FrodoKEM
practical quantum-secure key encapsulation from generic lattices

Erdem Alkim   Joppe W. Bos   Léo Ducas   Patrick Longa

Ilya Mironov   Michael Naehrig   Valeria Nikolaenko

Chris Peikert   Ananth Raghunathan   Douglas Stebila

FrodoKEM

FrodoKEM's security derives from plain *Learning With Errors*
on algebraically unstructured lattices,
parameterized cautiously to avoid known risk categories,
and to conform to a worst-case/average-case reduction.

FrodoKEM

FrodoKEM's security derives from plain *Learning With Errors*
on algebraically unstructured lattices,
parameterized cautiously to avoid known risk categories,
and to conform to a worst-case/average-case reduction.

FrodoKEM

FrodoKEM's security derives from plain *Learning With Errors*
on algebraically unstructured lattices,
parameterized cautiously to avoid known risk categories,
and to conform to a worst-case/average-case reduction.

FrodoKEM

FrodoKEM's security derives from plain *Learning With Errors*
on algebraically unstructured lattices,
parameterized cautiously to avoid known risk categories,
and to conform to a worst-case/average-case reduction.

FrodoPKE (IND-CPA) $\xrightarrow[\text{(generic transform)}]{\text{[FujisakiOkamoto'99,HHK'17]}}$ FrodoKEM (IND-CCA)

FrodoKEM

FrodoKEM's security derives from plain *Learning With Errors*
on algebraically unstructured lattices,
parameterized cautiously to avoid known risk categories,
and to conform to a worst-case/average-case reduction.

FrodoPKE (IND-CPA) $\xrightarrow[\text{(generic transform)}]{\text{[FujisakiOkamoto'99,HHK'17]}}$ FrodoKEM (IND-CCA)

Concrete Instantiations

1. FrodoKEM-640: targets Level 1 security ($\geq$ AES-128)
2. FrodoKEM-976: targets Level 3 security ($\geq$ AES-192)
3. FrodoKEM-1344 (new, round 2): Level 5 security ($\geq$ AES-256)

Pedigree

Learning With Errors (LWE) [Regev'05]

- Lineage of [Ajtai'96,AjtaiDwork'97]: worst-case/average-case reductions:

Pedigree

Learning With Errors (LWE) [Regev'05]

- Lineage of [Ajtai'96,AjtaiDwork'97]: worst-case/average-case reductions: breaking random inputs $\Longrightarrow$ solving famous problems on any lattice.

Pedigree

Learning With Errors (LWE) [Regev'05]

- Lineage of [Ajtai'96,AjtaiDwork'97]: worst-case/average-case reductions: breaking random inputs $\Longrightarrow$ solving famous problems on any lattice.

> *"[This] assures us that attacks on the cryptographic construction are likely to be effective only for small choices of parameters and not asymptotically. In other words . . . there are no fundamental flaws in the design of our cryptographic construction." [MicciancioRegev'09]*

Pedigree

Learning With Errors (LWE) [Regev'05]

- Lineage of [Ajtai'96,AjtaiDwork'97]: worst-case/average-case reductions: breaking random inputs $\Longrightarrow$ solving famous problems on any lattice.

  > *"[This] assures us that attacks on the cryptographic construction are likely to be effective only for small choices of parameters and not asymptotically. In other words . . . there are no fundamental flaws in the design of our cryptographic construction."* *[MicciancioRegev'09]*

- LWE has been heavily used and cryptanalyzed by countless works.

Pedigree

Learning With Errors (LWE) [Regev'05]

- Lineage of [Ajtai'96,AjtaiDwork'97]: worst-case/average-case reductions: breaking random inputs $\Longrightarrow$ solving famous problems on any lattice.

  > *"[This] assures us that attacks on the cryptographic construction are likely to be effective only for small choices of parameters and not asymptotically. In other words . . . there are no fundamental flaws in the design of our cryptographic construction."* *[MicciancioRegev'09]*

- LWE has been heavily used and cryptanalyzed by countless works.

Public-Key Encryption/Key Exchange

- Many schemes with tight (CPA-)security from LWE:

  [Regev'05,PVW'08,GPV'08,P'09,LP'11,. . .]

