample **a** and **s** from R_q at random, and e from the error distribution.

Ciphertext $c = (c_0, c_1)$

$c_1 = -a$

$c_0 = as + 2e + m$

m = message encoded in R_2

Decrypt: $c = (c_0, c_1)$, by computing $c_0 + c_1 s \pmod 2$.

Security of Ring-LWE

Specify:

R, dimension n, ciphertext modulus q, error distribution, secret distribution

Ring R. R a 2-power cyclotomic ring $R = Z[x]/(x^n + 1)$, n is a power of 2.

Error distribution. Discrete Gaussian distribution with width

$$\sigma = 8/\sqrt{2\pi}.$$

(n,q) chosen to achieve 128-bit security against known attacks

Plaintext modulus t restricted according to (n,q)

Parameter sizes

Secret picked from Uniform distribution

n	security level	log(q)	uSVP	dec	dual
1024	128	31	130.6	133.8	147.5
	192	22	203.6	211.2	231.8
	256	18	269.9	280.5	303.6
2048	128	59	129.5	129.7	139.2
	192	42	194.0	197.6	212.4
	256	33	263.8	270.7	289.9
4096	128	113	131.9	129.4	136.8
	192	80	192.7	193.2	203.2
	256	63	260.7	263.6	277.6
8192	128	222	132.9	128.9	134.9
	192	157	195.4	192.8	200.6
	256	124	257.0	256.8	266.7
16384	128	440	133.9	129.0	133.0
	192	310	196.4	192.4	198.7
	256	243	259.5	256.6	264.1
32768	128	880	134.3	129.1	131.6
	192	612	198.8	193.9	198.2
	256	480	261.6	257.6	263.6

Algorithm to select parameters ([BLN13])

Given a task:

determine the depth of the circuit required

determine bound on the potential plaintext growth

select plaintext modulus t to exceed this bound

now (n,q) selected to satisfy 2 conditions:

1. *q/t determines the error growth bound. Choose q large enough to allow for correct decryption after the circuit is evaluated (either with or without bootstrapping)*
2. *n must be chosen large enough to achieve 128-bit security with such a q*

Size of (n,q) and the size of the circuit determine the performance.

Issues:

1. manage plaintext growth (through clever encoding techniques)
2. manage error growth (through squashing circuits or bootstrapping orrelinearization)
3. coordinate-wise sampling justified only for 2-power cyclotomic fields
4. minimize storage overhead
5. maximize use of parallelization

Practical Applications

Will you have a heart attack?

Online service running in Windows Azure

1. Patient enters personal info on local machine:
   weight, age, height, blood pressure, body mass index
2. Data is encrypted on local machine
3. Encrypted data is sent to the cloud
4. Prediction is computed on encrypted data
5. Encrypted result is sent back to the patient
6. Patient enters decryption key to decrypt answer.

Evaluation took 0.2 seconds in the cloud! (2014)

GUILLAUME PAUMIER/WIKIMEDI

Genetic gold. Each spot in a DNA microarray, such as this one, contains large amounts of sensitive genetic information.

How to Hide Your Genome

Thomas is a news intern at *Science*.

Email Thomas

Follow @SumnerSci

By Thomas Sumner | 16 February 2014 6:45 pm | 4 Comments

CHICAGO, ILLINOIS—As the cost of genetic sequencing plummets, experts believe our genomes will help doctors detect diseases and save lives. But not all of us are comfortable releasing our biological blueprints into the world. Now, cryptologists are perfecting a new privacy tool that turns genetic information into a secure yet functional format. Called homomorphic encryption and presented here today at the annual meeting of AAAS, which publishes *Science,* the method could help keep genomes private even as genetic testing shifts to cheap online cloud services.

Existing encryption techniques make data secure at the expense of making it unusable. Because of this, most genetic sequences are simply anonymized before being sent out for analysis. However, computational biologist Yaniv Erlich at the Whitehead Institute for Biomedical Research in Cambridge, Massachusetts, told meeting attendees that with a little genetic sleuthing, this supposedly anonymous data can easily track back to its owner. Erlich says he positively matched 12% of male genomes with the exact person they originated from.

