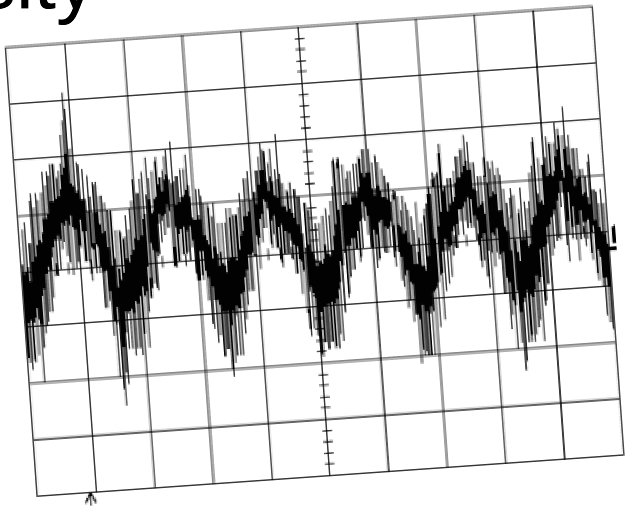
New Algorithms for Learning in Presence of Errors

Sanjeev Arora, Rong Ge

Princeton University



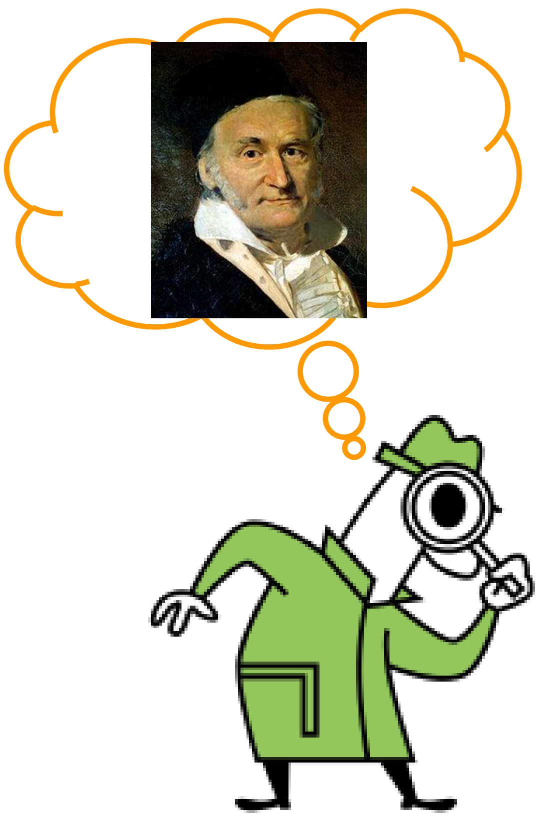
Hard(?) Problems

Treasure shall be found at u

$u \cdot (0,1,0,1,1) = 0$
$u \cdot (1,1,0,1,0) = 1$
$u \cdot (0,1,1,0,0) = 1$

…….

