Post-quantum key exchange from (ideal) lattices

Craig Costello

Microsoft® Research

Part 1: Recap and motivation

Part 2: Lattice basics

Part 3: PQ key exchange based on (R)LWE

Diffie-Hellman key exchange (circa 2016)

$q =$

580960599536995806285950253330457437068697517636289523666148615228720373099711022573733604453311840725132615775498051744399052959454004712166288567218703240103211163970644049884404985098905162720024476580704181239472968054002410482797658436938152229236120877904476989274322575173807697956881130957912551133309324351955378481630638158016186020024749256844815024251530444957718760413642873858099017255157393414625583036640591500086964373205321856683254529110790372283163413859958640669032595972518744716905954080501231020963901175074876001709536073423494575741627299485601330861695852995830467763701918159408852834506128586389827176345729488354663887955431161544644633019925438234001629205709075117553388816191898729559153153669870129226768546551743791579082315484463478026010289171803249539607504189948551381112697730747896907485704371071615012131592202455675924123901315291971095646840637944291494161435710791446256732969364 9

$g = 123456789$

$g^a \pmod q =$

197496648183227193286262018614250555971909799762533760654008147994875775445667054218578105133138217497206890599554928429450667899476
854668595594034093493637562451078938296960313488696178848142491351687253054602202966247046105770771577248321682117174246128321195678
537631520278649403464797353691996736993577092687178385602298873558954121056430522899619761453727082217823475746223803790014235051396
799049446508224661850168149957401474638456716624401906701394472447015052569417746372185093302535739383791980070572381421729029651639
304234361268764971707763484300668923972868709121665568669830978657804740157916611563508569886847487772676671207386096152947607114559
706340209059103703018182635521898738094546294558035569752596676346614699327742088471255741184755866117812209895514952436160199336532
605242210147489825669666012419572610049572551002200293281421876806011231076345540456724876139639963334490185787211920851855080379172 4

$g^a \pmod q =$

411604662069593306683228525653441872410777999220572079993574397237156368762038378332742471939666544968793817819321495269833613169937
986164811320795616949957400518206385310292475529284550626247132930124027703140131220968771142788394846592816111078275196955258045178
705254016469773509936925361994895894163065551105161929613139219782198757542984826465893457768888915561514505048091856159412977576049
073563225572809880970058396501719665853110101308432647427786565525121328772587167842037624190143909787938665842005691911997396726455
110758448552553744288464337906540312125397571803103278271979007681841394534114315726120595749993896347981789310754194864577435905673
172970033596584445206671223874399576560291954856168126236657381519414592940237018351232440467191228145585909045861278091800166330876
407323844719948807012687304886027922176162928196104625521958432771481724862624396241361307595677001801738572499949511777914941688218 8

$g^b \pmod q =$

655456209464694; 93360682685816031704
969423104727624682511774387497061288
879957701\936988268597627904791130623
089758634282837985890970179573655906
72\835713863895712246676094993008985
548024466403039544300748002507962036
386619315229886063541005322448463915
897986412102737725583739651\486539312
854838650709031919742048649235894391
903529930326769610051088404319792729
916038927477470940948581926791161465
028635214849871\086232861934222391717
121545686125300672760188085915004248
49476686\706784051068715397706852664
532638332403983747338379697022624261
377163163204493828299206039808703403
575100467337085017748387148822224875
309641791879395483731754620034884930
540399950519191679471224\055585570931
219350747155777569598163700850920394
705281936392411084\436006861835284651
724969562186437214972625833222544865
996160464558\546299370165894704252644456241578995869726529356478569670926896044\427965012098770368450012467926761563917639959736383038665362727158

ECDH key exchange (1999 – nowish)

$$p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$$

$p = 115792089210356248762697446949407573530086143415290314195533631308867097853951$

$$E/\mathbf{F}_p: y^2 = x^3 - 3x + b$$

$\#E = 115792089210356248762697446949407573529996955224135760342422259061068512044369$

$P = (48439561293906451759052585252797914202762949526041747995844080717082404635286,$
$36134250956749795798585127919587881956611106672985015071877198253568414405109)$

$a =$
89130644591246033577639
77064146285502314502849
28352556031837219223173
24614395

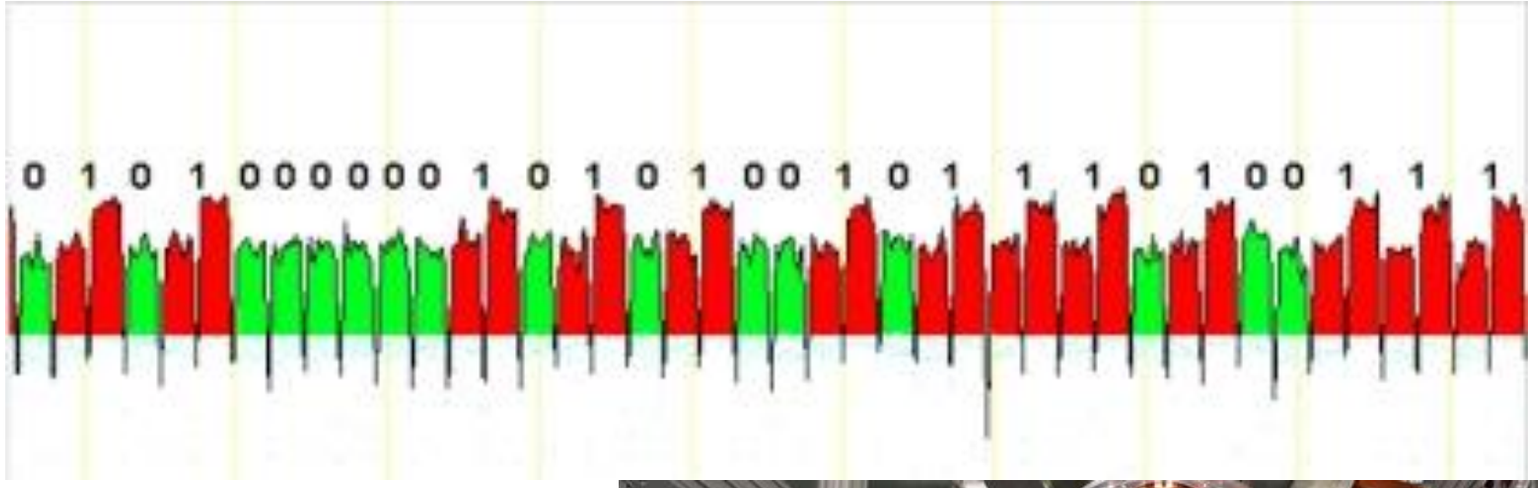
[a]$P = (841162082613158981675930678682005256123442218863337853315847934354495016584 16,$
$102885655542185598026739250172885300109680266058548048621945393128043427650740)$

[b]$P = (101228882920057626679704131545407930245895491542090988999577542687271695288383,$
$778874181903040229941165950345562577608071856156796893721381343639784983415 94)$

$b =$
10095557463932786418806
93831619070803277191091
90584053916797810821934
05190826

[ab]$P = (101228882920057626679704131545407930245895491542090988999577542687271695288383,$
$778874181903040229941165950345562577608071856156796893721381343639784983415 94)$

What's happened since 1999?



Quantum computers ↔ Cryptopocalypse

- Quantum computers break elliptic curves, finite fields, factoring, everything currently used for PKC

- Aug 2015: NSA announces plans to transition to quantum-resistant algorithms

NIST
National Institute of Standards and Technology

- Feb 2016: NIST calls for quantum-secure submissions

Elliot Carr
Friday 1 April: Maths Seminar by Craig Co... Tue 9:25 AM
Dear All,

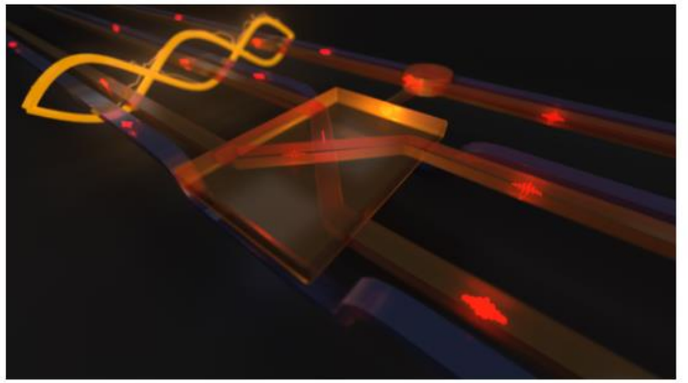
Christian Paquin
QC hardware breakthrough: researchers ... Tue 5:52 AM
One step closer, millions more to go J

Michael Naehrig
Status for the week ending March 25th, 2... Tue 4:07 AM
PQCrypto

Home / Hardware

Quantum computing is now a big step closer thanks to a new breakthrough: the Fredkin gate

For the first time ever, scientists have found a way to build a quantum Fredkin gate.



An artist's rendering of the quantum Fredkin gate, powered by entanglement, operating on photonic qubits.
Credit: Raj Patel and Geoff Pryde, Center for Quantum Dynamics, Griffith University.

COMMENTS

Katherine Noyes
IDG News Service Mar 28, 2016 11:24 AM

Quantum computing is now within closer reach thanks to a major breakthrough in which scientists have demonstrated that a key building block can be assembled.

Quantum computers are based on atomic-scale quantum bits, or qubits, that can

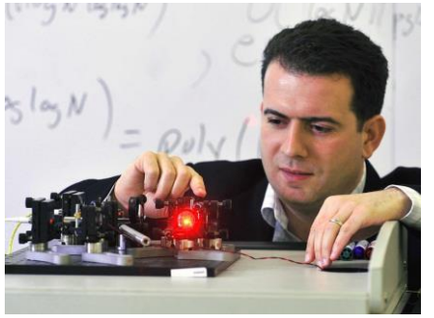
Cryptopocalypse now?

x = how long information needs to be secure

y = how long it takes to deploy

z = how far away is a quantum computer

Mosca's theorem:



if $x + y > z$, we're screwed!