In 2009, the first lattice-based cryptography scheme was announced by IBM. The geometry-based encryption method allows data to be manipulated through both multiplication and addition while remaining encrypted. Researchers realized that the complex algorithms used during genetic tests could be closely approximated by the two basic mathematical operations. Lattice cryptology enabled homomorphic encryption, allowing computers to analyze encrypted data and return encrypted results without ever being able to decode the information. Cryptologist Kristin Lauter, research manager for the cryptography group at Microsoft Research in Redmond, Washington, likened the method to locking a gold brick in a

GENEWATCH

A CIPHER FOR YOUR GENOME

By CRG staff - interview with Kristin Lauter

from *GeneWatch* 27-1 | Jan-Apr 2014

***Kristin Lauter**, PhD, is a Principal Researcher and Research Manager for the Cryptography group at Microsoft Research. She has been working on practical homomorphic encryption for several years and was a coauthor of the breakthrough paper "Can Homomorphic Encryption be Practical?"*

***GeneWatch:* How is homomorphic encryption different from other encryption technologies?**

Kristin Lauter: The primary new functionality enabled with homomorphic encryption is the ability to compute on encrypted data. This is very important for things like outsourcing storage and computation of data. The idea is that when using homomorphic encryption, the data owner - let's say it's a consumer or an enterprise - could encrypt the data locally and keep the key. Then they can upload that data to the cloud, and if they used homomorphic encryption, that data can still be operated on by the cloud and the encrypted results are available from the cloud to the data owner or anyone the data owner trusts to share the encryption key with. So it really allows a whole new functionality on encrypted data.

The problem with many other types of encryption is that it makes data secure at the expense of making it unusable. Can you say anything more about what that means?

With standard encryption systems, after you encrypt the data there is very little ability to do anything with it. For example, AES is the government's standardized block cipher. When you encrypt something with AES, you should not be able to distinguish anything about the original data or operate on it in any way which gives meaningful results. In the last ten years or so there has been a push in the field of cryptographic research to invent techniques that allow you to encrypt data and maintain its privacy but still get some functionality out of it. Homomorphic encryption is a very general and powerful tool to allow computation on encrypted data.

What type of genomic computation?

- Data quality testing
- Basic statistical functions
- Statistical computations on genomic data
- Building predictive models
- Predictive analysis
  - Classification tasks
  - Disease prediction
  - Sequence matching

Statistics on Genomic Data [LLN14]

- **Pearson Goodness-Of-Fit Test**
  - checks data for bias (Hardy-Weinberg equilibrium)
- **Cochran-Armitage Test for Trend**
  - Determine **correlation** between genome and traits
- **Linkage Disequilibrium Statistic**
  - Estimates correlations between genes
  - **Estimation Maximization (EM) algorithm for haplotyping**

The Register®
Biting the hand that feeds IT

Security

Microsoft boffins build better crypto for secure medical data crunching

Practical homomorphic encryption manual released

16 Nov 2015 at 03:57, Team Register

As genome research - and the genomes themselves - get passed around the scientific community, the world's woken up to the security and privacy risks this can involve. A Microsoft research quintet has therefore published ways to help scientists work on genomic data while reducing the risk of data theft.

The team published an informal manual to help scientists and other researchers to use the Simple Encrypted Arithmetic Library (SEAL).

Microsoft Helps Out Healthcare Sector with New Data Encryption Algorithm *UPDATE*

Microsoft reveals SEAL - Simple Encrypted Arithmetic Library

Previous reports have pointed the finger at the healthcare sector as being woefully unprepared for the modern age of Internet-enabled devices that are always online and present a constant danger to the patient, hospital, and insurer data.

This lack of security measures comes from the fact that, for many years, both the hardware and software part of healthcare applications couldn't handle the amount of data doctors and researchers needed, so no industry standards were put in place to protect sensitive data of any form.

Now, as technology has evolved, the healthcare sector is trying to catch up from behind with other industries, but the previous years, when it did not make a habit from protecting data, left a big hole to fill.

While many healthcare providers and medical research companies are putting more effort into catching up with modern-day security practices, there's still a lot of work to be done, which requires both time and financial resources to adapt various security tools to the medical industry.

Microsoft will provide a free tool to help with biomedical data processing and encryption

In a recent paper released on its research portal, Microsoft has announced a new encryption library that implements the theory of homomorphic encryption.

Homomorphic encryption is a method of encryption that encodes data in such a way that it allows developers to work with the encrypted data as if it were in unencrypted form.

Performance Summary

Data quality (Pearson Goodness-of-Fit)

~ 0.3 seconds, 1,000 patients

Predicting Heart Attack (Logistic Regression)

~ 0.2 seconds

Building models (Linear Means Classifier)

~0.9 secs train, classify: 30 features, 100 training samples

Sequence matching (Edit distance)

~27 seconds amortized, length 8

Core i7 3.4GHz

80-bit security

2014

4 years of iDASH benchmarks
Secure Genome Analysis competitions

Critical Assessment of Data Privacy and Protection (CADPP) competitions

2014: Differential Privacy (DP)

2015: Homomorphic Encryption (HE) and Secure Multi-Party Computation (MPC/SMP)

2016: Homomorphic Encryption (HE), Secure Multi-Party Computation (MPC/SMP), and Differential Privacy (DP)