Hard(?) Problems

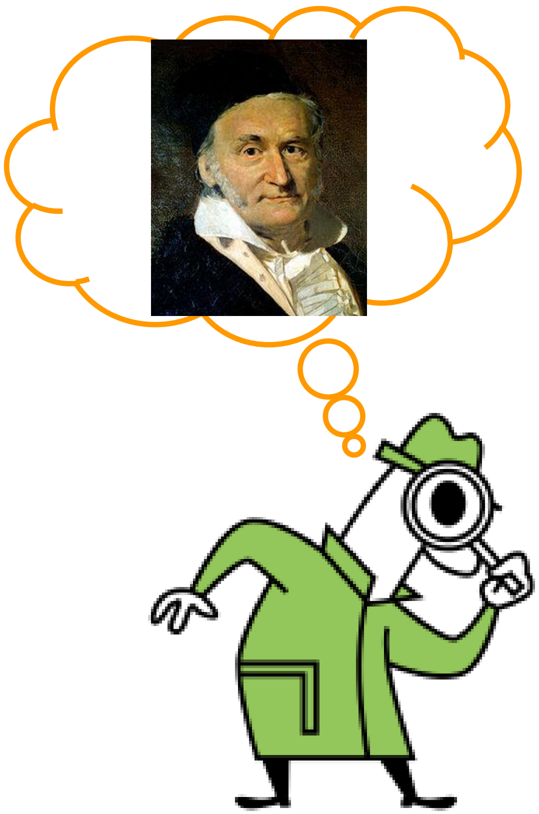


Treasure shall be found at u

$u \cdot (0,1,0,1,1) = 0$
$u \cdot (1,1,0,1,0) = 1$
$u \cdot (0,1,1,0,0) = 1$

……

Hard(?) Problems

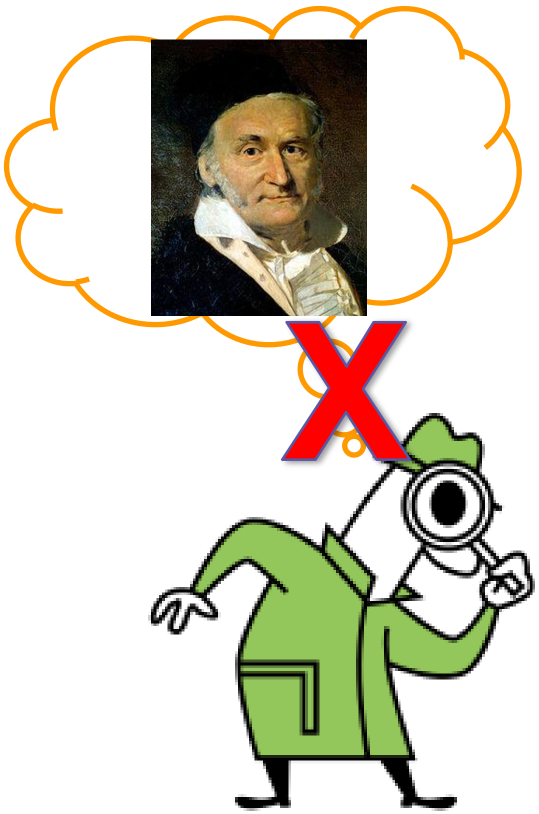


Treasure shall be found at u

u · (0,1,0,1,1) = 0
u · (1,1,0,1,0) = 1
u · (0,1,1,0,0) = 1
……

At least 90% of above are satisfied

Hard(?) Problems

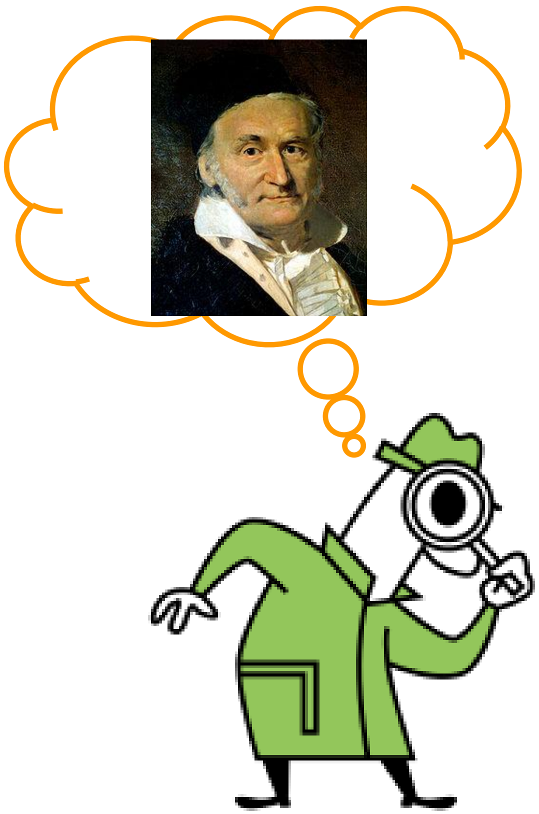


Treasure shall be found at u

u · (0,1,0,1,1) = 0
u · (1,1,0,1,0) = 1
u · (0,1,1,0,0) = 1
……

At least 90% of above are satisfied

Hard(?) Problems

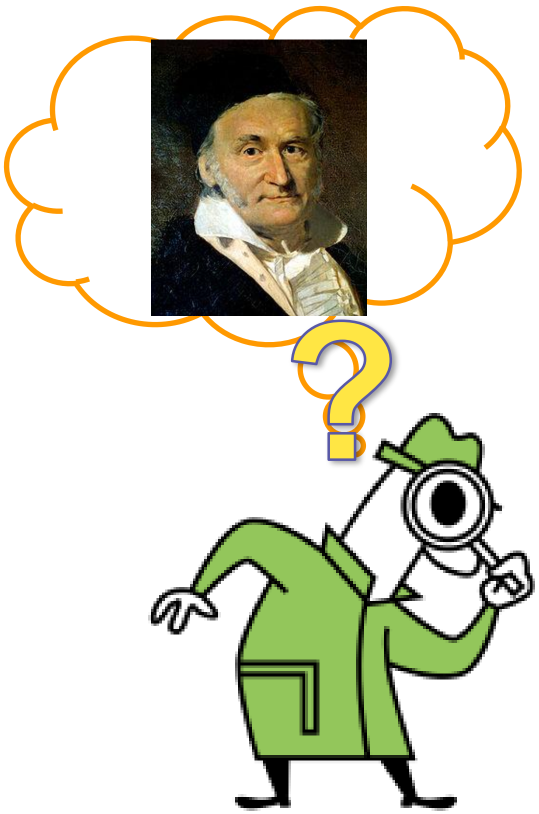


Treasure shall be found at u

u · (0,1,0,1,1) = 0
u · (1,1,0,1,0) = 1
u · (0,1,1,0,0) = 1
……

There might be 1 mistake in every 3 lines.

Hard(?) Problems



Treasure shall be found at u

$u \cdot (0,1,0,1,1) = 0$
$u \cdot (1,1,0,1,0) = 1$
$u \cdot (0,1,1,0,0) = 1$
……

There might be 1 mistake in every 3 lines.

Learning Parities with Noise

Learning Parities with Noise

Secret u = (1,0,1,1,1)

Learning Parities with Noise

Secret u = (1,0,1,1,1)

Learning Parities with Noise

Secret u = (1,0,1,1,1)

Learning Parities with Noise

Secret u = (1,0,1,1,1)

$u \cdot (0,1,0,1,1) = 0$

Learning Parities with Noise

Secret u = (1,0,1,1,1)

u · (0,1,0,1,1) = 0

u · (1,1,1,0,1) = 1

Learning Parities with Noise

Secret u = (1,0,1,1,1)

u · (0,1,0,1,1) = 0

u · (1,1,1,0,1) = 1

u · (0,1,1,1,0) = 1

Learning Parities with Noise

Learning Parities with Noise

- Secret vector u in $GF(2)^n$

Learning Parities with Noise

- Secret vector u in $GF(2)^n$
- Oracle returns random a and $b \approx u \cdot a$

Learning Parities with Noise

- Secret vector u in $GF(2)^n$
- Oracle returns random a and $b \approx u \cdot a$
- $u \cdot a$ is incorrect with probability p

Learning Parities with Noise

- Secret vector u in $GF(2)^n$
- Oracle returns random a and $b \approx u \cdot a$
- $u \cdot a$ is incorrect with probability p

- Best known algorithm: $2^{O(n/\log n)}$ [BKW'03]

Learning Parities with Noise

- Secret vector u in $GF(2)^n$
- Oracle returns random a and $b \approx u \cdot a$
- $u \cdot a$ is incorrect with probability p

- Best known algorithm: $2^{O(n/\log n)}$ [BKW'03]
- Used in designing public-key crypto [Alekhnovich'03]

Learning Parities with Structured Noise

Learning Parities with Structured Noise

Secret u = (1,0,1,1,1)

Learning Parities with Structured Noise

Secret u = (1,0,1,1,1)

Learning Parities with Structured Noise

Secret $u = (1,0,1,1,1)$

$u \cdot (0,1,0,1,1) = 0$
$u \cdot (1,1,0,1,0) = 1$
$u \cdot (0,1,1,0,0) = 1$

Learning Parities with Structured Noise

Secret u = (1,0,1,1,1)

u · (0,1,0,1,1) = 0

~~u · (1,1,0,1,0) = 1~~

u · (0,1,1,0,0) = 1

Learning Parities with Structured Noise

Learning Parities with Structured Noise

- Secret vector u

Learning Parities with Structured Noise

- Secret vector u
- Oracle returns random $a^1, a^2, \ldots, a^m$ and $b_1 \approx u \cdot a^1, b_2 \approx u \cdot a^2, \ldots, b_m \approx u \cdot a^m$

Learning Parities with Structured Noise

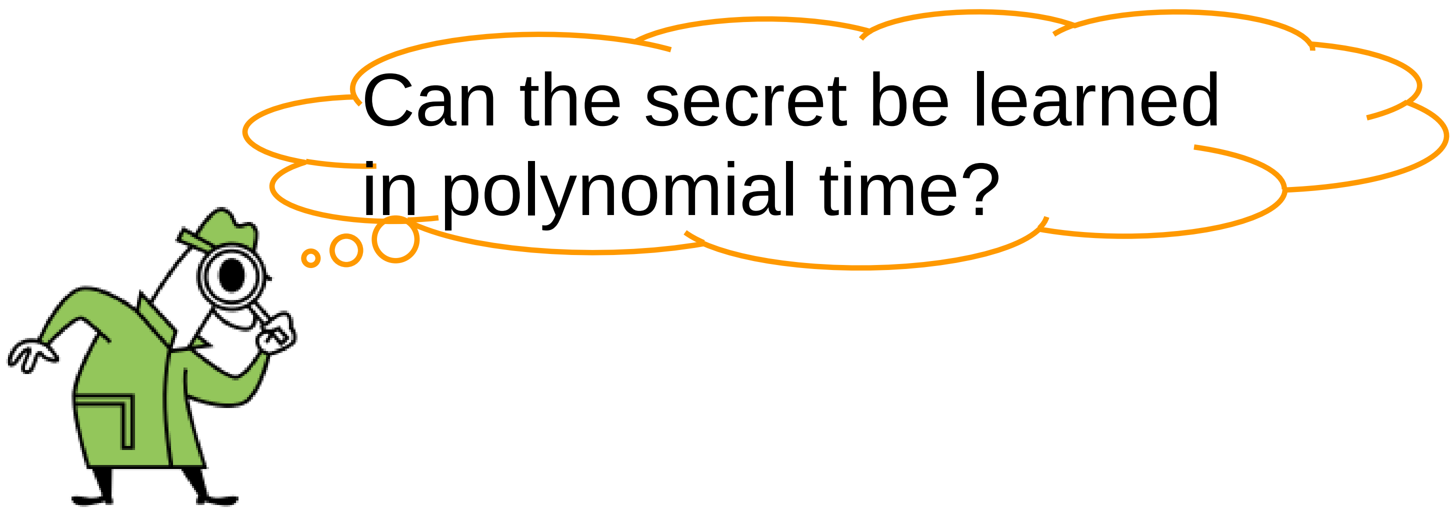
- Secret vector u
- Oracle returns random $a^1, a^2, \ldots, a^m$ and $b_1 \approx u \cdot a^1, b_2 \approx u \cdot a^2, \ldots, b_m \approx u \cdot a^m$
- "Not all inner-products are incorrect"