Pedigree

Learning With Errors (LWE) [Regev'05]

- Lineage of [Ajtai'96,AjtaiDwork'97]: worst-case/average-case reductions: breaking random inputs $\Longrightarrow$ solving famous problems on any lattice.

  > *"[This] assures us that attacks on the cryptographic construction are likely to be effective only for small choices of parameters and not asymptotically. In other words . . . there are no fundamental flaws in the design of our cryptographic construction."* *[MicciancioRegev'09]*

- LWE has been heavily used and cryptanalyzed by countless works.

Public-Key Encryption/Key Exchange

- Many schemes with tight (CPA-)security from LWE:
  [Regev'05,PVW'08,GPV'08,P'09,LP'11,. . .]

- FrodoCCS [BCDMNNRS'16] instantiated and implemented [LP'11], using pseudorandom public matrix $\mathbf{A}$ to reduce public key size.

Pedigree

Learning With Errors (LWE) [Regev'05]

- Lineage of [Ajtai'96,AjtaiDwork'97]: worst-case/average-case reductions: breaking random inputs $\Longrightarrow$ solving famous problems on any lattice.

  > *"[This] assures us that attacks on the cryptographic construction are likely to be effective only for small choices of parameters and not asymptotically. In other words ... there are no fundamental flaws in the design of our cryptographic construction."* *[MicciancioRegev'09]*

- LWE has been heavily used and cryptanalyzed by countless works.

Public-Key Encryption/Key Exchange

- Many schemes with tight (CPA-)security from LWE:
  [Regev'05,PVW'08,GPV'08,P'09,LP'11,. . .]
- FrodoCCS [BCDMNNRS'16] instantiated and implemented [LP'11], using pseudorandom public matrix $\mathbf{A}$ to reduce public key size.
- FrodoPKE/KEM [this work]: wider error, new params, CCA security

LWE and FrodoPKE

Learning With Errors

- Dimension n, modulus q, error distribution χ on ‘small’ integers.

LWE and FrodoPKE

Learning With Errors

- Dimension n, modulus q, error distribution χ on 'small' integers.

  Assumption: for uniformly random matrix $\mathbf{A}$ over $\mathbb{Z}_q$ and $\mathbf{S}$ from χ,

  $$[\mathbf{A}\ ,\ \mathbf{B} \approx \mathbf{S}\mathbf{A}] \stackrel{c}{\equiv} \text{uniform over } \mathbb{Z}_q.$$

LWE and FrodoPKE

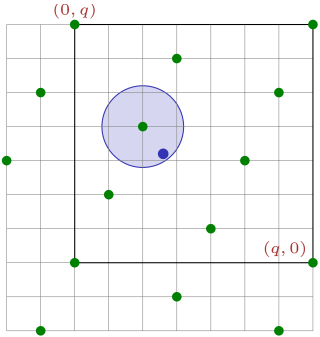
Learning With Errors

- Dimension n, modulus q, error distribution χ on 'small' integers.

  Assumption: for uniformly random matrix $\mathbf{A}$ over $\mathbb{Z}_q$ and $\mathbf{S}$ from χ,

$$[\mathbf{A}\ ,\ \mathbf{B} \approx \mathbf{S}\mathbf{A}] \stackrel{c}{\equiv} \text{uniform over } \mathbb{Z}_q.$$

Bounded-distance decoding on a random 'q-ary' lattice defined by $\mathbf{A}$:



LWE and FrodoPKE

Learning With Errors

- Dimension n, modulus q, error distribution χ on 'small' integers.

  Assumption: for uniformly random matrix $\mathbf{A}$ over $\mathbb{Z}_q$ and $\mathbf{S}$ from χ,

$$[\mathbf{A}\ ,\ \mathbf{B} \approx \mathbf{S}\mathbf{A}] \stackrel{c}{\equiv} \text{uniform over } \mathbb{Z}_q.$$

$\mathbf{S} \leftarrow \chi^{k \times n}$

$$\xrightarrow[(\mathbf{A} = \mathsf{expand}(\mathsf{seed}_{\mathbf{A}}) \in \mathbb{Z}_q^{n \times n})]{pk = \mathsf{seed}_{\mathbf{A}}\ ,\ \mathbf{B} \approx \mathbf{S}\mathbf{A}}$$

(Images courtesy xkcd.org)

LWE and FrodoPKE

Learning With Errors

- Dimension n, modulus q, error distribution χ on 'small' integers.