2017: Homomorphic Encryption (HE), Secure Multi-Party Computation (MPC/SMP), and SGX

iDASH tasks

2015: chi^2 statistics, edit distance

2016: check for presence of string in a set of records (VCF format)

2017: train logistic regression model

Protecting genomic data analytics in the cloud: state of the art and opportunities

Haixu Tang Xiaoqian Jiang, Xiaofeng Wang, Shuang Wang, Heidi Sofia, Dov Fox, Kristin Lauter, Bradley Malin, Amalio Telenti, Li Xiong and Lucila Ohno-Machado. BMC Medical Genomics BMC series 2016**9**:63 https://doi.org/10.1186/s12920-016-0224-3

2015 iDASH Competition—UCSD

Benchmarks: iDASH 2015 (HE Track)

		5 k			10 k			100 k		
		Accuracy	Time	Memory	Accuracy	Time	Memory	Accuracy	Time	Memory
Task 1.2	Plaintext data	3099	0.076 s	1.64 MB	3306	0.118 s	2.43 MB	134252	134252	13.52 MB
	IBM	**3099**	**79.4 s**	**1.416 GB**	**3306**	**86.8 s**	**1.419 GB**	**134260**	**134260**	**2.168 G**
	Microsoft	3099	44.664 s	513.7 MB	3306	80.031 s	720.5 MB			
	Stanford/ MIT	3082	20m37s	2.77 GB	3275	36m27s	4.03 GB	132703	132703	7.50 GB

		5 k			10 k			100 k		
		Accuracy	Time	Memory	Accuracy	Time	Memory	Accuracy	Time	Memory
Task 1.2	Plaintext data	9089	0.106 s	2.45 MB	16667	0.144 s	2.53 MB	191986	1.528 s	25.8 MB
	IBM[b]	5328	91.7 s	1.42 GB	8318	106.3 s	1.45 GB	153266	555.2 s	2.29 GB
	Microsoft	**9089**	**91.09 s**	**701 MB**	**16665**	**181.92 s**	**1.29 GB**			

2016 iDASH Competition—Chicago

Track 1: *harden Beacons from detection of an individual's presence in a data set*

Track 2: *privacy-preserving searches of patient genomic data across organizations*

Track 3: *securing data resulting from genetic testing in a public cloud*

17 solutions from 16 teams in 7 countries.

Benchmarks: iDASH 2016 (HE Track)

Table: A summary of the results for Track 3 (secure outsourcing)

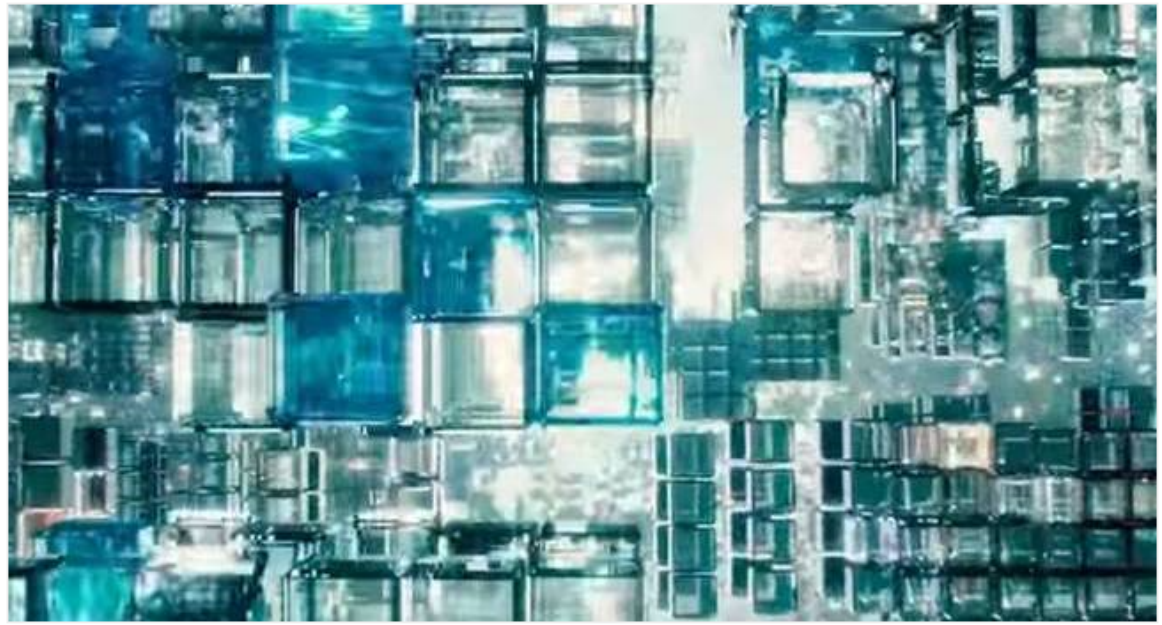
Team	Data encryption time (s)	Encrypted data size (MB)	Secure computing time (s)	Result decryption time (s)	Total time (s) for computing, result decryption and transfer
Microsoft [47]	1.86	24.00	3.09	0.02	3.63
RWTH Aachen University [48]	34.90	255.00	15.28	0.68	16.32
EPFL [49]	137.60	147.00	6.79	9.28	19.26
Seoul National University [50]	51.02	10.00	21.10	0.005	25.11
IBM team 2	478.10	1660.00	959.10	200.70	1178.2
Waseda University	109.72	5447.82	8937.51	0.058	8938.81