Learning Parities with Structured Noise

- Secret vector u
- Oracle returns random $a^1, a^2, \ldots, a^m$ and $b_1 \approx u \cdot a^1, b_2 \approx u \cdot a^2, \ldots, b_m \approx u \cdot a^m$
- The error has a certain structure

Learning Parities with Structured Noise

- Secret vector u
- Oracle returns random $a^1, a^2, \ldots, a^m$ and $b_1 \approx u \cdot a^1, b_2 \approx u \cdot a^2, \ldots, b_m \approx u \cdot a^m$
- The error has a certain structure



Our Results

Our Results

- Learning parities with structured noise
  - $n^{O(d)}$ time, adversarial noise

Our Results

- Learning parities with structured noise
  - $n^{O(d)}$ time, adversarial noise
- Learning With Errors
  - Subexp algorithm when noise < $n^{1/2}$
  - Open problem since [Regev'05]

Our Results

- Learning parities with structured noise
  - $n^{O(d)}$ time, adversarial noise
- Learning With Errors
  - Subexp algorithm when noise $< n^{1/2}$
  - Open problem since [Regev'05]
- Majority of 3 parities
  - Can inverse with $O(n^2 \log n)$ queries.
  - Pseudorandom generator purposed in [ABW'10]

Structures as Polynomials

Structures as Polynomials

- c_i=1 iff i-th inner-product is incorrect
  - $b_i = a^i \cdot u + c_i$

Structures as Polynomials

- c_i=1 iff i-th inner-product is incorrect
  - $b_i = a^i \cdot u + c_i$
- P(c) = 0 if an answer pattern is allowed

Structures as Polynomials

- c_i=1 iff i-th inner-product is incorrect
  - $b_i = a^i \cdot u + c_i$
- P(c) = 0 if an answer pattern is allowed

- “At least one of the inner-products is correct”
  - $P(c) = c_1 c_2 c_3 \ldots c_m = 0$

Structures as Polynomials

- c_i=1 iff i-th inner-product is incorrect
  - $b_i = a^i \cdot u + c_i$
- P(c) = 0 if an answer pattern is allowed

- "At least one of the inner-products is correct"
  - $P(c) = c_1c_2c_3\ldots c_m = 0$
- "No 3 consecutive wrong inner-products"
  - $P(c) = c_1c_2c_3+c_2c_3c_4+\ldots+c_{m-2}c_{m-1}c_m = 0$

Notations

- Subscripts are used for indexing vectors
  - u_i, c_i
- Superscripts are used for a list of vectors
  - a^i
- High dimensional vectors are indexed like $Z_{i,j,k}$
- a, b are known constants, u, c are unknown constants used in analysis, x, y, Z are variables in equations.

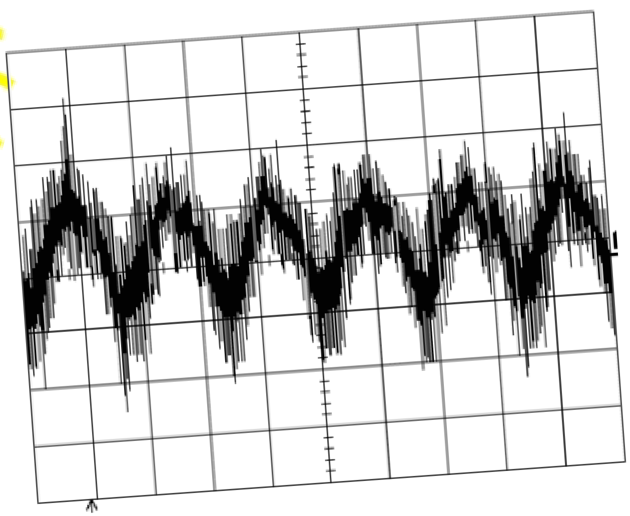
Main Result

Main Result

- For ANY non-trivial structure P of degree d, the secret can be learned using $n^{O(d)}$ queries and $n^{O(d)}$ time.

Main Result

- For ANY non-trivial structure P of degree d, the secret can be learned using $n^{O(d)}$ queries and $n^{O(d)}$ time.



The Algorithm

The Algorithm

- Query the Oracle

The Algorithm

- Query the Oracle
- Write out polynomial equations over x
  - Solution in mind: $x = u$

The Algorithm

- Query the Oracle
- Write out polynomial equations over x
  - Solution in mind: x = u
- Linearize all equations to get equations over y

The Algorithm

- Query the Oracle
- Write out polynomial equations over x
  - Solution in mind: $x = u$
- Linearize all equations to get equations over y
- Solve the equations for y (Gaussian Elimination)

The Algorithm

- Query the Oracle
- Write out polynomial equations over x
  - Solution in mind: $x = u$
- Linearize all equations to get equations over y
- Solve the equations for y (Gaussian Elimination)

- The unique solution will recover secret u

The Algorithm

- Query the Oracle
- Write out polynomial equations over x
  - Solution in mind: $x = u$
- Linearize all equations to get equations over y
- Solve the equations for y (Gaussian Elimination)

- The unique solution will recover secret u

Linearization

Linearization

Answers $b_i = c_i + a^i \cdot x$ $c_1 c_2 c_3 = 0$

Linearization

Answers

$$b_i = c_i + a^i \cdot x \qquad c_1 c_2 c_3 = 0$$

$$(a^1 \cdot x + b_1)(a^2 \cdot x + b_2)(a^3 \cdot x + b_3) = 0 (*)$$

Linearization

Answers $b_i = c_i + a^i \cdot x$ $c_1 c_2 c_3 = 0$

$$(a^1 \cdot x + b_1)(a^2 \cdot x + b_2)(a^3 \cdot x + b_3) = 0 (*)$$

Degree 3 polynomial over x

Linearization

Answers $b_i = c_i + a^i \cdot x$ $c_1 c_2 c_3 = 0$

$$(a^1 \cdot x + b_1)(a^2 \cdot x + b_2)(a^3 \cdot x + b_3) = 0 (*)$$

Degree 3 polynomial over x

Linearization $y_1 = x_1,\ y_2 = x_2, \ldots,\ y_{1,2} = x_1 x_2, \ldots,\ y_{1,2,3} = x_1 x_2 x_3$