  Assumption: for uniformly random matrix $\mathbf{A}$ over $\mathbb{Z}_q$ and $\mathbf{S}$ from χ,

  $$[\mathbf{A} \ , \ \mathbf{B} \approx \mathbf{S}\mathbf{A}] \stackrel{c}{\equiv} \text{uniform over } \mathbb{Z}_q.$$

$\mathbf{S} \leftarrow \chi^{k \times n}$

$$\xrightarrow[(\mathbf{A} = \mathsf{expand}(\mathsf{seed}_{\mathbf{A}}) \in \mathbb{Z}_q^{n \times n})]{pk = \mathsf{seed}_{\mathbf{A}} \ , \ \mathbf{B} \approx \mathbf{S}\mathbf{A}}$$

$\mathbf{M} \in \{0,1\}^{k \times \ell}$

(Images courtesy xkcd.org)

LWE and FrodoPKE

Learning With Errors

- Dimension n, modulus q, error distribution χ on ‘small’ integers.

  Assumption: for uniformly random matrix $\mathbf{A}$ over $\mathbb{Z}_q$ and $\mathbf{S}$ from χ,

  $$[\mathbf{A}\ ,\ \mathbf{B} \approx \mathbf{SA}] \stackrel{c}{\equiv} \text{uniform over } \mathbb{Z}_q.$$

$\mathbf{S} \leftarrow \chi^{k \times n}$

$$\xrightarrow[(\mathbf{A} = \mathsf{expand}(\mathsf{seed}_{\mathbf{A}}) \in \mathbb{Z}_q^{n \times n})]{pk = \mathsf{seed}_{\mathbf{A}}\ ,\ \mathbf{B} \approx \mathbf{SA}}$$

$$\xleftarrow[\mathbf{C}' \approx \mathbf{BR} + \frac{q}{2} \cdot \mathbf{M}]{\mathbf{C}\ \approx \mathbf{AR}}$$

$\mathbf{M} \in \{0,1\}^{k \times \ell}$

(Images courtesy xkcd.org)

LWE and FrodoPKE

Learning With Errors

- Dimension n, modulus q, error distribution χ on 'small' integers.

  Assumption: for uniformly random matrix $\mathbf{A}$ over $\mathbb{Z}_q$ and $\mathbf{S}$ from χ,

  $$[\mathbf{A}\ ,\ \mathbf{B} \approx \mathbf{SA}] \stackrel{c}{\equiv} \text{uniform over } \mathbb{Z}_q.$$

$\mathbf{S} \leftarrow \chi^{k \times n}$

$$\xrightarrow[(\mathbf{A} = \mathsf{expand}(\mathsf{seed}_{\mathbf{A}}) \in \mathbb{Z}_q^{n \times n})]{pk = \mathsf{seed}_{\mathbf{A}}\ ,\ \mathbf{B} \approx \mathbf{SA}}$$

$$\xleftarrow[\mathbf{C}' \approx \mathbf{BR} + \frac{q}{2} \cdot \mathbf{M}]{\mathbf{C} \approx \mathbf{AR}}$$

$\mathbf{M} \in \{0,1\}^{k \times \ell}$

$\mathbf{C}' - \mathbf{SC} \approx \frac{q}{2} \cdot \mathbf{M}$

(Images courtesy xkcd.org)

LWE and FrodoPKE

Learning With Errors

- Dimension n, modulus q, error distribution χ on 'small' integers.