Post-quantum key exchange

What hard problem(s) do we use now???

Codes?

Isogenies?

This talk: lattice problems

Multivariate eq's?

Just key exchange for this talk…

- Simplicity and ease of exposition
- [BCNS'15]'s excuse: more important to quantum-secure key exchange first
- Possible to do lattice-based [*insert favorite primitive*]

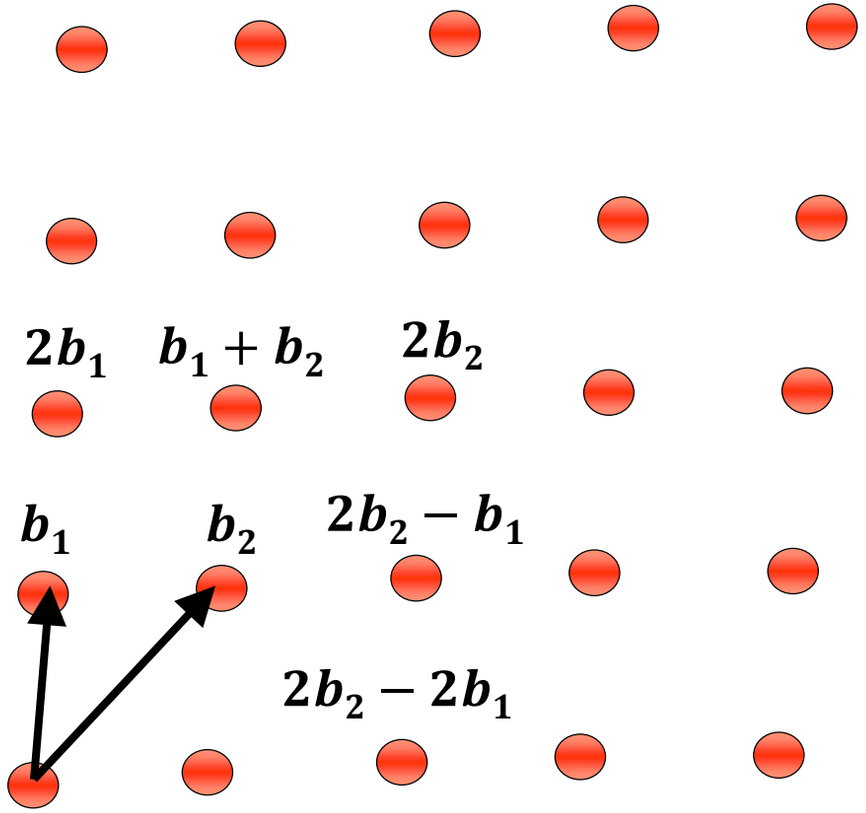
Part 1: Recap and motivation

Part 2: Lattice basics

Part 3: PQ key exchange based on (R)LWE

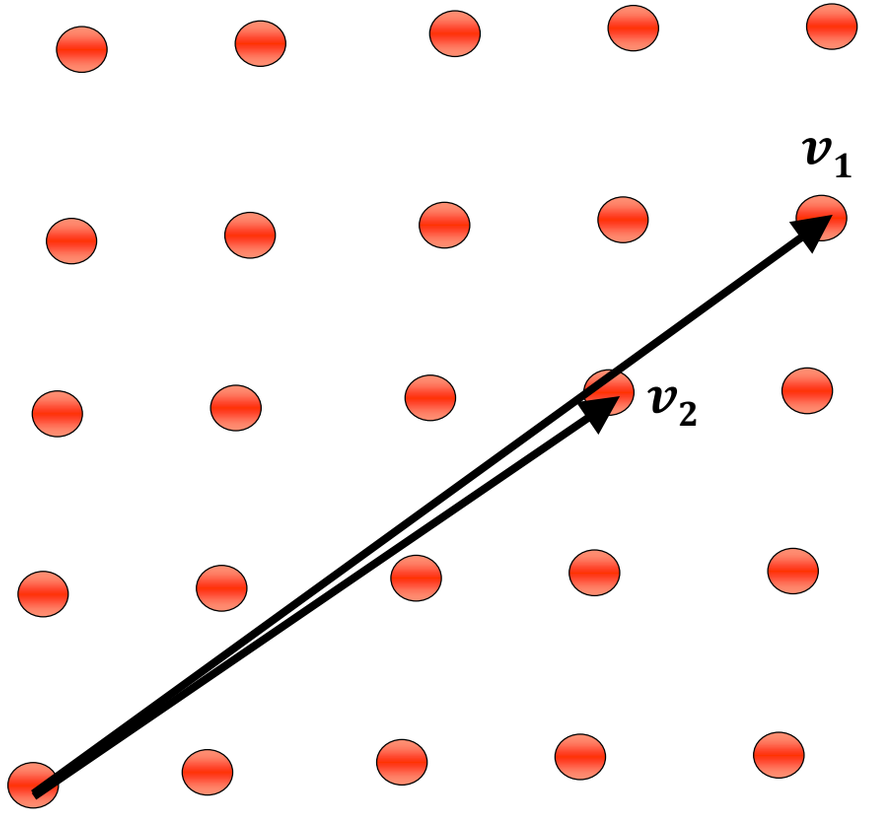
Lattices

- Basis $\boldsymbol{b_1, \ldots, b_n} \in \mathbb{R}^n$
- Lattice $\boldsymbol{L} = \{a_1\boldsymbol{b_1} + \cdots + a_n\boldsymbol{b_n} : a_i \in \mathbb{Z}\}$



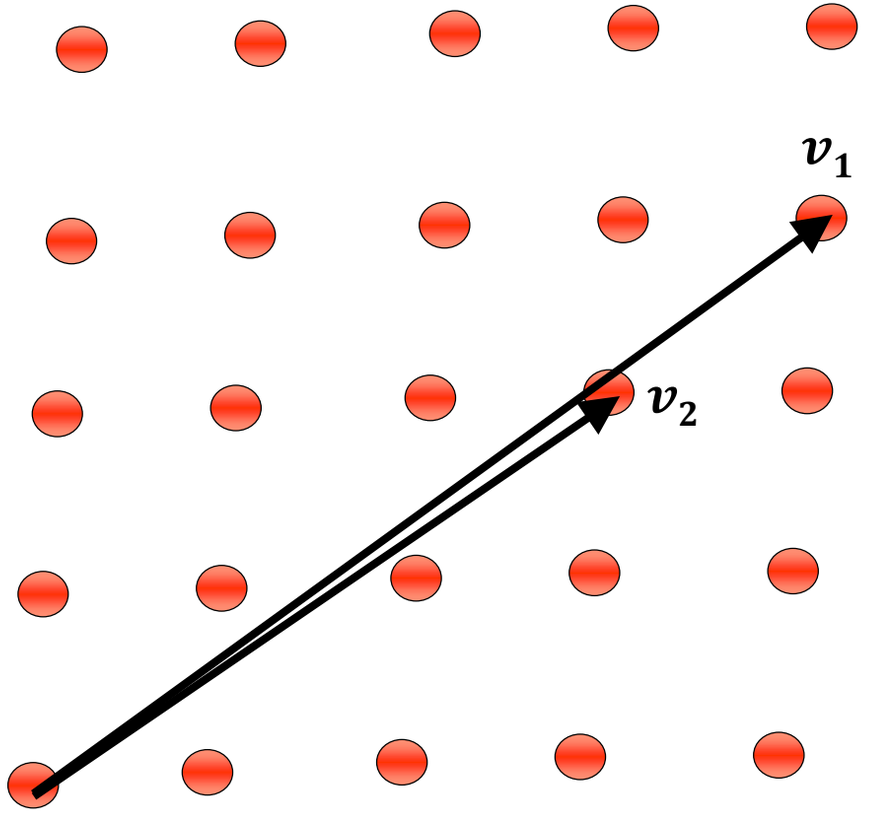
Lattices

- Basis $\boldsymbol{b_1, \ldots, b_n} \in \mathbb{R}^n$
- Lattice $\boldsymbol{L} = \{a_1\boldsymbol{b_1} + \cdots + a_n\boldsymbol{b_n} : a_i \in \mathbb{Z}\}$
- Bases not unique $\boldsymbol{L} = \sum a_i \boldsymbol{v_i}$
- e.g., $b_1 = (-2,\ 1),\ b_2 = (10,6)$
  $v_1 = (4, -3),\ v_2 = (2,4)$