The Register®
Biting the hand that feeds IT

Software

Microsoft researchers smash homomorphic encryption speed barrier

Artificial intelligence CryptoNets chew data fast but keep it safe



Ultron it isn't, thank goodness

9 Feb 2016 at 08:02, Iain Thomson

Exclusive Microsoft researchers, in partnership with academia, have published a paper detailing how they have dramatically increased the speed of homomorphic encryption systems.

With a standard encryption system, data is scrambled and then decrypted when it needs to be

Demos

SEAL Team: Ran Gilad-Bachrach, Kim Laine, Hao Chen, Kristin Lauter, with Nicolo Fusi, Rich Caruana, John Wernsing, Michael Naehrig, Nate Dowlin

Sealcrypto.org

Publicly available HE Libraries

HELib: This is an early and widely used library from IBM that supports the BGV scheme and bootstrapping.

SEAL: This is a widely used library from Microsoft that supports the FV scheme.

NFLlib: This library is an outgrowth of the European HEAT project to explore high-performance homomorphic encryption using low-level processor primitives.

PALISADE: This is a general lattice encryption library that supports several lattice encryption schemes, including multiple homomorphic encryption schemes.

cuHE: This library explores the use of GPGPUs to accelerate homomorphic encryption.

HeaAn: This library implements a scheme with native support for fixed point approximate arithmetic.

TFHE: bootstrapping after every gate

Community effort to benchmark and standardize

*NIH sponsored iDASH competitions (2014-2017)

Secure Genome Analysis Competitions

*Homomorphic Encryption Standardization effort

- Hosted by Microsoft Research, July 2017
- Whitepapers on Security, Applications, and APIs
- Working group formed to draft standard
- 2nd Workshop at MIT in March 2018, draft standard approved
- *http://homomorphicencryption.org/*

Add yourself to the community by joining the mailing list! standards@HomomorphicEncryption.org

Homomorphic Encryption Standardization Workshop:

Microsoft Research, July 13-14, 2017

Homomorphic Encryption

Established: March 27, 2016

Featured News

Tales from the Crypt(ography) Lab with Dr. Kristin Lauter | Microsoft Research Podcast, April 11, 2018

Second homomorphic encryption standardization workshop delivers the goods | Microsoft Research Blog, April 10, 2018

Homomorphic Encryption

Homomorphic Encryption (HE) refers to a special type of encryption technique that allows for computations to be done on encrypted data, without requiring access to a secret (decryption) key.

People

Kristin Lauter
Principal Researcher, Research Manager

Kim Laine
Researcher

Ran Gilad-Bachrach
Researcher

Hao Chen

Paths to Standardization in Cryptography

- Industry consortiums (e.g. PKCS for RSA)
- Government standards NIST (e.g. 800-90 Random Number Generators)
- IEEE (P1363 for Elliptic Curve Cryptography), ietf, ISO standards, …
- X.509 certificates (International Telecommunications Union's Standardization sector
- ANSI Financial Standards (X9.62 and X9.63 for ECDH and ECDSA)
- FIPS 140 certification process

Joint work with:

…and thanks to iDASH and co-authors for selected slides…

SEAL Team: Kim Laine, John Wernsing, Michael Naehrig, Ran Gilad-Bachrach, Nathan Dowlin, Kristin Lauter, Hao Chen

Can Homomorphic Encryption be Practical?

[LNV11] Kristin Lauter, Michael Naehrig, Vinod Vaikuntanathan, CCSW 2011

ML Confidential: Machine Learning on Encrypted Data

[GLN12] Thore Graepel, Kristin Lauter, Michael Naehrig, ICISC 2012

Predictive Analysis on Encrypted Medical Data

[BLN14] Joppe W. Bos, Kristin Lauter, and Michael Naehrig, Journal of Biomedical Informatics, 2014.

Private Computation on Encrypted Genomic Data

[LLN14] Kristin Lauter, Adriana Lopez-Alt, Michael Naehrig, GenoPri2014, LatinCrypt2014.

Homomorphic Computation of Edit Distance

[CKL15] Jung Hee Cheon, Miran Kim, Kristin Lauter, WAHC, FC 2015