Fully Homomorphic Encryption

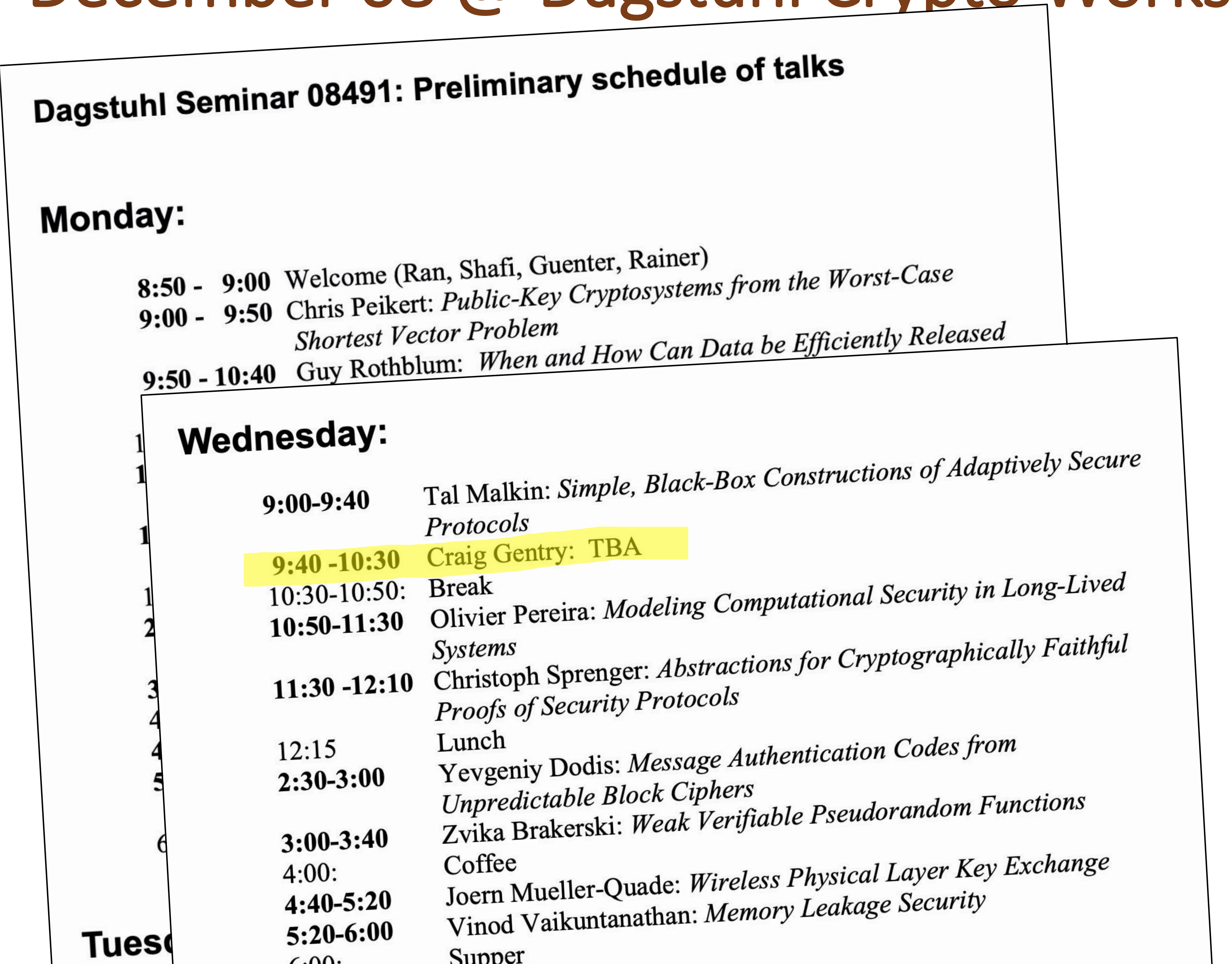
Zvika Brakerski

Weizmann

European Research Council
Established by the European Commission

Homomorphic Encryption Standards Meeting, September 2022

December 08 @ Dagstuhl Crypto Workshop



Dagstuhl Seminar 08491: Preliminary schedule of talks

Monday:

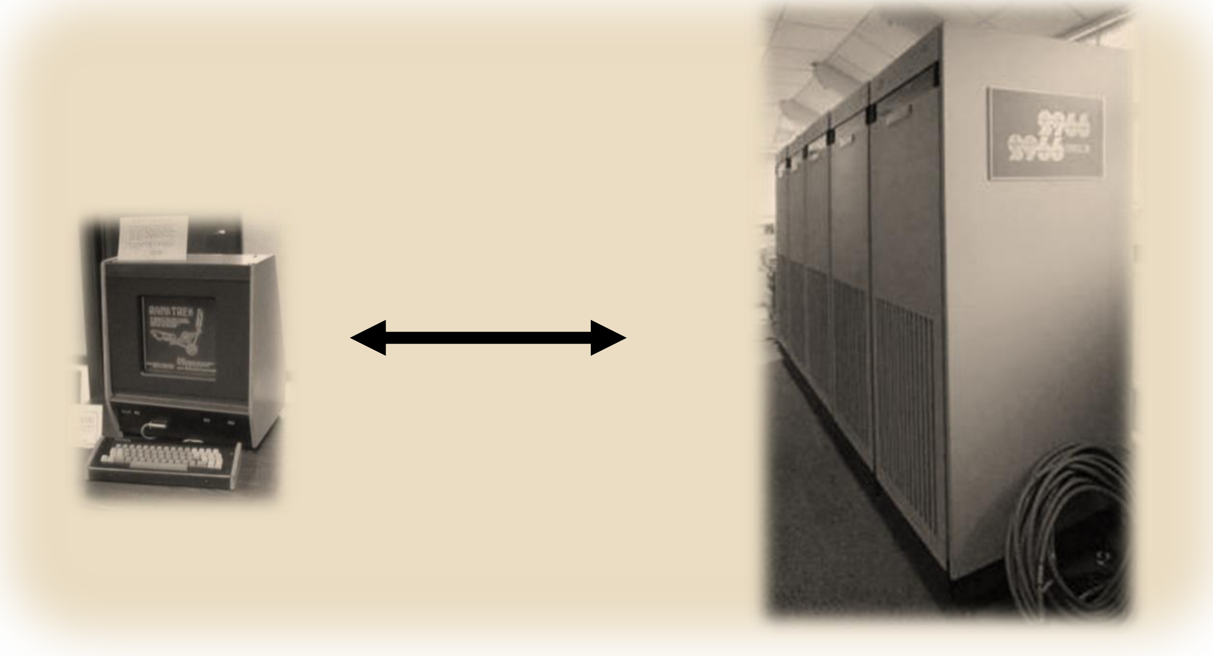
8:50 - 9:00	Welcome (Ran, Shafi, Guenter, Rainer)
9:00 - 9:50	Chris Peikert: *Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem*
9:50 - 10:40	Guy Rothblum: *When and How Can Data be Efficiently Released*