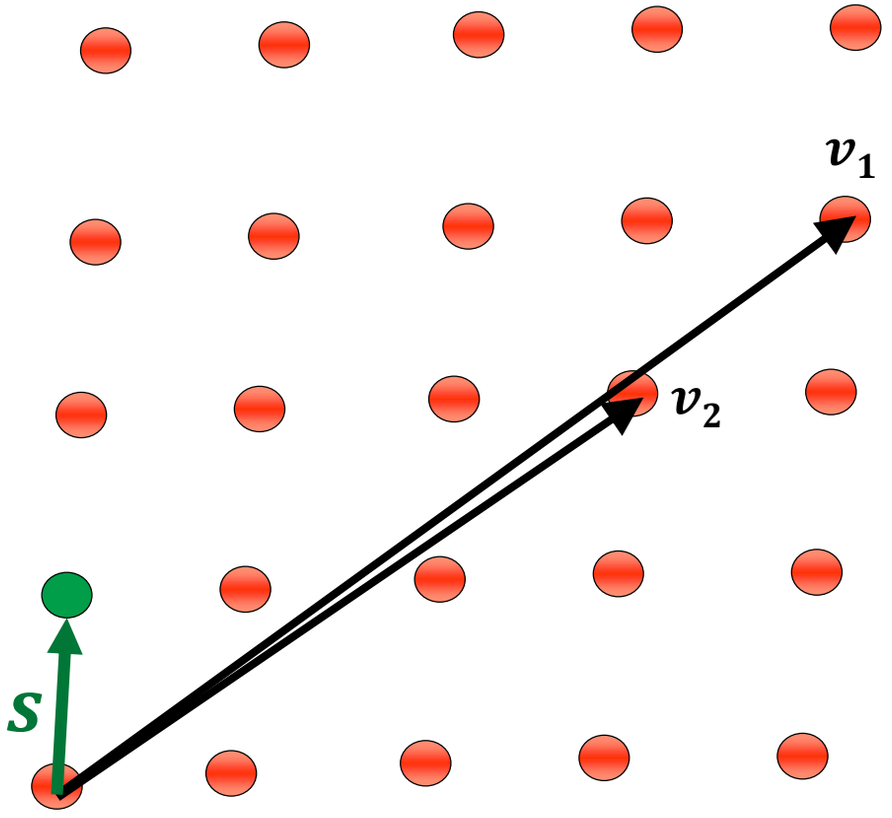
Lattices

- Basis $\boldsymbol{b_1, \ldots, b_n} \in \mathbb{R}^n$
- Lattice $\boldsymbol{L} = \{a_1\boldsymbol{b_1} + \cdots + a_n\boldsymbol{b_n} : a_i \in \mathbb{Z}\}$
- Bases not unique $\boldsymbol{L} = \sum a_i \boldsymbol{v_i}$
- e.g., $b_1 = (-2,\ 1),\ b_2 = (10,6)$
  $v_1 = (4, -3),\ v_2 = (2,4)$
- Invariant $\det(\boldsymbol{L}) = |\det(\boldsymbol{b_i})| = |\det(\boldsymbol{v_i})|$



$$\underbrace{\begin{bmatrix} -2 & 1 \\ 10 & 6 \end{bmatrix}}_{\boldsymbol{b_i}} = \underbrace{\begin{bmatrix} 4 & -3 \\ 2 & 4 \end{bmatrix}}_{\boldsymbol{v_i}} \cdot \underbrace{\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}}_{\boldsymbol{\det = \pm 1}}$$

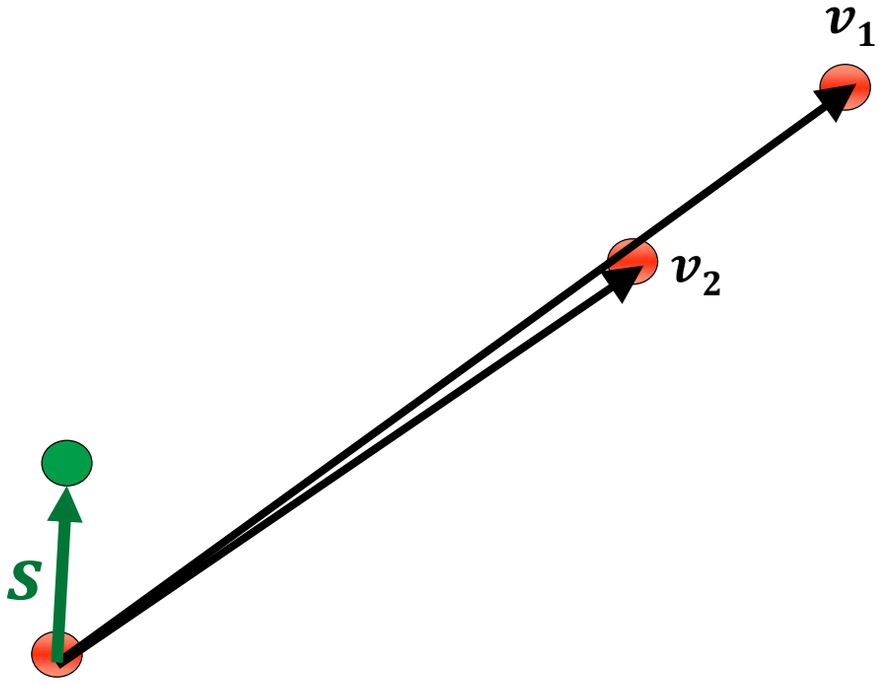
Hard Lattice Problem #1: Shortest Vector Problem (SVP_γ)



SVP: Given lattice $L = \{v_1, v_2\}$, find short vector $|s| \leq \gamma \cdot \lambda(L)$

($\gamma = 1$ means shortest vector)

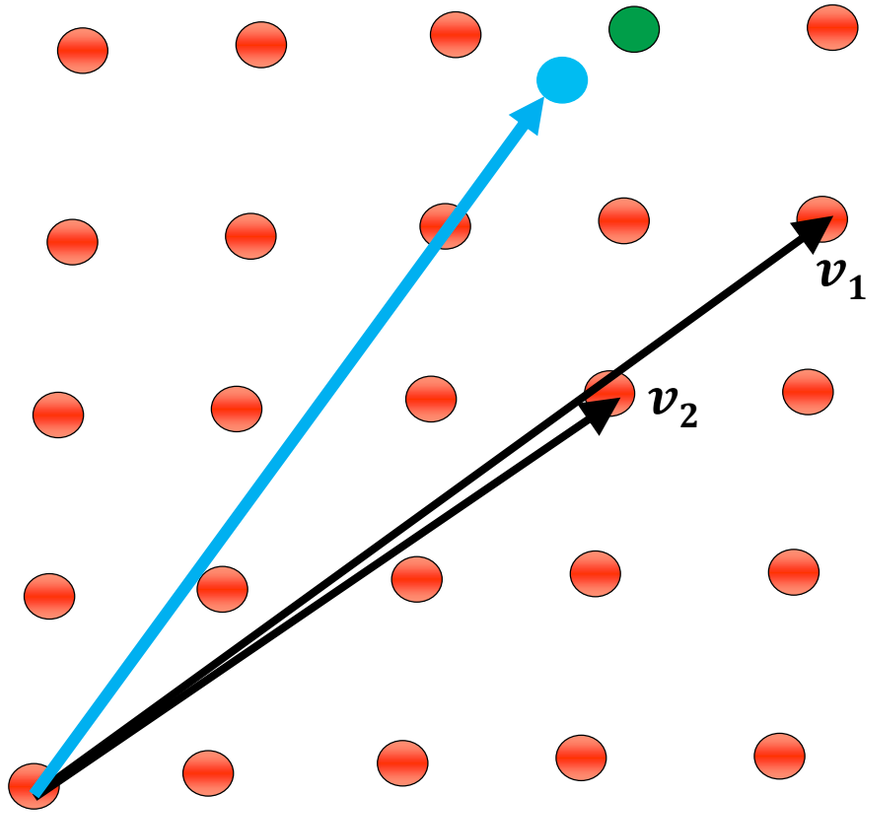
Hard Lattice Problem #1: Shortest Vector Problem (SVP_γ)



SVP_γ is NP-hard for $\gamma = O(1)$

SVP_γ is P for $\gamma = 2^{\Omega(n)}$

Hard Lattice Problem #2: Closest Vector Problem (CVP_d)



$\mathbf{CVP_d}$: Given lattice $L = \{v_1, v_2\}$ and target vector $\boldsymbol{v} \notin L$ within distance $\boldsymbol{d}$, find the closest lattice point

Why are they hard?

- Gaussian elimination for CVP? What about least-squares?
- Gram-Schmidt to reduce basis?

$$b_i^* \leftarrow b_i - \sum_{1 \le j \le i-1} \mu_{ij} \cdot b_j^*$$

$$\mu_{ij} = \frac{\langle b_i^*, b_j^* \rangle}{|b_j|^2}$$

- SVP_γ NP-hard for $\gamma = O(1)$: "at least as hard as the hardest problems in NP" (if $P \neq NP$, then no polynomial time alg.)