  Assumption: for uniformly random matrix $\mathbf{A}$ over $\mathbb{Z}_q$ and $\mathbf{S}$ from χ,

  $$[\mathbf{A}\ ,\ \mathbf{B} \approx \mathbf{S}\mathbf{A}] \stackrel{c}{\equiv} \text{uniform over } \mathbb{Z}_q.$$

$\mathbf{S} \leftarrow \chi^{k \times n}$

$$\xrightarrow[(\mathbf{A} = \text{expand}(\text{seed}_{\mathbf{A}}) \in \mathbb{Z}_q^{n \times n})]{pk = \text{seed}_{\mathbf{A}}\ ,\ \mathbf{B} \approx \mathbf{S}\mathbf{A}}$$

$$\xleftarrow[\mathbf{C}' \approx \mathbf{B}\mathbf{R} + \frac{q}{2} \cdot \mathbf{M}]{\mathbf{C} \approx \mathbf{A}\mathbf{R}}$$

$\mathbf{M} \in \{0,1\}^{k \times \ell}$

$\mathbf{C}' - \mathbf{S}\mathbf{C} \approx \frac{q}{2} \cdot \mathbf{M}$

$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{C}') \stackrel{c}{\equiv}$ unif

(Images courtesy xkcd.org)

Distinctive Features of FrodoPKE/KEM

1. Generic, algebraically unstructured lattices: plain LWE.
   (No algebraic ring structure for potential exploitation.)

2. 'Medium-sized' errors conforming to a worst-case/average-case reduction from a previously studied lattice problem (BDD with DGS).

3. Very simple design and constant-time implementation:
   - power-of-2 modulus q for cheap & easy modular arithmetic
   - straightforward error sampling
   - no 'reconciliation' or error-correcting codes for removing noise
   - x64 implementation: 256 lines of plain C code
     (+ preexisting symmetric primitives)

Distinctive Features of FrodoPKE/KEM

1. Generic, algebraically unstructured lattices: plain LWE.
   (No algebraic ring structure for potential exploitation.)

2. 'Medium-sized' errors conforming to a worst-case/average-case reduction from a previously studied lattice problem (BDD with DGS).

3. Very simple design and constant-time implementation:
   - power-of-2 modulus q for cheap & easy modular arithmetic
   - straightforward error sampling
   - no 'reconciliation' or error-correcting codes for removing noise
   - x64 implementation: 256 lines of plain C code
     (+ preexisting symmetric primitives)

Distinctive Features of FrodoPKE/KEM

1. Generic, algebraically unstructured lattices: plain LWE.
   (No algebraic ring structure for potential exploitation.)

2. 'Medium-sized' errors conforming to a worst-case/average-case reduction from a previously studied lattice problem (BDD with DGS).

3. Very simple design and constant-time implementation:
   - power-of-2 modulus q for cheap & easy modular arithmetic
   - straightforward error sampling
   - no 'reconciliation' or error-correcting codes for removing noise
   - x64 implementation: 256 lines of plain C code
     (+ preexisting symmetric primitives)

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\Longrightarrow$ smaller parameters q, n $\Longrightarrow$ better efficiency.

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\implies$ smaller parameters q, n $\implies$ better efficiency.
- But how narrow can the error distribution *safely* be?

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\Longrightarrow$ smaller parameters q, n $\Longrightarrow$ better efficiency.
- But how narrow can the error distribution *safely* be?

Risk Category: Small Errors

1. LWE with $O(1)$-bounded error is poly(n)-time solvable [AG'11,ACFP'14]

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\implies$ smaller parameters q, n $\implies$ better efficiency.
- But how narrow can the error distribution *safely* be?

Risk Category: Small Errors

1. LWE with $O(1)$-bounded error is poly(n)-time solvable [AG'11,ACFP'14] given large-poly(n)-many samples. (PKEs don't reveal this many!)

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\Longrightarrow$ smaller parameters q, n $\Longrightarrow$ better efficiency.
- But how narrow can the error distribution *safely* be?

Risk Category: Small Errors

1. LWE with $O(1)$-bounded error is poly(n)-time solvable [AG'11,ACFP'14] given large-poly(n)-many samples. (PKEs don't reveal this many!)
2. Prior worst-case hardness needs Gaussian error of $\sigma > \sqrt{n}/(2\pi)$.

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\Longrightarrow$ smaller parameters q, n $\Longrightarrow$ better efficiency.
- But how narrow can the error distribution *safely* be?