Tues

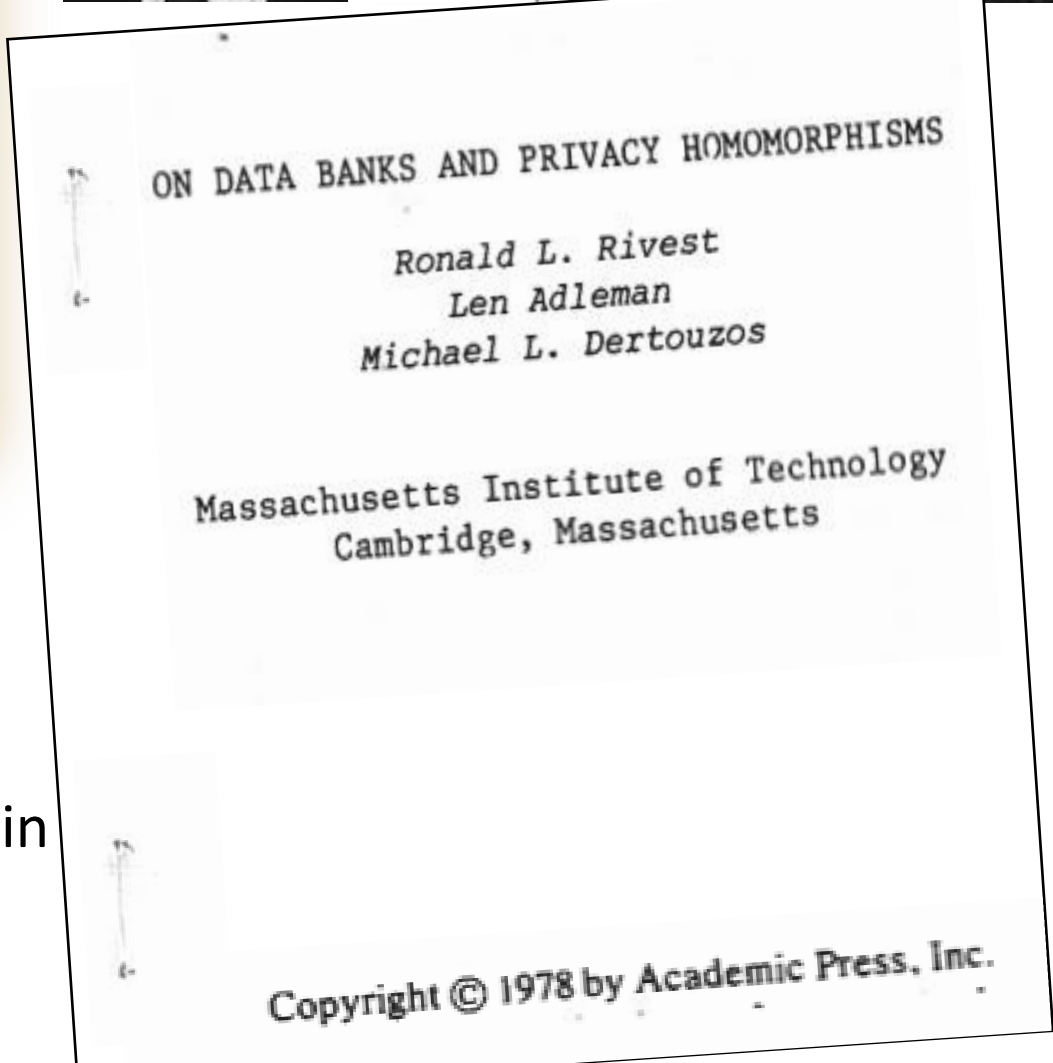
Wednesday:

9:00-9:40	Tal Malkin: *Simple, Black-Box Constructions of Adaptively Secure Protocols*
9:40 -10:30	Craig Gentry: TBA
10:30-10:50:	Break
10:50-11:30	Olivier Pereira: *Modeling Computational Security in Long-Lived Systems*
11:30 -12:10	Christoph Sprenger: *Abstractions for Cryptographically Faithful Proofs of Security Protocols*
12:15	Lunch
2:30-3:00	Yevgeniy Dodis: *Message Authentication Codes from Unpredictable Block Ciphers*
3:00-3:40	Zvika Brakerski: *Weak Verifiable Pseudorandom Functions*
4:00:	Coffee
4:40-5:20	Joern Mueller-Quade: *Wireless Physical Layer Key Exchange*
5:20-6:00	Vinod Vaikuntanathan: *Memory Leakage Security*
6:00:	Supper

Rewind 30 Years…



FHE unsolved for a long time, feasibility questioned (e.g. in light of impossibility of obfuscation [B+01])



ON DATA BANKS AND PRIVACY HOMOMORPHISMS

Ronald L. Rivest
Len Adleman
Michael L. Dertouzos

Massachusetts Institute of Technology
Cambridge, Massachusetts

Copyright © 1978 by Academic Press, Inc.