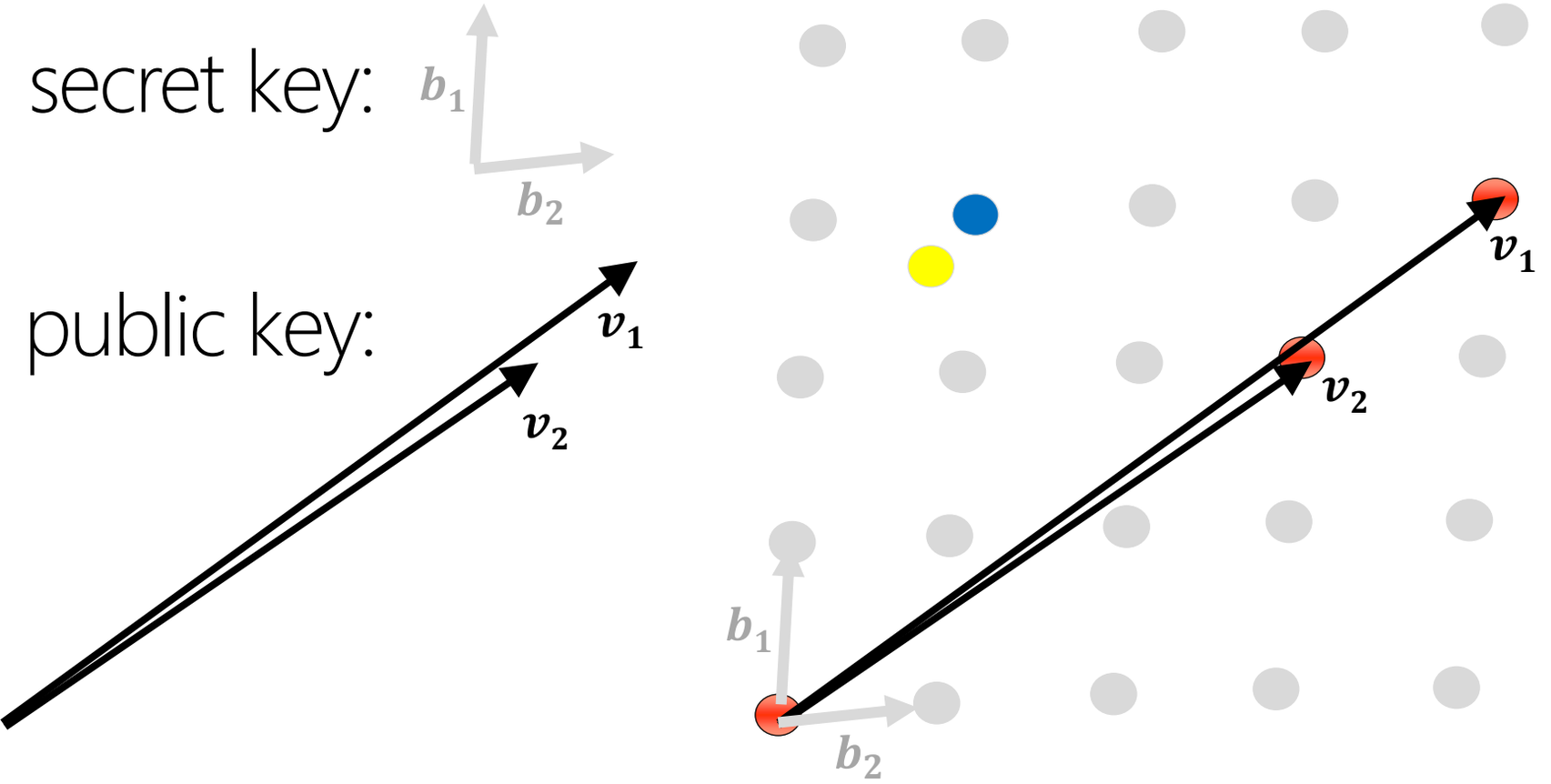
Play time…

e.g., GGH'97 signatures (≈ NTRUsign)

Idea: CVP is hard, but easy with good basis

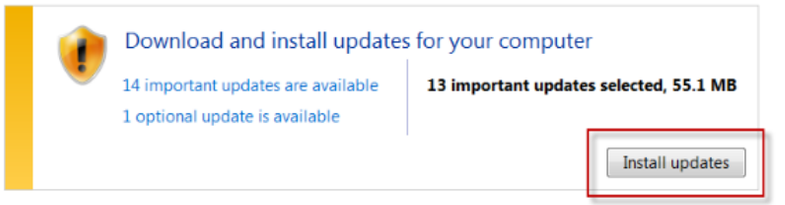
secret key: b_1 b_2

public key: v_1 v_2



message:

signature:



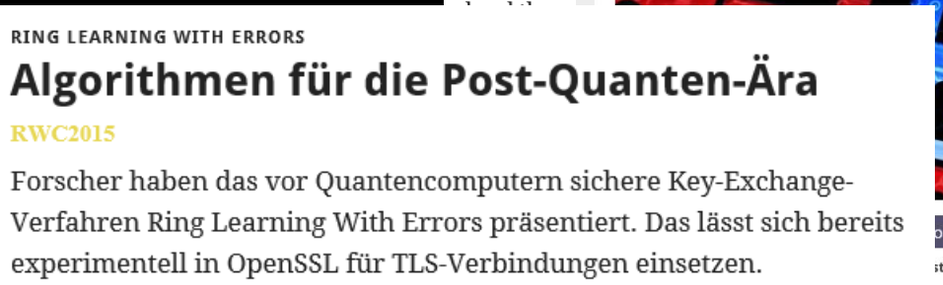
Security reductions

- GGH'97 (≈ right idea, but) did not come with a "security proof"
- If you can solve CVP, you can obviously forge messages, but this scheme was completely broken without solving CVP
- We want **Thm: "if you can forge signatures, you can solve CVP"**
- **Ajtai'96:** worst-to-average-case reduction unlocks lattice-based crypto "if you can break an average case, you can break the worst case"

Part 1: Recap and motivation

Part 2: Lattice basics

Part 3: PQ key exchange based on (R)LWE



Post-quantum key exchange for the TLS protocol from the ring learning with errors problem. IEEE Security & Privacy, 2015. DOI: 10.1109/SP.2015.40

joint work with
Joppe Bos (NXP), Michael Naehrig (MSR), Douglas Stebila (QUT)

NXP

Microsoft Research

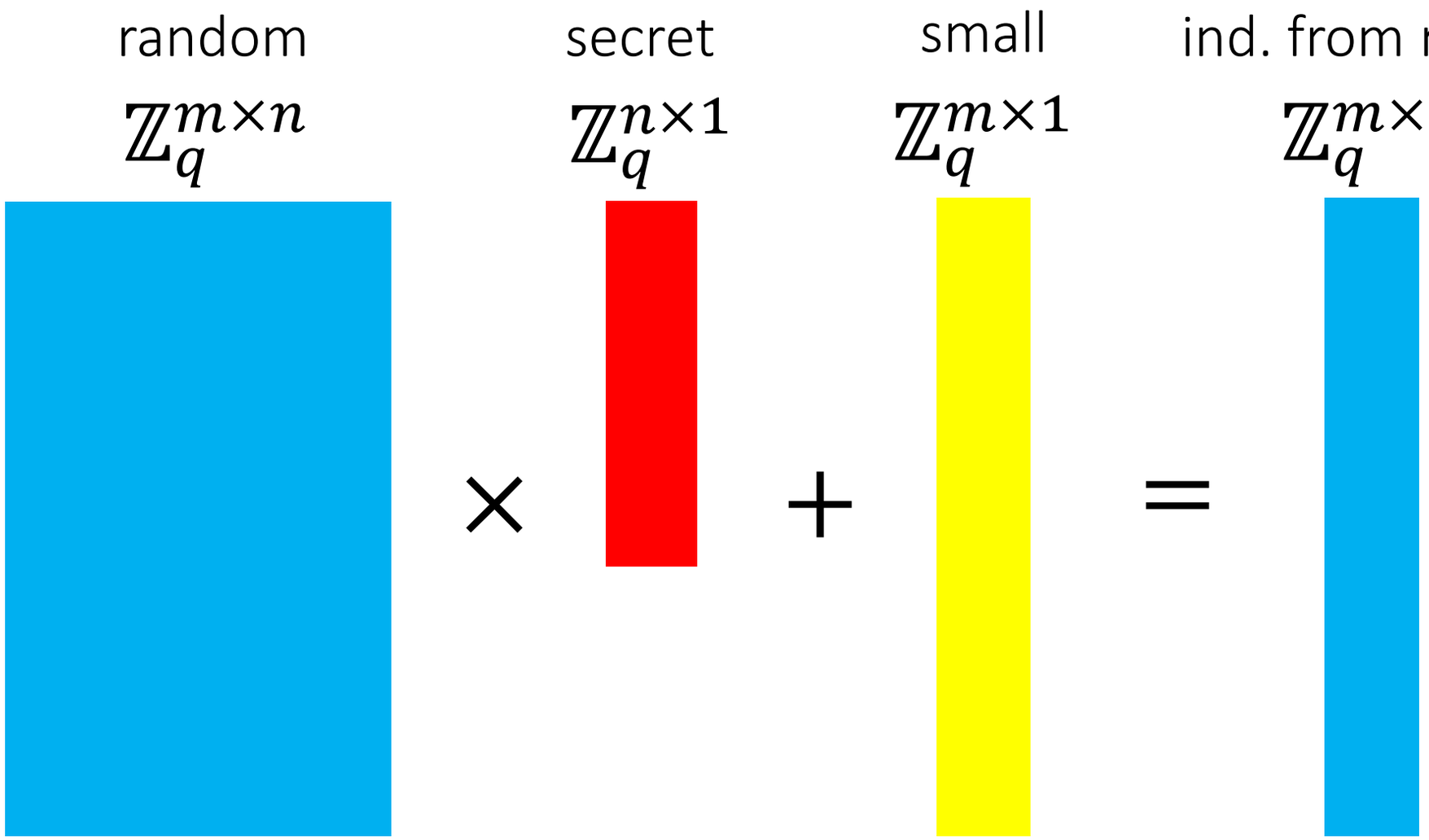
QUT

Regev'05

- Introduces the "Learning with Errors" (LWE) problem
- Uses it to construct LWE encryption
- Shows that breaking LWE implies (quantum) solving hard lattice problems (GapSVP and SIVP)