Risk Category: Small Errors

1. LWE with $O(1)$-bounded error is poly(n)-time solvable [AG'11,ACFP'14] given large-poly(n)-many samples. (PKEs don't reveal this many!)
2. Prior worst-case hardness needs Gaussian error of $\sigma > \sqrt{n}/(2\pi)$.
   Or narrower error, but only for few LWE samples. (PKEs reveal more!)

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\Longrightarrow$ smaller parameters $q, n \Longrightarrow$ better efficiency.
- But how narrow can the error distribution *safely* be?

Risk Category: Small Errors

1. LWE with $O(1)$-bounded error is poly(n)-time solvable [AG'11,ACFP'14] given large-poly(n)-many samples. (PKEs don't reveal this many!)
2. Prior worst-case hardness needs Gaussian error of $\sigma > \sqrt{n}/(2\pi)$. Or narrower error, but only for few LWE samples. (PKEs reveal more!)

$\Longrightarrow$ Sizeable gap between known-vulnerable and worst-case-hard params.

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\implies$ smaller parameters q, n $\implies$ better efficiency.
- But how narrow can the error distribution *safely* be?

Risk Category: Small Errors

1. LWE with $O(1)$-bounded error is poly(n)-time solvable [AG'11,ACFP'14] given large-poly(n)-many samples. (PKEs don't reveal this many!)
2. Prior worst-case hardness needs Gaussian error of $\sigma > \sqrt{n}/(2\pi)$. Or narrower error, but only for few LWE samples. (PKEs reveal more!)

$\implies$ Sizeable gap between known-vulnerable and worst-case-hard params.

New Worst-Case Hardness

- A latent reduction from [R'05,PRS'17] works for our $\sigma \approx \eta(\mathbb{Z})$.

Medium-Sized Errors

Choosing an Error Distribution

- Narrower errors $\implies$ smaller parameters q, n $\implies$ better efficiency.
- But how narrow can the error distribution *safely* be?

Risk Category: Small Errors

1. LWE with $O(1)$-bounded error is poly(n)-time solvable [AG'11,ACFP'14] given large-poly(n)-many samples. (PKEs don't reveal this many!)
2. Prior worst-case hardness needs Gaussian error of $\sigma > \sqrt{n}/(2\pi)$. Or narrower error, but only for few LWE samples. (PKEs reveal more!)

$\implies$ Sizeable gap between known-vulnerable and worst-case-hard params.

New Worst-Case Hardness

- A latent reduction from [R'05,PRS'17] works for our $\sigma \approx \eta(\mathbb{Z})$.
- Works for a bounded poly(n) number of LWE samples: covers PKEs!

What's New in Round 2

1. Level 5 parameter set: FrodoKEM-1344

What's New in Round 2

1. Level 5 parameter set: FrodoKEM-1344
2. cSHAKE $\rightarrow$ SHAKE, refined domain separation, fewer calls to Keccak

What's New in Round 2

1. Level 5 parameter set: FrodoKEM-1344
2. cSHAKE $\rightarrow$ SHAKE, refined domain separation, fewer calls to Keccak
3. $\mathsf{QFO}^{\not\perp} \rightarrow \mathsf{FO}^{\not\perp}$ transformation: removed extra hash value in ct.

What's New in Round 2

1. Level 5 parameter set: FrodoKEM-1344
2. cSHAKE $\rightarrow$ SHAKE, refined domain separation, fewer calls to Keccak
3. $\mathsf{QFO}^{\not\perp} \rightarrow \mathsf{FO}^{\not\perp}$ transformation: removed extra hash value in ct.
   Rationale: (non-tight) QROM proof [JZCWM'18] of

   OW-CPA PKE $\Rightarrow$ IND-CCA KEM.