Gentry’s Breakthrough

Basic scheme from ideal lattices – tailored key generation process to get ideal lattices with “good” structure

Keygen very exhaustive

Noise accumulates double-exponentially fast

+

Bootstrapping technique for noise reduction

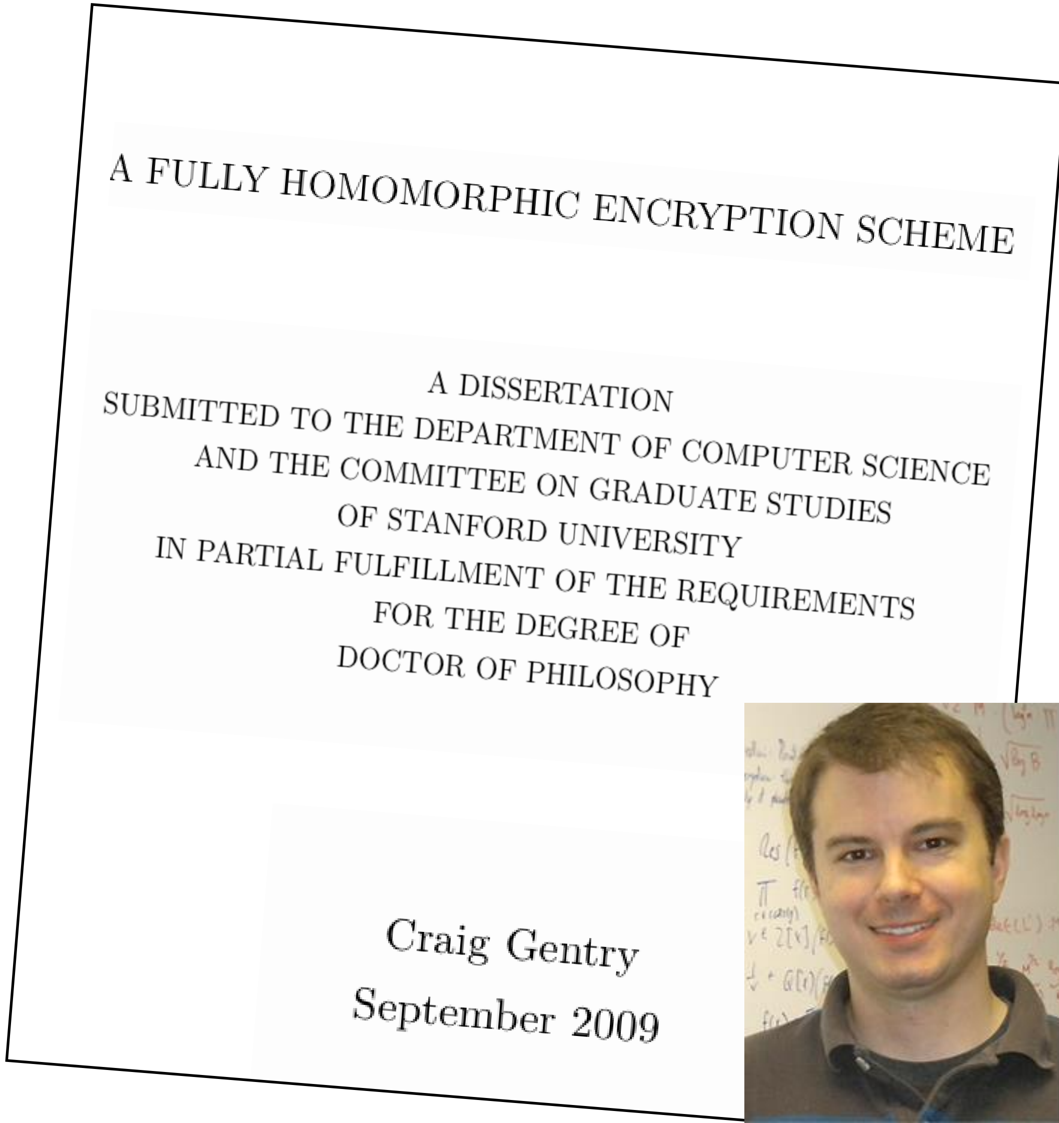
Circular security

Still doesn’t fit together!