Linearization

Answers $b_i = c_i + a^i \cdot x$ $c_1 c_2 c_3 = 0$

$$(a^1 \cdot x + b_1)(a^2 \cdot x + b_2)(a^3 \cdot x + b_3) = 0 (*)$$

Degree 3 polynomial over x

Linearization $y_1 = x_1, y_2 = x_2, \ldots, y_{1,2} = x_1 x_2, \ldots, y_{1,2,3} = x_1 x_2 x_3$

$$a^1{}_1 a^2{}_2 a^3{}_3 y_{1,2,3} + \ldots + b_1 b_2 b_3 = 0 \ (**)$$

Linearization

Answers $b_i = c_i + a^i \cdot x$ $c_1 c_2 c_3 = 0$

$$(a^1 \cdot x + b_1)(a^2 \cdot x + b_2)(a^3 \cdot x + b_3) = 0 (*)$$

Degree 3 polynomial over x

Linearization $y_1 = x_1,\ y_2 = x_2, \ldots,\ y_{1,2} = x_1 x_2, \ldots,\ y_{1,2,3} = x_1 x_2 x_3$

$$a^1{}_1 a^2{}_2 a^3{}_3 y_{1,2,3} + \ldots + b_1 b_2 b_3 = 0 \ (**)$$

Linear Equations over y

(**) = L((*))

Canonical Solution

Canonical Solution

- $(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$
  - Always satisfied when $x_i = u_i$

Canonical Solution

- $(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$
  - Always satisfied when $x_i = u_i$
- $a^1{}_1 a^2{}_2 a^3{}_3 y_{1,2,3}+\ldots+b_1 b_2 b_3 = 0$ (**)
  - Always satisfied when $y_1=u_1, y_2=u_2, \ldots, y_{1,2,3}=u_1 u_2 u_3$

Canonical Solution

- $(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$
  - Always satisfied when $x_i = u_i$
- $a^1{}_1 a^2{}_2 a^3{}_3 y_{1,2,3}+\ldots+b_1 b_2 b_3 = 0$ (**)
  - Always satisfied when $y_1=u_1,y_2=u_2,\ldots,y_{1,2,3}=u_1 u_2 u_3$
- Canonical Solution: $y_1=u_1,y_2=u_2,\ldots,y_{1,2,3}=u_1 u_2 u_3$

Canonical Solution

- $(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0$(*)
  - Always satisfied when $x_i = u_i$
- $a^1{}_1 a^2{}_2 a^3{}_3 y_{1,2,3} + \ldots + b_1 b_2 b_3 = 0$ (**)
  - Always satisfied when $y_1=u_1, y_2=u_2, \ldots, y_{1,2,3}=u_1u_2u_3$

- Canonical Solution: $y_1=u_1, y_2=u_2, \ldots, y_{1,2,3}=u_1u_2u_3$
- Coming up: This is the only solution to the system of linear equations

Proof Outline

Proof Outline

- Express (*) and (**) in a special form
  - Tensor-Expansion

Proof Outline

- Express (*) and (**) in a special form
  - Tensor-Expansion

- Change view: treat y as constants, a as variables

Proof Outline

- Express (*) and (**) in a special form
  - Tensor-Expansion
- Change view: treat y as constants, a as variables
- Pr[fix y sat. all equations] = extremely small

Proof Outline

- Express (*) and (**) in a special form
  - Tensor-Expansion
- Change view: treat y as constants, a as variables
- Pr[fix y sat. all equations] = extremely small
- Union bound over all “non-canonical” solutions.