What's New in Round 2

1. Level 5 parameter set: FrodoKEM-1344
2. cSHAKE $\to$ SHAKE, refined domain separation, fewer calls to Keccak
3. $\mathsf{QFO}^{\not\perp} \to \mathsf{FO}^{\not\perp}$ transformation: removed extra hash value in ct.
   Rationale: (non-tight) QROM proof [JZCWM'18] of

   $$\text{OW-CPA PKE} \Rightarrow \text{IND-CCA KEM}.$$

4. Detailed, tight ROM proof [HHK'17,LSS'14] of

   $$\text{IND-CPA PKE} \Rightarrow \text{OW-PCA PKE} \Rightarrow \text{IND-CCA KEM},$$

   with 'Rényi switch' at OW-PCA step.

What's New in Round 2

1. Level 5 parameter set: FrodoKEM-1344
2. cSHAKE $\rightarrow$ SHAKE, refined domain separation, fewer calls to Keccak
3. $\mathsf{QFO}^{\not\perp} \rightarrow \mathsf{FO}^{\not\perp}$ transformation: removed extra hash value in ct.
   Rationale: (non-tight) QROM proof [JZCWM'18] of

   $$\text{OW-CPA PKE} \Rightarrow \text{IND-CCA KEM}.$$

4. Detailed, tight ROM proof [HHK'17,LSS'14] of

   $$\text{IND-CPA PKE} \Rightarrow \text{OW-PCA PKE} \Rightarrow \text{IND-CCA KEM},$$

   with 'Rényi switch' at OW-PCA step.
5. WIP: Cortex M4 implementation with 2x memory improvement

Tight ROM Proof of CCA Security

- Generic, tight transforms following [HHK'17]:

$$\begin{array}{c}\text{FrodoPKE}\\ \text{(IND-CPA)}\end{array} \xrightarrow{T} \begin{array}{c}\text{T[FrodoPKE]}\\ \text{(OW-PCA)}\end{array} \xrightarrow{U^{\not\perp}} \begin{array}{c}\text{FrodoKEM}\\ \text{(IND-CCA)}\end{array}$$

Tight ROM Proof of CCA Security

- Generic, tight transforms following [HHK'17]:

  FrodoPKE (IND-CPA) $\xrightarrow{T}$ T[FrodoPKE] (OW-PCA) $\xrightarrow{U^{\not\perp}}$ FrodoKEM (IND-CCA)

- For worst-case hardness, FrodoPKE uses 'ideal' Gaussian errors.
  For implementation, FrodoKEM uses 'approximate' Gaussian errors.

Tight ROM Proof of CCA Security

- Generic, tight transforms following [HHK'17]:

FrodoPKE (IND-CPA) $\xrightarrow{T}$ T[FrodoPKE] (OW-PCA) $\xrightarrow{U^{\not\perp}}$ FrodoKEM (IND-CCA)

- For worst-case hardness, FrodoPKE uses 'ideal' Gaussian errors.
  For implementation, FrodoKEM uses 'approximate' Gaussian errors.
- Switch at OW-PCA (search), security loss ≈ 0 by Rényi div [LSS'14].

(Precise, tiny bounds given in spec.)

Tight ROM Proof of CCA Security

- Generic, tight transforms following [HHK'17]:

$$\text{FrodoPKE (IND-CPA)} \xrightarrow{T} \text{T[FrodoPKE] (OW-PCA)} \xrightarrow{U^{\not\perp}} \text{FrodoKEM (IND-CCA)}$$

- For worst-case hardness, FrodoPKE uses 'ideal' Gaussian errors.
  For implementation, FrodoKEM uses 'approximate' Gaussian errors.
- Switch at OW-PCA (search), security loss ≈ 0 by Rényi div [LSS'14].
  (Precise, tiny bounds given in spec.)

Alternative Assumption: OW-PCA of T[FrodoPKE]

- OW-PCA $\equiv$ OW-CPA, unless attacker queries an $m \neq \mathsf{Dec}(\mathsf{Enc}(m))$.

Tight ROM Proof of CCA Security

- Generic, tight transforms following [HHK'17]:

$$\boxed{\begin{array}{c}\text{FrodoPKE}\\\text{(IND-CPA)}\end{array}} \xrightarrow{T} \boxed{\begin{array}{c}\text{T[FrodoPKE]}\\\text{(OW-PCA)}\end{array}} \xrightarrow{U^{\not\perp}} \boxed{\begin{array}{c}\text{FrodoKEM}\\\text{(IND-CCA)}\end{array}}$$

- For worst-case hardness, FrodoPKE uses 'ideal' Gaussian errors. For implementation, FrodoKEM uses 'approximate' Gaussian errors.
- Switch at OW-PCA (search), security loss ≈ 0 by Rényi div [LSS'14]. (Precise, tiny bounds given in spec.)