+

“Squashing”

Other schemes using this blueprint [SV10,DGHV10]



A FULLY HOMOMORPHIC ENCRYPTION SCHEME

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Craig Gentry
September 2009

FHE Evolution – From Ideal Lattices to R/LWE

RLWE-Based Tensoring [BV11a]

dim-blowup in RLWE not so bad since $\vec{s} = (s, 1) \Rightarrow$ can still use squashing

In R/LWE: $m \approx \vec{c} \cdot \vec{s} \ (mod\ q)$

Therefore $m_1 \cdot m_2 \approx (\vec{c}_1 \cdot \vec{s})(\vec{c}_2 \cdot \vec{s}) = (\vec{c}_1 \otimes \vec{c}_2) \cdot \underbrace{(\vec{s} \otimes \vec{s})}_{\vec{s}^2}$

oh no! dimension-blowup!

Makes KeyGen trivial!

LWE-Based Tensoring [BV11b] – Squashing not needed, only LWE

Key Switching – Use $evk = C = Enc_{\vec{s}}(\vec{s}^2)$ to switch sk but keep message $\vec{s}^2 \rightarrow \vec{s}$

more generally $Enc_{\vec{t}}(\vec{s})$ switches $\vec{s} \rightarrow \vec{t}$

Binary Decomposition – CT must be “small”, achieve this by breaking into bits (cost $\times \log q$)

Modulus Switching – Modulus too big to bootstrap, so chop LSBs of ciphertext to decrease

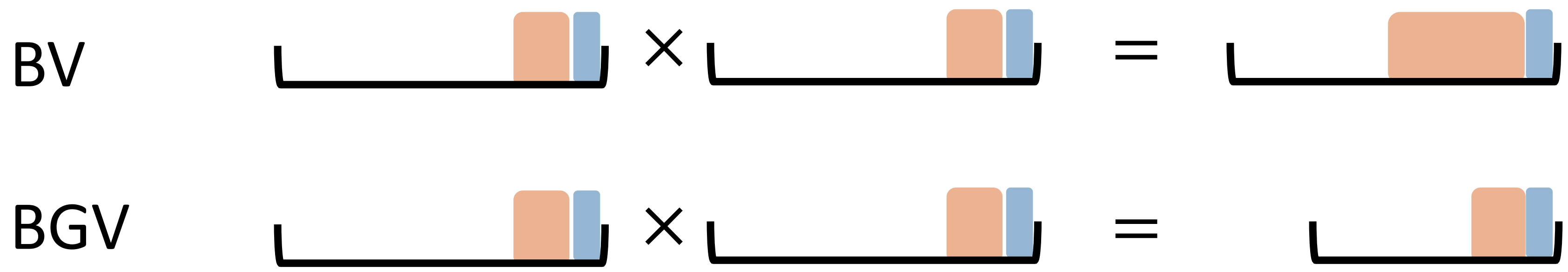
FHE Evolution – Towards a Usable Scheme

Continual Modulus Switch [BGV12]

Error growth becomes (single) exponential – “reasonable” functionality without bootstrapping

Boost in efficiency and security follows

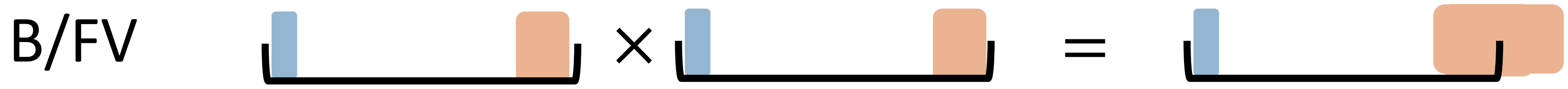
Batching techniques using RLWE introduced and further improved in [GHS12a,GHS12b].