Tensor-expansion

Tensor-expansion

$(a^1 \cdot x + b_1)(a^2 \cdot x + b_2)(a^3 \cdot x + b_3) = 0 (*)$

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

Tensor-expansion

$(a^1 \cdot x + b_1)(a^2 \cdot x + b_2)(a^3 \cdot x + b_3) = 0 (*)$

Problem: b depends on a

$(a^1 \cdot (x+u) + c_1)(a^2 \cdot (x+u) + c_2)(a^3 \cdot (x+u) + c_3) = 0$

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

$(a^1 \cdot (x+u)+c_1)(a^2 \cdot (x+u)+c_2)(a^3 \cdot (x+u)+c_3) = 0$

$(a^1 \cdot X+c_1)(a^2 \cdot X+c_2)(a^3 \cdot X+c_3) = 0$

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

$(a^1 \cdot (x+u)+c_1)(a^2 \cdot (x+u)+c_2)(a^3 \cdot (x+u)+c_3) = 0$

$(a^1 \cdot X+c_1)(a^2 \cdot X+c_2)(a^3 \cdot X+c_3) = 0$

a,c:numbers; X: variable

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

$(a^1 \cdot (x+u)+c_1)(a^2 \cdot (x+u)+c_2)(a^3 \cdot (x+u)+c_3) = 0$

$(a^1 \cdot X+c_1)(a^2 \cdot X+c_2)(a^3 \cdot X+c_3) = 0$

a,c:numbers; X: variable

$a^1 a^2 a^3 \cdot X^3+c_1 a^2 a^3 \cdot X^2+\ldots+c_1 c_2 c_3 = 0$

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

$(a^1 \cdot (x+u)+c_1)(a^2 \cdot (x+u)+c_2)(a^3 \cdot (x+u)+c_3) = 0$

$(a^1 \cdot X+c_1)(a^2 \cdot X+c_2)(a^3 \cdot X+c_3) = 0$

a,c:numbers; X: variable

$$a^1 a^2 a^3 \cdot X^3 = 0$$

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

$(a^1 \cdot (x+u)+c_1)(a^2 \cdot (x+u)+c_2)(a^3 \cdot (x+u)+c_3) = 0$

$(a^1 \cdot X+c_1)(a^2 \cdot X+c_2)(a^3 \cdot X+c_3) = 0$

a,c:numbers; X: variable

$$a^1 a^2 a^3 \cdot X^3 = 0$$

$(a^1 \cdot x)(a^2 \cdot x) = (a^1 \otimes a^2)(x \otimes x)$

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

$(a^1 \cdot (x+u)+c_1)(a^2 \cdot (x+u)+c_2)(a^3 \cdot (x+u)+c_3) = 0$

$(a^1 \cdot X+c_1)(a^2 \cdot X+c_2)(a^3 \cdot X+c_3) = 0$

a,c:numbers; X: variable

$$a^1 a^2 a^3 \cdot X^3 = 0$$

$(a^1 \cdot x)(a^2 \cdot x) = (a^1 \otimes a^2)(x \otimes x)$

Linearize

Let $Z^3_{i,j,k} = L((x_i+u_i)(x_j+u_j)(x_k+u_k))$

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

$(a^1 \cdot (x+u)+c_1)(a^2 \cdot (x+u)+c_2)(a^3 \cdot (x+u)+c_3) = 0$

$(a^1 \cdot X+c_1)(a^2 \cdot X+c_2)(a^3 \cdot X+c_3) = 0$

a,c:numbers; X: variable

$a^1 a^2 a^3 \cdot X^3 = 0$

$(a^1 \cdot x)(a^2 \cdot x) = (a^1 \otimes a^2)(x \otimes x)$

Linearize

Let $Z^3_{i,j,k} = L((x_i+u_i)(x_j+u_j)(x_k+u_k))$

$$(a^1 \otimes a^2 \otimes a^3)Z^3 = 0$$

Tensor-expansion

$(a^1 \cdot x+b_1)(a^2 \cdot x+b_2)(a^3 \cdot x+b_3) = 0(*)$

Problem: b depends on a

$(a^1 \cdot (x+u)+c_1)(a^2 \cdot (x+u)+c_2)(a^3 \cdot (x+u)+c_3) = 0$

$(a^1 \cdot X+c_1)(a^2 \cdot X+c_2)(a^3 \cdot X+c_3) = 0$

a,c:numbers; X: variable

$$a^1 a^2 a^3 \cdot X^3 = 0$$

$(a^1 \cdot x)(a^2 \cdot x) = (a^1 \otimes a^2)(x \otimes x)$

Linearize

Let $Z^3_{i,j,k} = L((x_i+u_i)(x_j+u_j)(x_k+u_k))$

$$(a^1 \otimes a^2 \otimes a^3) Z^3 = 0$$

$Z^3 = 0 \Leftrightarrow$ y is the canonical solution

Change View

$(a^1 \otimes a^2 \otimes a^3)Z^3 = 0$ Linear Equation over y variables

Change View

$$(a^1 \otimes a^2 \otimes a^3)Z^3 = 0$$ Linear Equation over y variables

$$Z^3(a^1 \otimes a^2 \otimes a^3) = 0$$

Change View

$$(a^1 \otimes a^2 \otimes a^3)Z^3 = 0$$ Linear Equation over y variables

$$Z^3(a^1 \otimes a^2 \otimes a^3) = 0$$ Polynomial over a’s

Change View

$(a^1 \otimes a^2 \otimes a^3)Z^3 = 0$ Linear Equation over y variables

$Z^3(a^1 \otimes a^2 \otimes a^3) = 0$ Polynomial over a's

Uniformly random

Change View

$$(a^1 \otimes a^2 \otimes a^3)Z^3 = 0$$ Linear Equation over y variables

$$Z^3(a^1 \otimes a^2 \otimes a^3) = 0$$ Polynomial over a's

Uniformly random

- Lemma
  - When $Z^3 \neq 0$ (y non-canonical), the equation is a non-zero polynomial over a's

Change View

$$(a^1 \otimes a^2 \otimes a^3) Z^3 = 0$$ Linear Equation over y variables

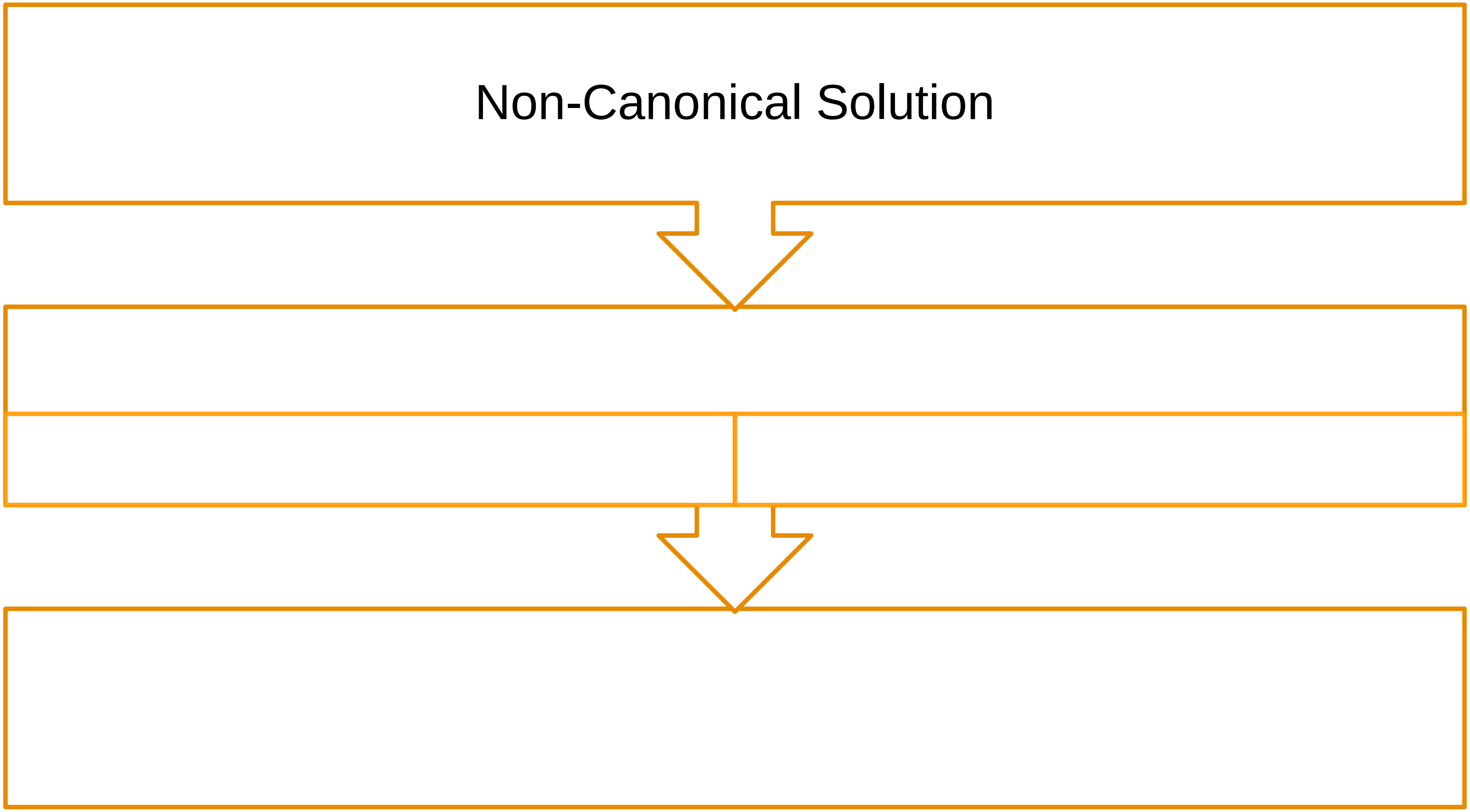
$$Z^3 (a^1 \otimes a^2 \otimes a^3) = 0$$ Polynomial over a's

Uniformly random

- Lemma
  - When $Z^3 \neq 0$ (y non-canonical), the equation is a non-zero polynomial over a's
- Schwartz-Zippel
  - The polynomial is non-zero w.p. at least 2^{-d}

Main Lemma → Theorem

Main Lemma ➔ Theorem



Main Lemma → Theorem

Non-Canonical Solution

↓

Non-zero Z^3 vector, Poly(a) = 0 for all equations

↓

Main Lemma → Theorem

Non-Canonical Solution

Non-zero Z^3 vector, Poly(a) = 0 for all equations

Schwartz-Zippel

Union Bound

Main Lemma → Theorem

Non-Canonical Solution

↓

Non-zero Z^3 vector, Poly(a) = 0 for all equations

Schwartz-Zippel | Union Bound

↓

Low Probability

Main Lemma → Theorem

Non-Canonical Solution

↓

Non-zero Z^3 vector, Poly(a) = 0 for all equations

Schwartz-Zippel | Union Bound

↓

With High Probability

Main Lemma → Theorem

No Non-Canonical Solutions

Non-zero Z^3 vector, F[illegible]) = 0 for all equations

Schwartz-Zippel	Union Bound

With High Probability

Main Lemma → Theorem

No Non-Canonical Solutions

Non-zero Z^3 vector, P[illegible]) = 0 for all equations

Schwartz-Zippel

Union Bound

With High Probability

□

Adversarial Noise

- Structure = “not all inner-products are incorrect”