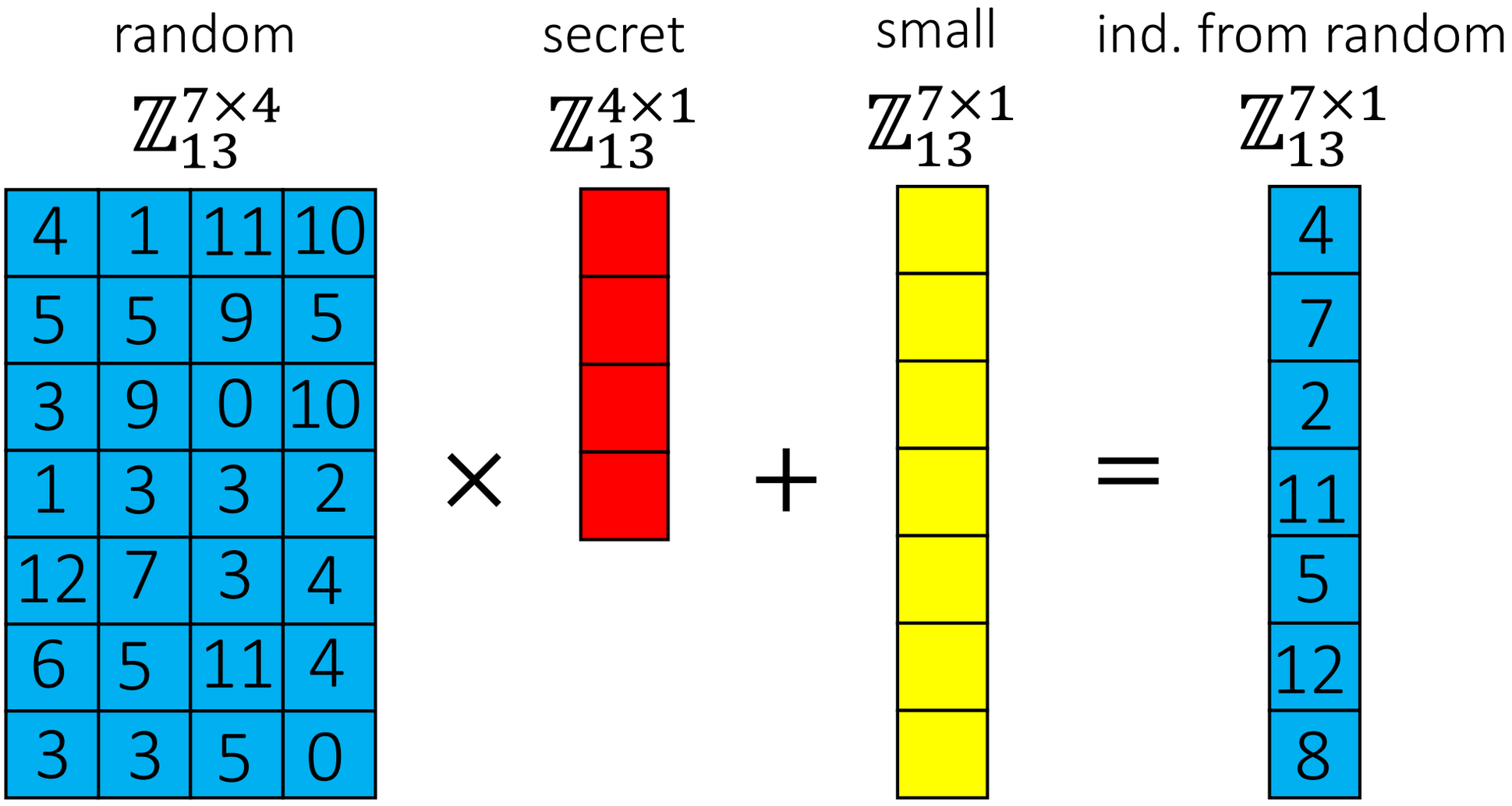
see his 2012 talk
http://research.microsoft.com/apps/video/default.aspx?id=166559

The learning with errors (LWE) problem



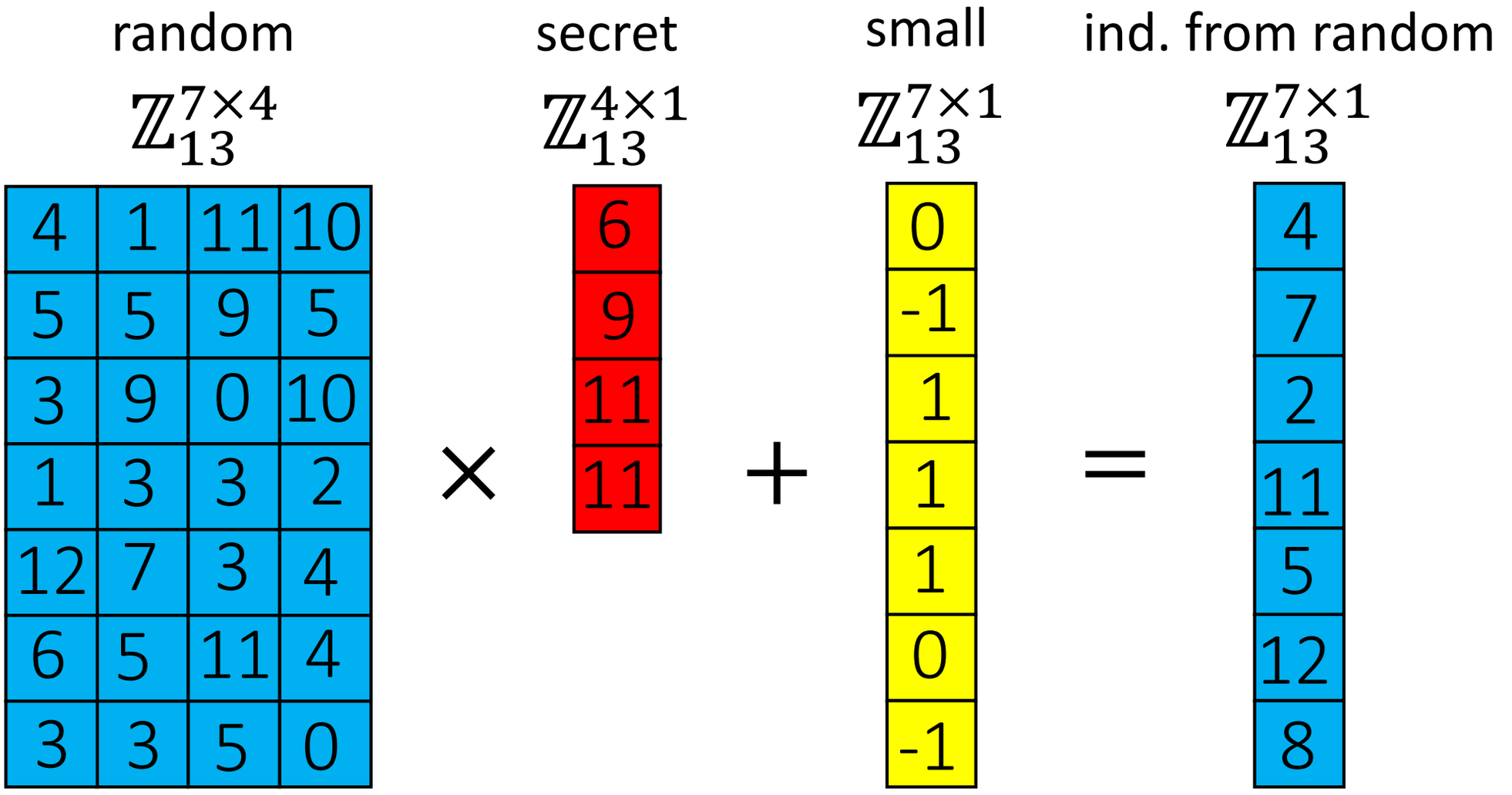
LWE problem: given blue, find (or just decide if ∃) red