Scale Invariance [B12]

Why should the modulus q affect FHE capacity? (Except for representation length)

Adopted to RLWE by [FV12] with some useful optimizations (B/FV)



FHE Evolution – Make Matrices, Not Vectors

Approximate Eigenvalue Method [GSW13]

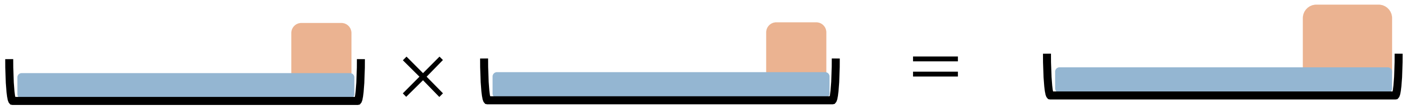
Ciphertexts are matrices, hom. operations use bit decomposition

(My perspective: Composition of key-switching gadgets)

Not immediately clear what it is good for…

Noise can be controlled better, with computational cost [BV14]

Efficient bootstrapping implementation leading to FHEW [DM16] and TFHE [CGGI16]



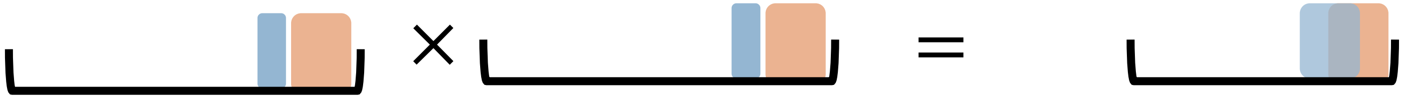
Combination of [GSW13,AP14] with other advances [ABB10,MP12] led to an extraordinary advancement in cryptography with new schemes for attribute-based encryption, forms of program obfuscation, traitor tracing schemes, constrained pseudorandom functions…

FHE Evolution – The Real (Valued) World

FHE Over the Reals [CKKS17]

Message LSBs don’t matter, can afford to lose them – great for ML algorithms

New approaches to bootstrapping by considering real-valued functions



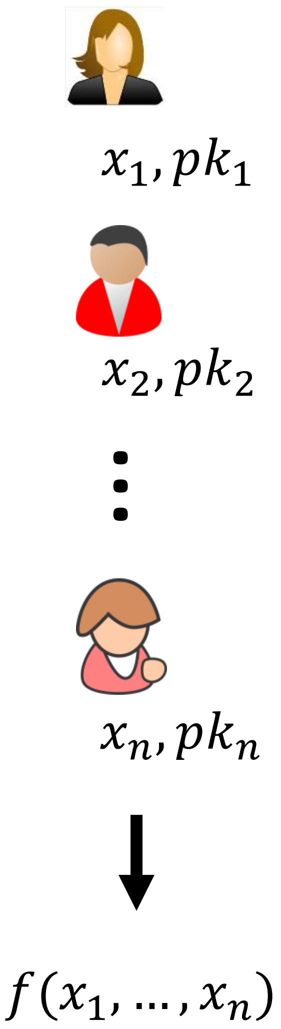
FHE Evolution – Multiple Users

Multi-Key FHE [LTV12,CM15,MW16,…]

Processing information from a few sources together

All keys are needed to decrypt – need to run MPC protocol for decryption

Threshold MK-FHE : MPC protocol is just a single message



x_1, pk_1

x_2, pk_2

$\vdots$

x_n, pk_n

$f(x_1, \dots, x_n)$

Implementing FHE

Pioneering work by Shai Halevi

Implementation of Gentry’s scheme + challenges for KeyGen

HELib – First FHE Library

Implements BGV w/ multiple optimizations and improvements

[GHS12a, GHS12b, GHPS13]

Paved the way for many others, some examples:

SEAL

PALISADE

HEEAN

FHEW

TFHE

Concrete

NFLLib

$\Lambda \circ \lambda$

Lattigo

cuFHE

FHE Meets World

Workshops, online resources, software libraries, programing languages, compilers, hardware design, companies, government initiatives, non-profit initiatives, collaboration, standartidation…

Selling FHE

What is the right way to promote FHE without creating misconceptions?