Adversarial Noise

- Structure = “not all inner-products are incorrect”

  Secret u = (1,0,1,1,1)

Adversarial Noise

- Structure = “not all inner-products are incorrect”

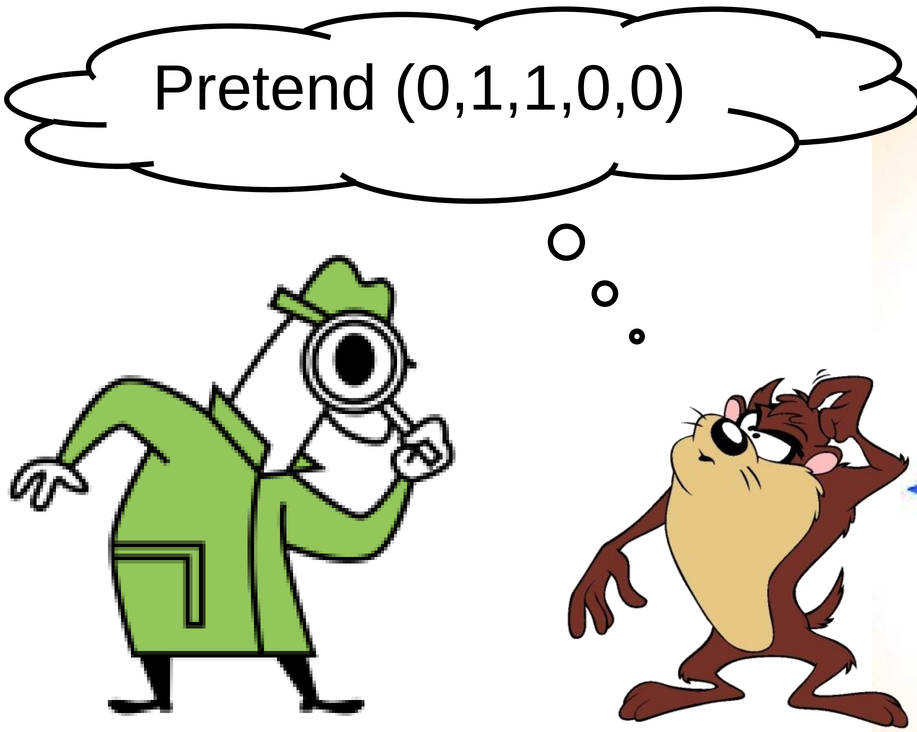
  Secret u = (1,0,1,1,1)

Adversarial Noise

- Structure = “not all inner-products are incorrect”

Secret u = (1,0,1,1,1)

u · (0,1,0,1,1) = 0 1 1
u · (1,1,0,1,0) = 0 0 1
u · (0,1,1,0,0) = 1 1 0



Adversarial Noise

- The adversary can fool ANY algorithm for some structures.

Adversarial Noise

- The adversary can fool ANY algorithm for some structures.

- Thm: If exists c that cannot be represented as
  $c = c^1+c^2$, $P(c^1)=P(c^2)=0$,
  the secret can be learned in $n^{O(m)}$ time
  otherwise no algorithm can learn the secret

Handling Adversarial Noise

Handling Adversarial Noise

- Compute polynomial R,
  $R(c) = 0 \Leftrightarrow c = c_1+c_2$, $P(c_1)=P(c_2)=0$

Handling Adversarial Noise

- Compute polynomial R,
  $R(c) = 0 \Leftrightarrow c = c_1+c_2, P(c_1)=P(c_2)=0$
- For each oracle answer (A,b), generate a group of oracle answers (A, b+c’) for all P(c’) = 0.

Handling Adversarial Noise

- Compute polynomial R,
  $R(c) = 0 \Leftrightarrow c = c_1+c_2, P(c_1)=P(c_2)=0$
- For each oracle answer (A,b), generate a group of oracle answers (A, b+c') for all P(c') = 0.

- Apply the white-noise algorithm

Handling Adversarial Noise

- C
  R

$$P = c_1c_2+c_2c_3+c_3c_1$$
$$R = c_1c_2c_3$$

- For each oracle answer (A,b), generate a group of oracle answers (A, b+c') for all P(c') = 0.
- Apply the white-noise algorithm

Handling Adversarial Noise

- C
  R

$$P = c_1c_2+c_2c_3+c_3c_1$$
$$R = c_1c_2c_3$$

- For each oracle answer (A,b), generate a group of oracle answers (A, b+c') for all P(c') = 0.

b = (1,0,1)
b = (0,0,1), (1,0,1), (1,1,1), (1,0,0)

- Apply the white-noise algorithm

Handling Adversarial Noise

- C
  R

$$P = c_1c_2+c_2c_3+c_3c_1$$
$$R = c_1c_2c_3$$

- For each oracle answer (A,b), generate a group of oracle answers (A, b+c') for all P(c') = 0.

b = (1,0,1)
b = (0,0,1), (1,0,1), (1,1,1), (1,0,0)

- Apply the white-noise algorithm

Canonical Solution: still satisfied
Non-Canonical: cannot be satisfied because noise c = (0,0,0) is always present

Learning With Errors

Learning With Errors

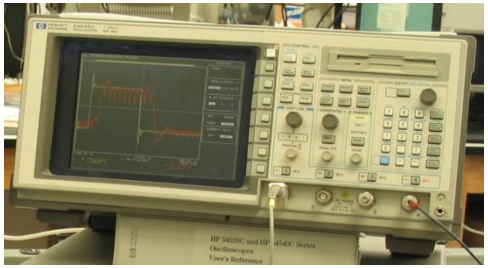
- Used in designing new crypto systems

Learning With Errors

- Used in designing new crypto systems
- Resistant to “side channel attacks”

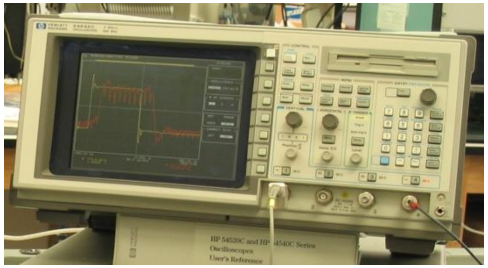
Learning With Errors

- Used in designing new crypto systems
- Resistant to “side channel attacks”



Learning With Errors

- Used in designing new crypto systems
- Resistant to “side channel attacks”



- Provable reduction from worst case lattice problems

Learning With Errors

Learning With Errors

- Secret u in Z_q^n

Learning With Errors

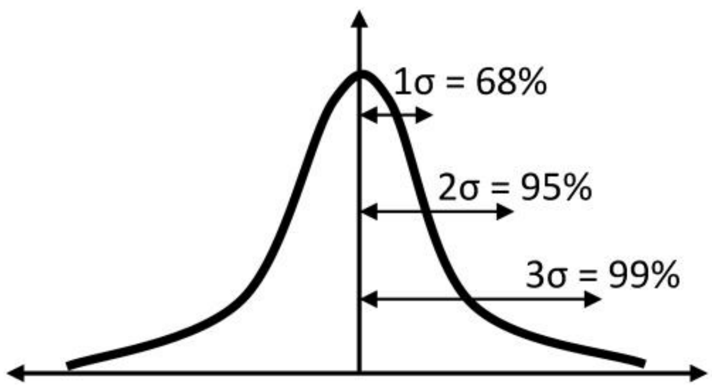
- Secret u in $\mathbb{Z}_q^n$
- Oracle returns random a and $a \cdot u + c$