The learning with errors (LWE) problem



LWE problem: given blue, find red

The learning with errors (LWE) problem



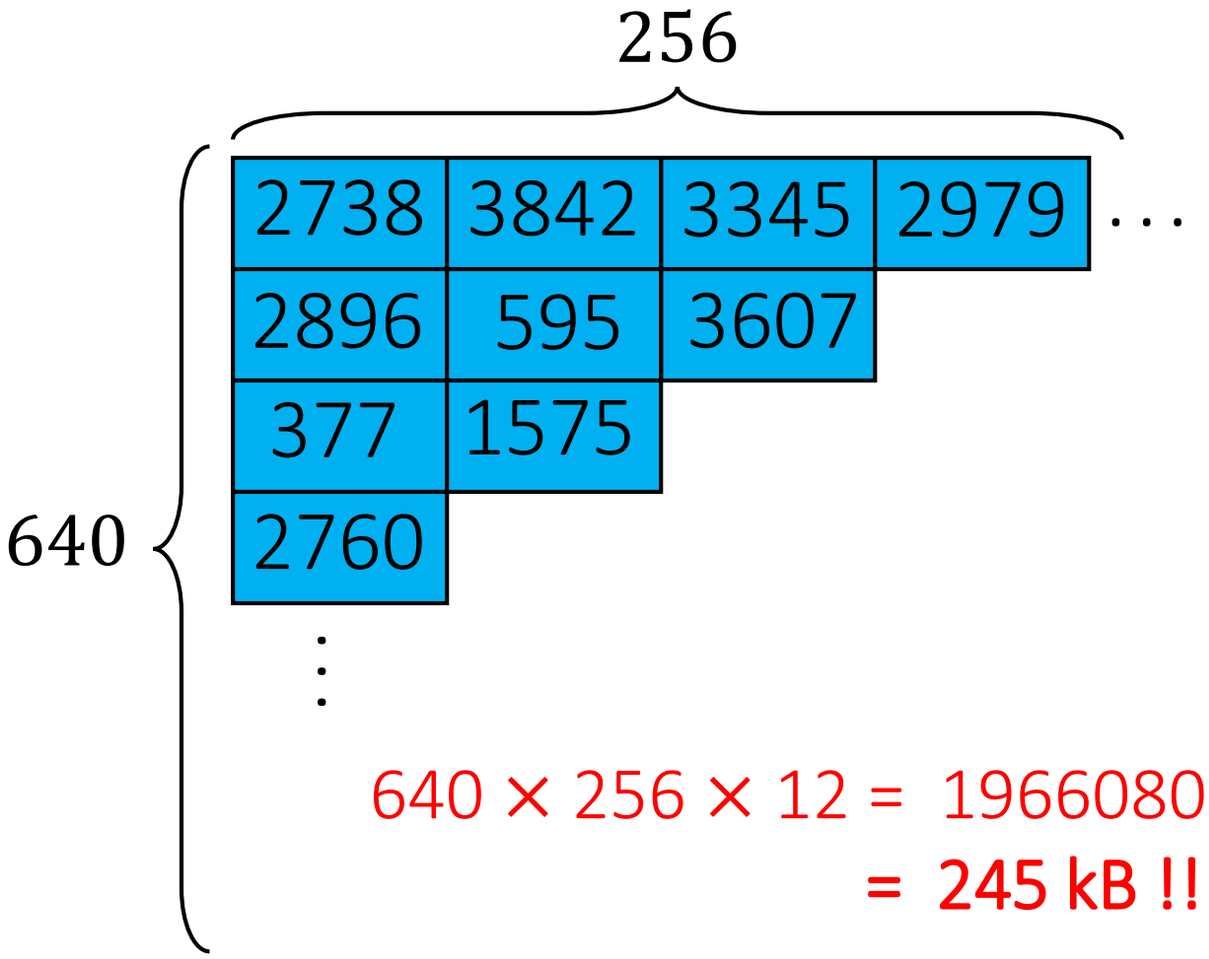
LWE problem: given blue, find red

Toy example versus real-world example

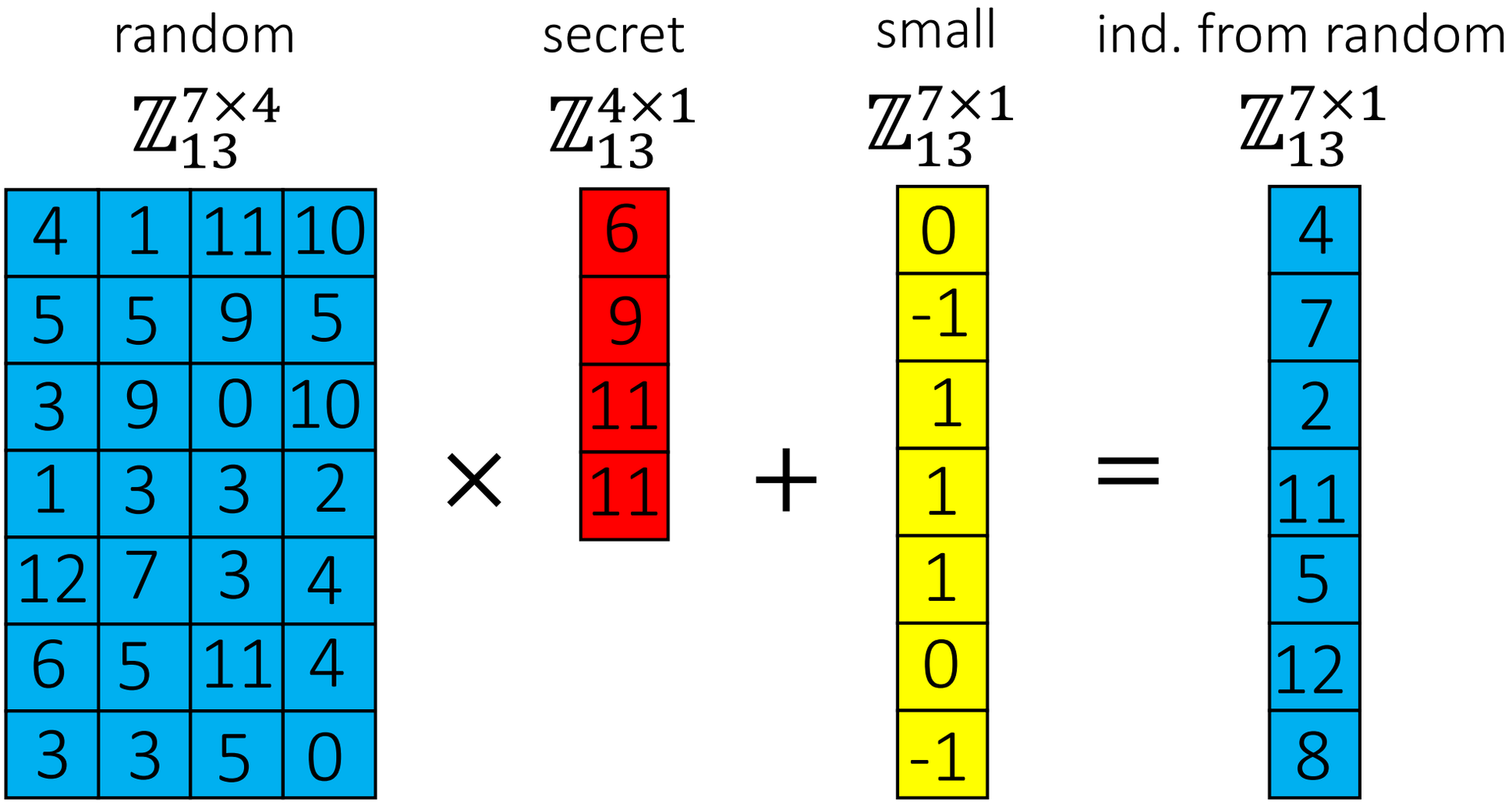
$\mathbb{Z}_{13}^{7\times4}$

4	1	11	10
5	5	9	5
3	9	0	10
1	3	3	2
12	7	3	4
6	5	11	4
3	3	5	0

$\mathbb{Z}_{4093}^{640\times256}$