FHE is amazing – non-experts can easily believe it can solve all privacy issues

Making FHE more accessible is great! Should also make the user aware of where it can and cannot be used

Are We Done with FHE Theory?

FHE w/o circular security assumption?

Maybe ”fully circular” harder than FHE? Can be achieved via obfuscation path

Break away from “noisy lattice” paradigm?

Only known method uses obfuscation [CLTV15]+[JLS22]
Very indirect approach – is it necessary?

Bootstrapping used even there – possible to do without?

Asymptotic efficiency for MK-FHE?

CT size independent of #parties, simulation soundness under polynomial assumptions

Vinod’s bounties to solve – better be quick before it’s eroded by inflation

The Grand Challenge: Concrete Efficiency

Can FHE have “almost zero” overhead?

Best per-operation overhead = polylog [GHS12b], concrete performance unsatisfactory

Communication Overhead

Recently rate-1 FHE [BDGM19,GH19] – asymptotically no overheard

- Requires significant batching
- Only compressing “at transport” – need to unpack to compute

Key Sizes

Ciphertexts are big, but evk is *huge*

- Asymptotically “not a problem”
- Actually, need to swap them in-and-out of processor for every operation (esp. bootstrapping)
- Can data busses handle the load?

Beyond Per-Operation Efficiency

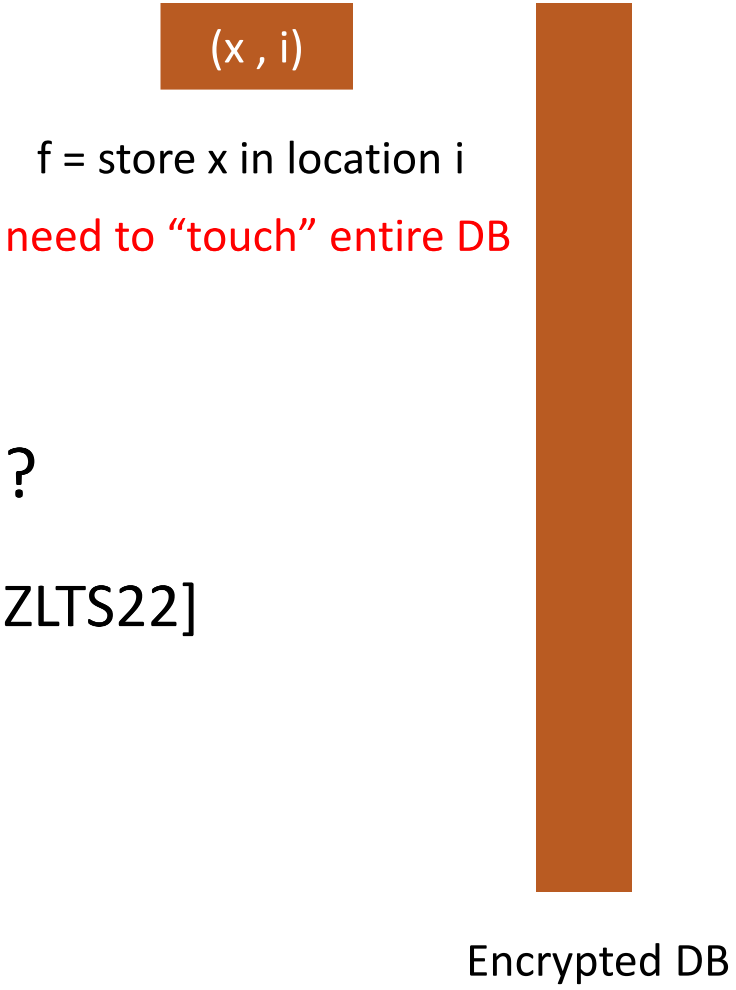
Computational model issue – FHE applies in the circuit model

Branching ops are expensive – lucky we can do anything at all…

Pipelining? Conditional execution?

What can we hope for? Can we get FHE for RAM [HHWW19]?