Learning With Errors

- Secret u in $\mathbb{Z}_q^n$
- Oracle returns random a and $a \cdot u + c$
- c is chosen from Discrete Gaussian distribution with standard deviation δ

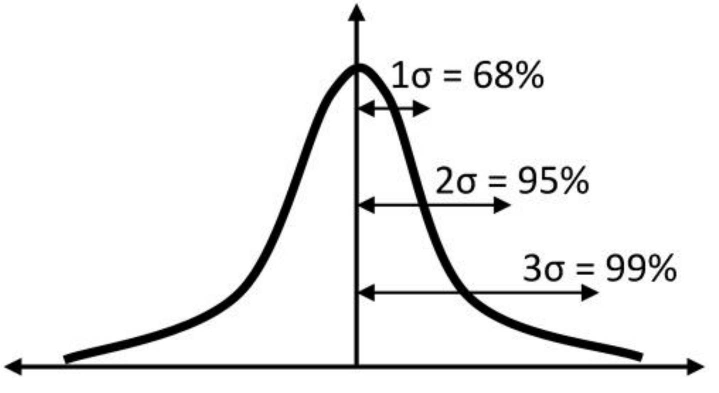
Learning With Errors

- Secret u in $\mathbb{Z}_q^n$
- Oracle returns random a and a·u+c
- c is chosen from Discrete Gaussian distribution with standard deviation δ



Learning With Errors

- Secret u in Z_q^n
- Oracle returns random a and a·u+c
- c is chosen from Discrete Gaussian distribution with standard deviation δ



- When $\delta = \Omega(n^{1/2})$ lattice problems can be reduced to LWE [Regev09]

Learning With Structured Errors

Learning With Structured Errors

- Structure specifies a set of possible errors
  - e.g. $|c| < \delta^2$
  - Still represented using polynomial $P(c) = 0$

Learning With Structured Errors

- Structure specifies a set of possible errors
  - e.g. $|c| < \delta^2$
  - Still represented using polynomial $P(c) = 0$

- Thm: When the polynomial has degree $d < q/4$, the secret can be learned in $n^{O(d)}$ time.

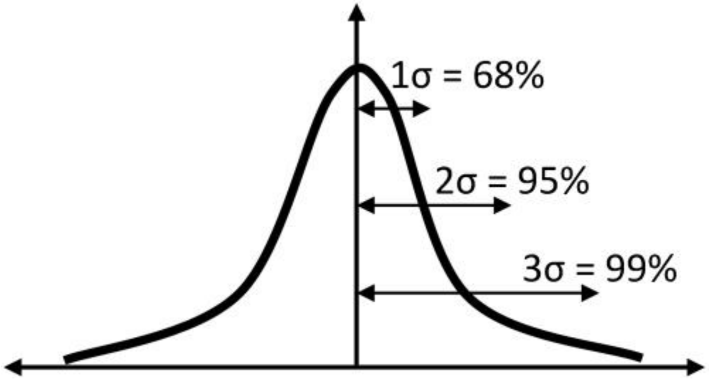
Learning With Structured Errors

- Structure specifies a set of possible errors
  - e.g. $|c| < \delta^2$
  - Still represented using polynomial $P(c) = 0$

- Thm: When the polynomial has degree $d < q/4$, the secret can be learned in $n^{O(d)}$ time.

- Cor: When $\delta = o(n^{1/2})$, LWE has a sub-exponential time algorithm

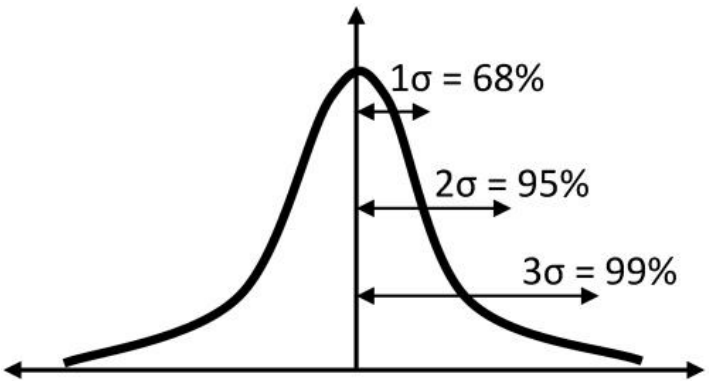
Thm → Cor



Thm → Cor

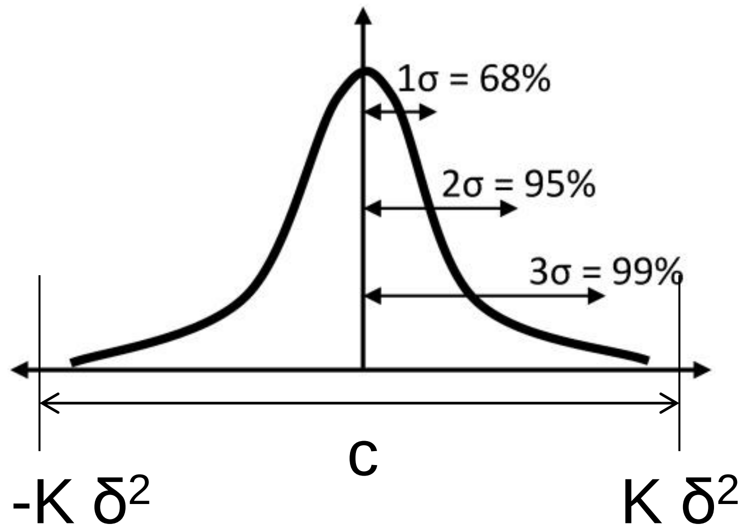
Structure:

$|c| < K \delta^2$



Thm → Cor

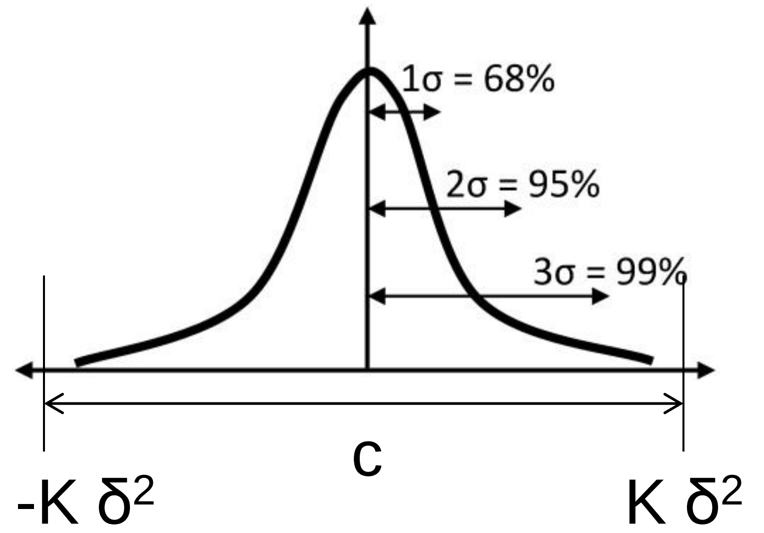
Structure:
$|c| < K\,\delta^2$



Thm → Cor

Structure:
$|c| < K\,\delta^2$

In LWE:
$\Pr[|c| > K\,\delta^2] < \exp(-O(K^2\delta^2))$

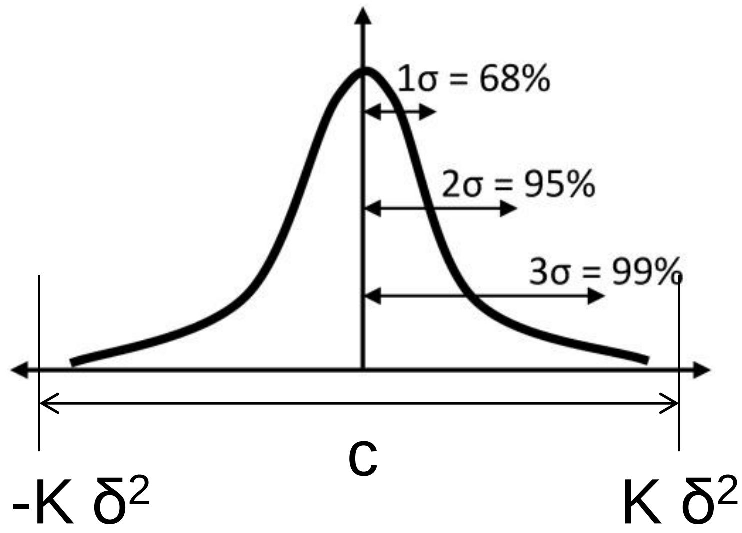


Thm → Cor

Structure:
$|c| < K\ \delta^2$

In LWE:
$\Pr[|c| > K\ \delta^2] <$ $\exp(-O(K^2\delta^2))$

# of equations:
$n^{\wedge}(O(K\ \delta^2)) <$ $\exp(O(K^2\delta^2))$



$K = 100 \log n$

Thm ➔ Cor

Structure:

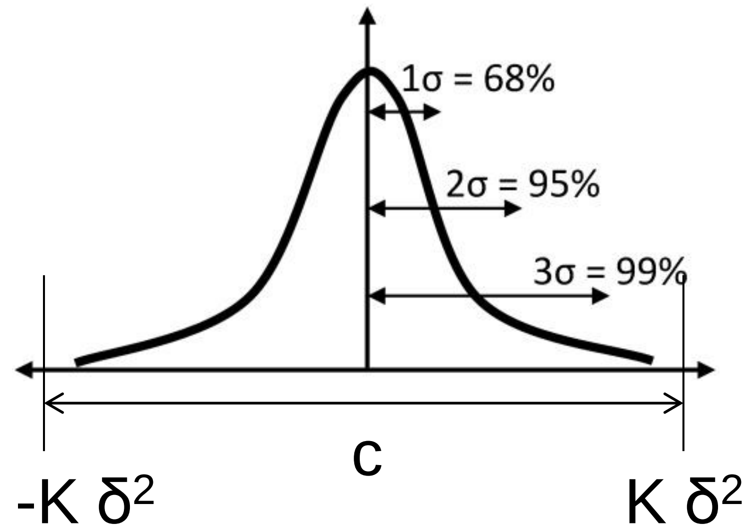
$|c| < K\,\delta^2$

In LWE:

$\Pr[|c| > K\,\delta^2] <$ $\exp(-O(K^2\delta^2))$

# of equations:

$n^{(O(K\,\delta^2))} <$ $\exp(O(K^2\delta^2))$



$K = 100 \log n$

Thm → Cor

Structure:

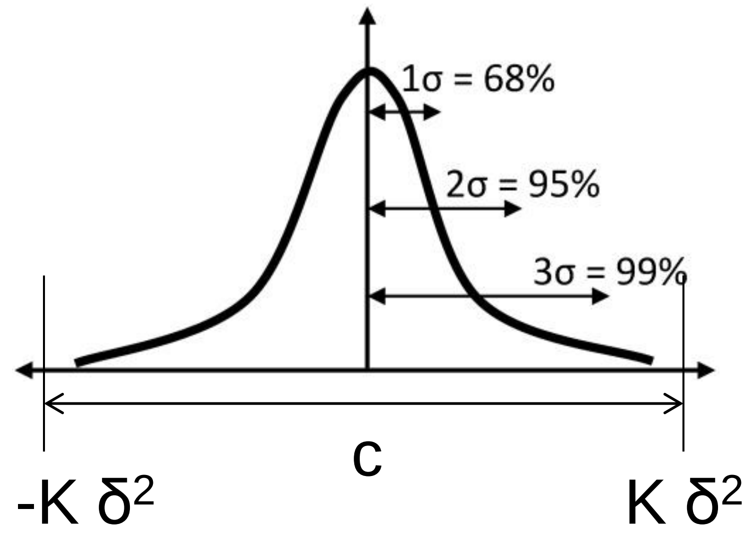
$|c| < K\,\delta^2$

In LWE:

$\Pr[|c|>K\,\delta^2] <$ $\exp(-O(K^2\delta^2))$

# of equations:

$n^{(O(K\,\delta^2))} <$ $\exp(O(K^2\delta^2))$



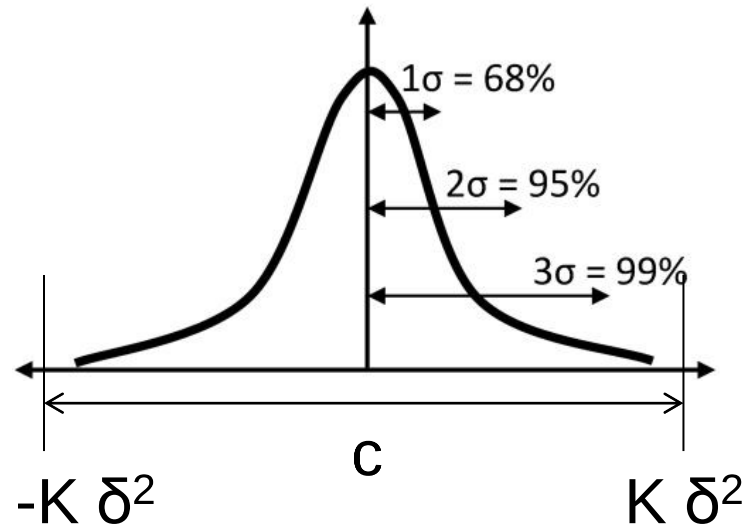
$K = 100 \log n$

Thm ➔ Cor

Structure:
$|c| < K\,\delta^2$

In LWE:
$\Pr[|c| > K\,\delta^2] <$ $\exp(-O(K^2\delta^2))$

# of equations:
$n^{(O(K\,\delta^2))} <$ $\exp(O(K^2\delta^2))$



$K = 100 \log n$

Negligible difference between LWE and LWSE,
Algorithm still success with high probability

Open Problems

- Non-trivial algorithm for the original model using linearization
- Possible lower bound for special kind of linear equation systems
- Improve the algorithm for learning with errors?

Thank You

Questions?