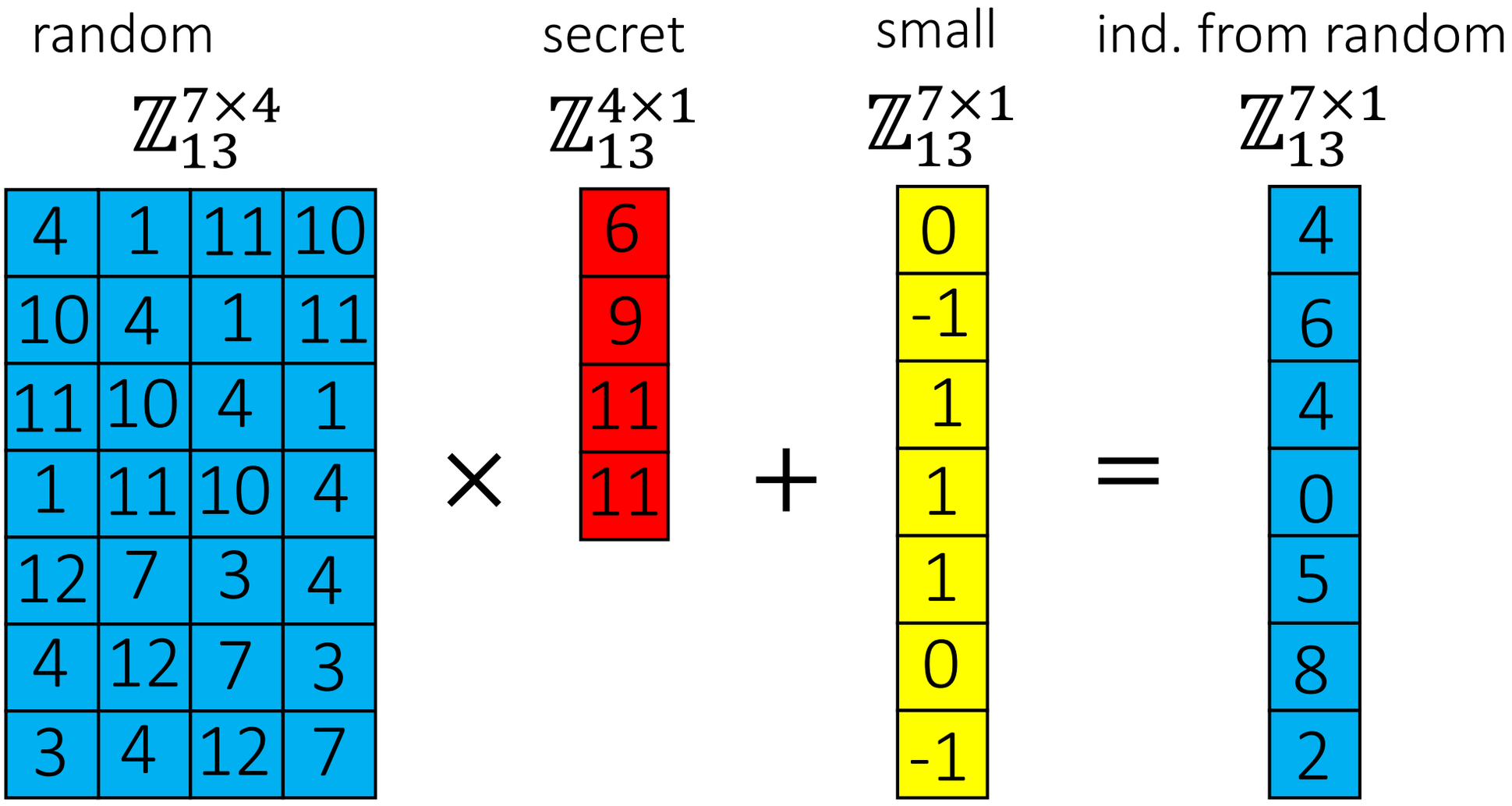


The learning with errors (LWE) problem



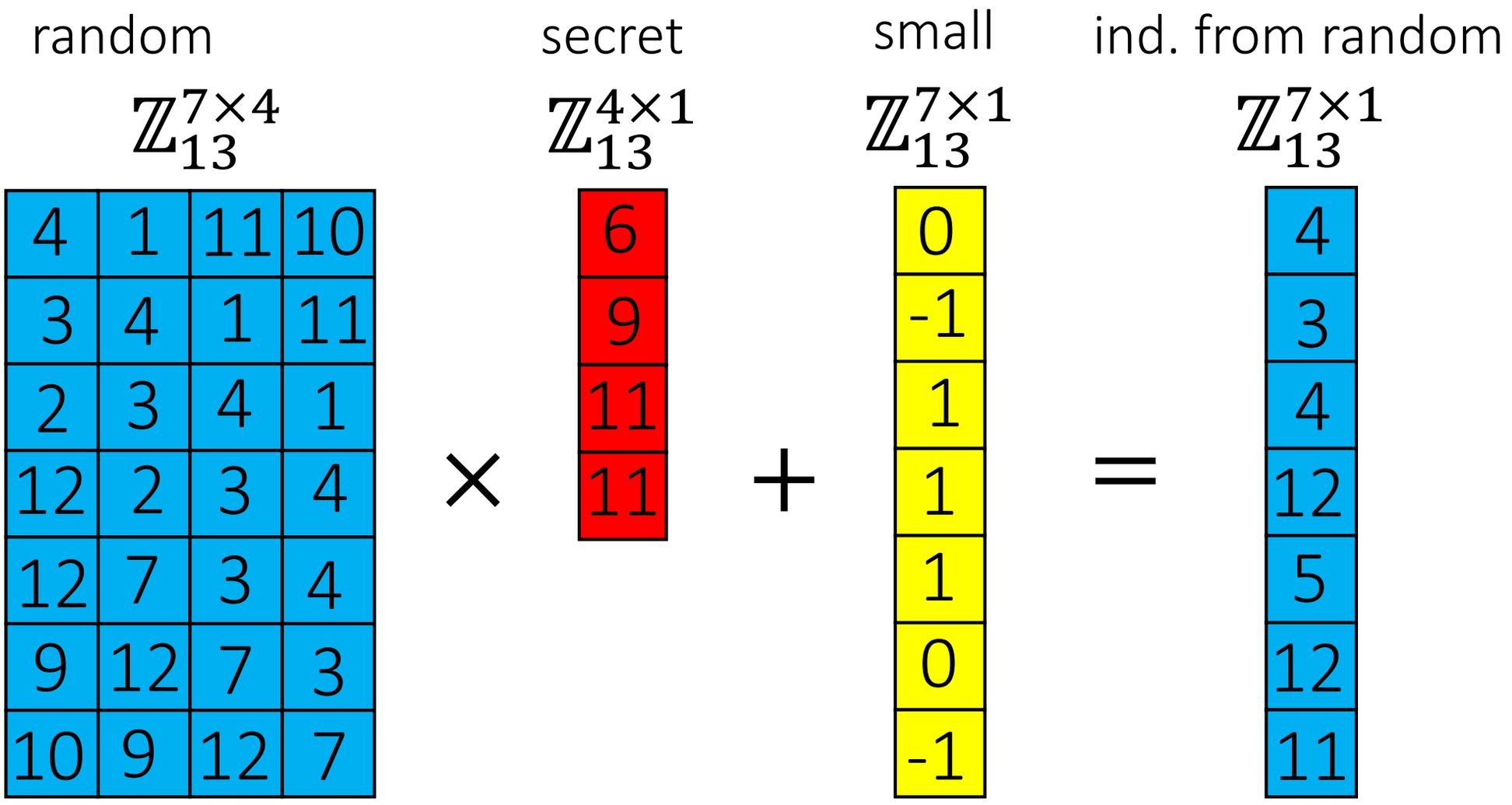
LWE problem: given blue, find red

The ring learning with errors (R-LWE) problem

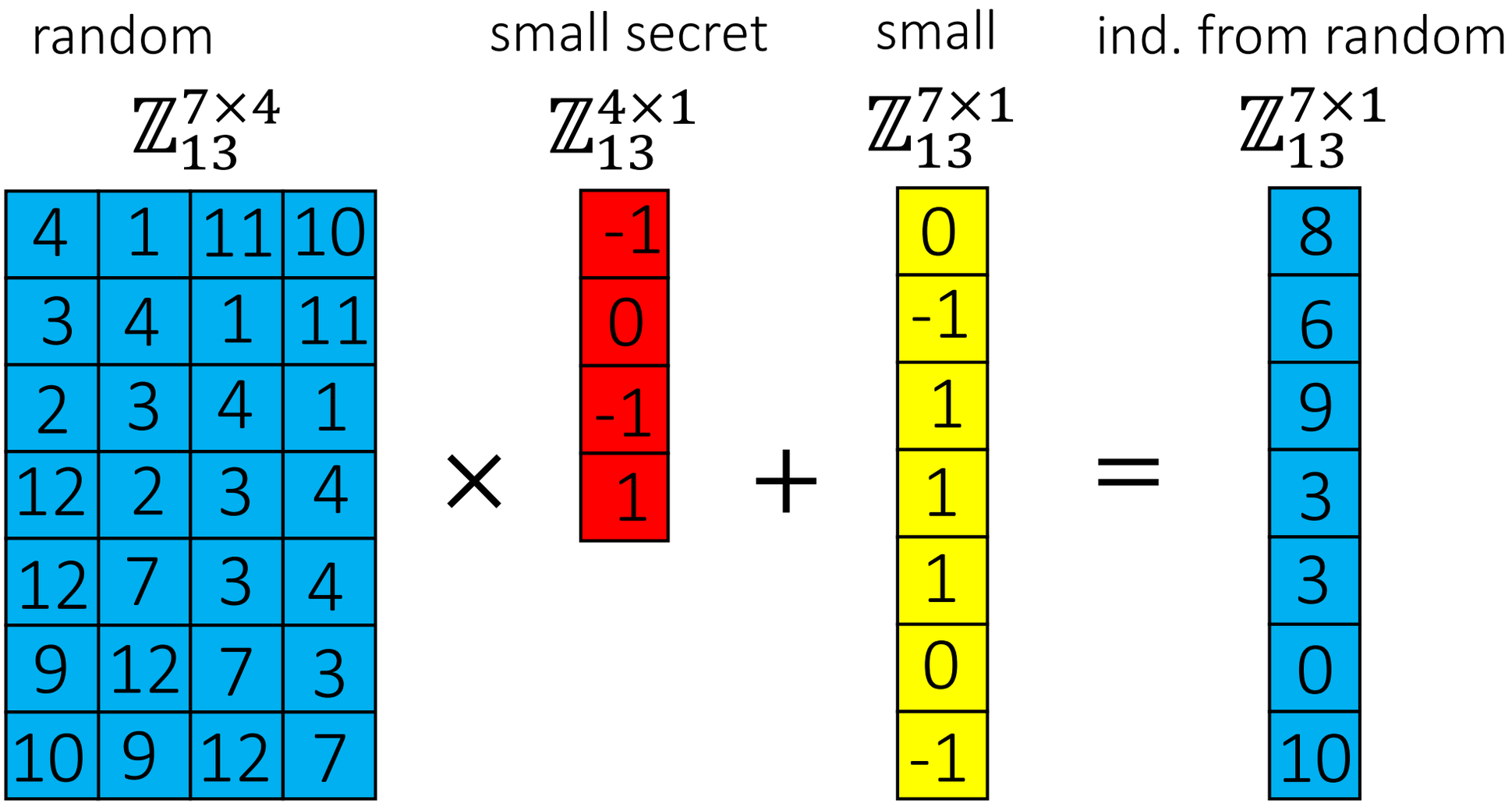


Lyubashevsky-Peikert-Regev ‘10: add ring structure

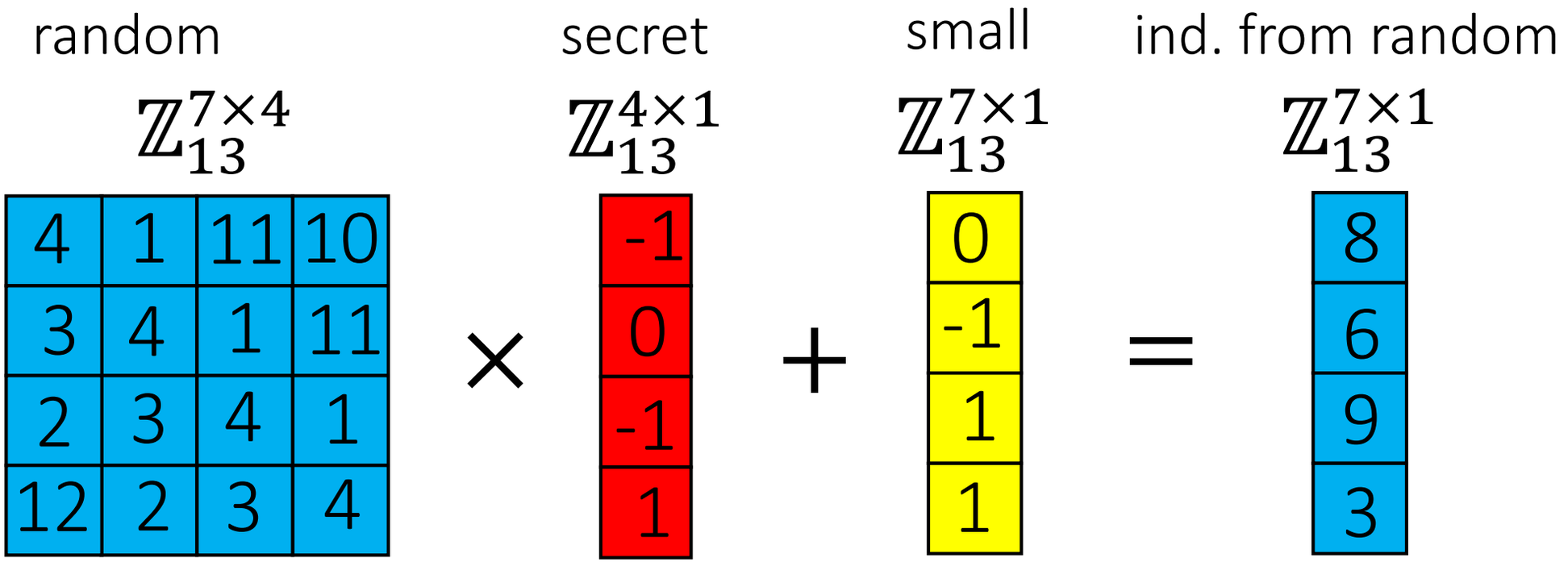
The ring learning with errors (R-LWE) problem