Recent progress on PIR with different tradeoffs e.g. [CHK22,ZLTS22]



f = store x in location i

need to “touch” entire DB

FHE & Quantum Computing

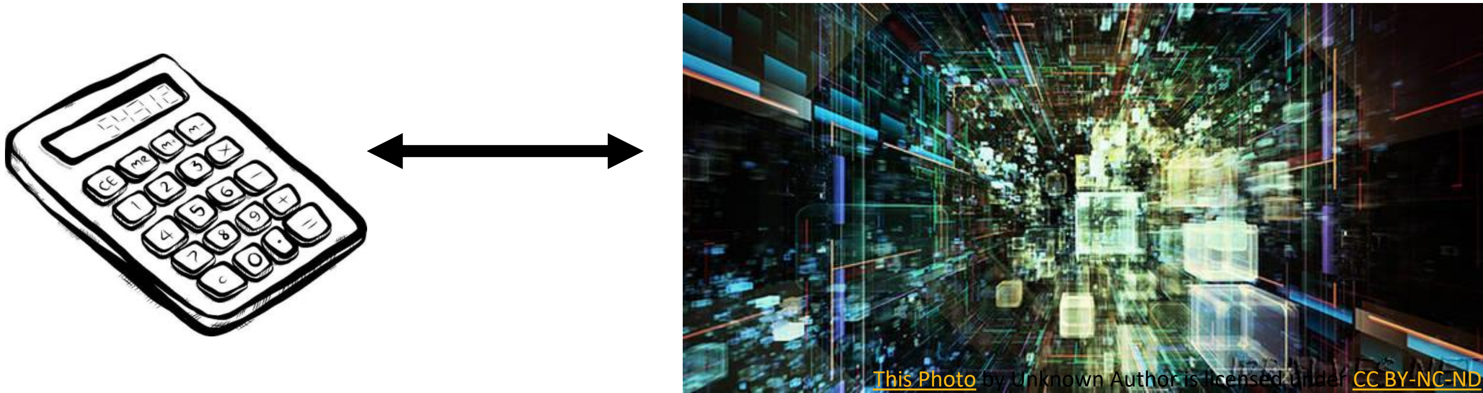
Usually: “Post-quantum” security, i.e. security against quantum *attacker*

What about using quantum computing for *good*? Still need crypto

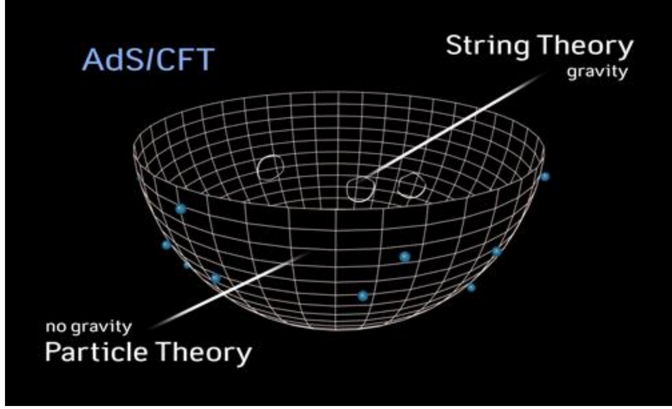
Classical party wants quantum server to solve a problem Verification? Privacy?

Quantum FHE (QFHE) should allow Q to compute on data while preserving privacy

Other applications: “Proofs of Quantumness” [KLVY22], Q. money [S22a,S22b], …



FHE and the Secrets of the Universe



Can FHE emerge from the fundamental laws of nature?

AdS/CFT : Most researched topic in high-energy physics

A duality between quantum gravity and “standard” quantum fields

mapping is computationally hard [BFV18]
(at least in strong gravity)

Speculation [Aaronson, Gottesman, Susskind (reinterpreted, simplified)]:

- Imagine “black-hole computer” with input x in memory in AdS space
- Take the CFT description of this universe (going back to AdS is hard)
- Execute the universe evolution function
- Tada, a function is hom. evaluated on x

Clearly some crucial parts are missing (e.g. keys)

Thank you!