The ring learning with errors (R-LWE) problem



The ring learning with errors (R-LWE) problem



LWE problem: given blue, find red

The ring learning with errors (R-LWE) problem

$$\mathbb{Z}_{13}^{4\times 4} \longrightarrow \mathbb{Z}_{13}[x]/\langle x^4+1\rangle$$

4	1	11	10
3	4	1	11
2	3	4	1
12	2	3	4

$$\longrightarrow 4 + 1x + 11x^2 + 10x^3$$

$$\longrightarrow = x \cdot (4 + 1x + 11x^2 + 10x^3)$$

$$\longrightarrow = x^2 \cdot (4 + 1x + 11x^2 + 10x^3)$$

$$\longrightarrow = x^3 \cdot (4 + 1x + 11x^2 + 10x^3)$$

Ideal lattice: lattice modulo ideal

The ring learning with errors (R-LWE) problem

$$\begin{array}{rl} & \mathbf{4 + 1x + 11x^2 + 10x^3} \\ \times & \mathbf{-1 + 0x - 1x^2 + 1x^3} \\ + & \mathbf{0 - 1x + 1x^2 + 1x^3} \\ \hline & \mathbf{10 + 5x + 10x^2 + 7x^3} \\ \hline \end{array}$$

$$\frac{\mathbb{Z}_{13}[x]}{\langle x^4 + 1 \rangle}$$

R-LWE problem: given **blue**, find **red**

The ring learning with errors (R-LWE) problem (the 128-bit secure version)

$$\begin{array}{rl} & \mathbf{2792930407 + \cdots + 2938465015}x^{1023} \\ \times & \mathbf{5 - 3}\,x \ldots + \mathbf{9}x^{1022} - \mathbf{1}x^{1023} \\ + & \mathbf{2 + 4}\,x \ldots - \mathbf{0}x^{1022} + \mathbf{6}x^{1023} \\ \hline & \mathbf{3159804584 + \cdots + 1153769078}x^{1023} \\ \hline \end{array} \qquad \frac{\mathbb{Z}_{2^{32}-1}[x]}{\langle x^{1024}+1 \rangle}$$

R-LWE problem: given blue, find (small!) red

R-LWE-DH: key agreement in $R_q = \mathbb{Z}_q[x]/\langle x^n + 1 \rangle$

public: "big" a $\in R_q$

secret: "small" $e, s \in R_q$

$a \cdot s + e$

$a \cdot s' + e'$

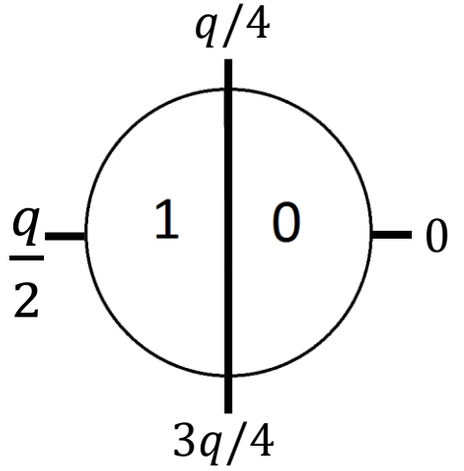
secret: "small" $e', s' \in R_q$

$\left(s \cdot (a \cdot s' + e')\right) \approx s \cdot a \cdot s'$

$\left(s' \cdot (a \cdot s + e)\right) \approx s \cdot a \cdot s'$

Approximate agreement mod q

the usual **ROUND** function



$$4079331841 + 1894732145 \cdot x + \cdots + 472608255 \cdot x^{1022} + 516748383 \cdot x^{1023}$$

$$\approx \quad \approx \quad \approx \quad \approx$$

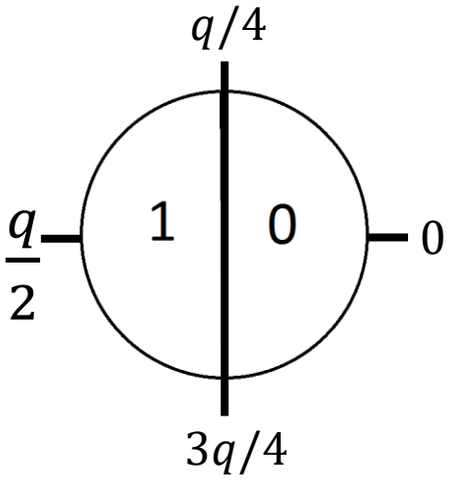
$$4079332556 + 1894733033 \cdot x + \cdots + 472607765 \cdot x^{1022} + 516748363 \cdot x^{1023}$$

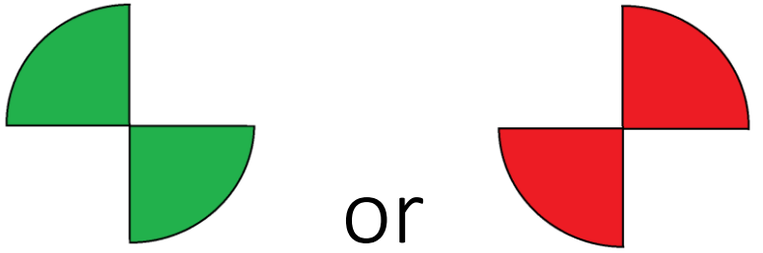
$$\| \quad \| \quad \| \quad \|$$

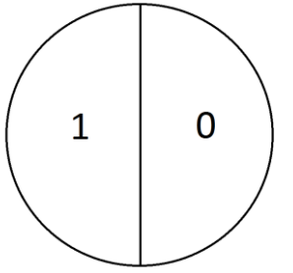
ROUND $\quad 0 \quad 1 \quad 0 \quad 0$

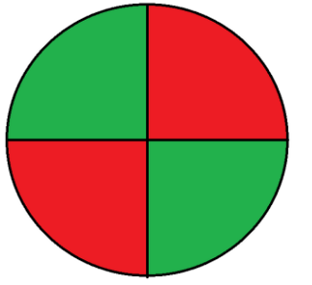
This will work most of the time (fails $\approx 1/2^{10}$), but we need **exact agreement**
i.e., what happens if one of the coefficients is in the **"danger zone(s)"**

Making approximate agreement exact in $\mathbb{Z}_q$

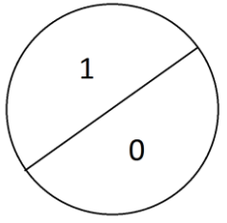




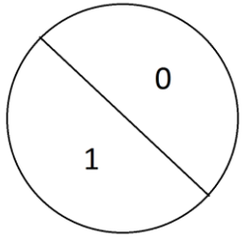




if



else



R-LWE-DH: exact key agreement

public: "big" a $\in R_q$

secret: "small" $e, s \in R_q$

$a \cdot s + e$

$a \cdot s' + e'$ and $\{$, $\}^n \in \{0,1\}^n$

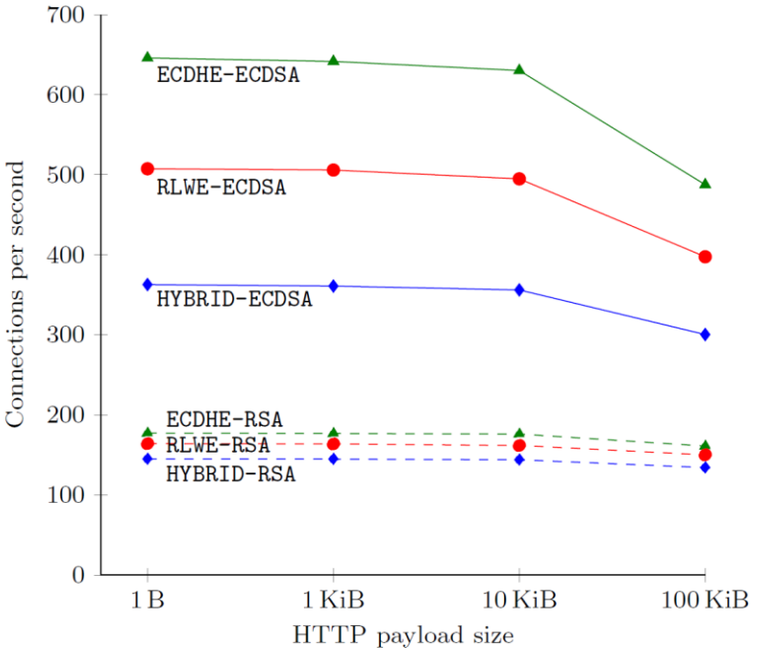
secret: "small" $e', s' \in R_q$

RECONCILE$(s \cdot (a \cdot s' + e'), \{$  ,  $\}^n)$ = ROUND$\big(s' \cdot (a \cdot s + e)\big)$

both parties now share $k \in \{0,1\}^n$

Summary

- [BCNS'15] developed and implemented key exchange protocol based on ring-LWE
- Proof of security based on decision-RLWE



- Plugged into TLS protocol and open sourced the software. Showed performance comparable to existing (non-PQ) stuff
- Subsequent work (by others and us) blows our performance out of the water, shows that RLWE-DH much faster than ECC/RSA/finite fields
- **Price of PQ paranoia: big keys**

Questions?