Alternative Assumption: OW-PCA of T[FrodoPKE]

- OW-PCA $\equiv$ OW-CPA, unless attacker queries an $m \neq \mathsf{Dec}(\mathsf{Enc}(m))$.
- Costs more than claimed security for our FrodoKEM params [DVV'19].

Tight ROM Proof of CCA Security

- Generic, tight transforms following [HHK'17]:

FrodoPKE (IND-CPA) $\xrightarrow{T}$ T[FrodoPKE] (OW-PCA) $\xrightarrow{U^{\not\perp}}$ FrodoKEM (IND-CCA)

- For worst-case hardness, FrodoPKE uses 'ideal' Gaussian errors.
  For implementation, FrodoKEM uses 'approximate' Gaussian errors.
- Switch at OW-PCA (search), security loss ≈ 0 by Rényi div [LSS'14].
  (Precise, tiny bounds given in spec.)

Alternative Assumption: OW-PCA of T[FrodoPKE]

- OW-PCA $\equiv$ OW-CPA, unless attacker queries an $m \neq \mathsf{Dec}(\mathsf{Enc}(m))$.
- Costs more than claimed security for our FrodoKEM params [DVV'19].
- So, $\approx$ OW-CPA of T[FrodoPKE] also suffices for CCA.

Concrete Parameters and Security

- Use 'core-SVP' methodology [ADPS'16] to lower-bound the *first-order exponential time* (and space) of SVP in appropriate dimension.

Concrete Parameters and Security

- Use 'core-SVP' methodology [ADPS'16] to lower-bound the *first-order exponential time* (and space) of SVP in appropriate dimension.

  This significantly underestimates the cost of known attacks, but it is prudent to expect better lower-order terms with further research.

Concrete Parameters and Security

- Use ‘core-SVP’ methodology [ADPS’16] to lower-bound the *first-order exponential time* (and space) of SVP in appropriate dimension.

  This significantly underestimates the cost of known attacks, but it is prudent to expect better lower-order terms with further research.

- LWE and classical CCA security (end-to-end from ROM proof):

				LWE Security		CCA (ROM)
	n	q	σ	C $\geq$	Q $\geq$	Sec $\geq$
FrodoKEM-640	640	2^{15}	2.75	145	104	141
FrodoKEM-976	976	2^{16}	2.3	210	150	206
FrodoKEM-1344	1344	2^{16}	1.4	275	197	268

Performance

- Sizes (in bytes):

	secret key	**public key**	**ciphertext**
FrodoKEM-640	10,272	9,616	9,720
FrodoKEM-976	15,664	15,632	15,744
FrodoKEM-1344	21,568	21,520	21,632

Performance

- Sizes (in bytes):

	secret key	public key	ciphertext
FrodoKEM-640	10,272	9,616	9,720
FrodoKEM-976	15,664	15,632	15,744
FrodoKEM-1344	21,568	21,520	21,632

- Speed (in kilocycles, 3.4GHz Intel Core i7-6700 Skylake, AES-NI):

	KeyGen	Encaps	Decaps
FrodoKEM-640	1,384	1,858	1,749
FrodoKEM-976	2,820	3,559	3,400
FrodoKEM-1344	4,756	5,981	5,748

Performance

- Sizes (in bytes):

	secret key	public key	ciphertext
FrodoKEM-640	10,272	9,616	9,720
FrodoKEM-976	15,664	15,632	15,744
FrodoKEM-1344	21,568	21,520	21,632

- Speed (in kilocycles, 3.4GHz Intel Core i7-6700 Skylake, AES-NI):

	KeyGen	Encaps	Decaps
FrodoKEM-640	1,384	1,858	1,749
FrodoKEM-976	2,820	3,559	3,400
FrodoKEM-1344	4,756	5,981	5,748

- Cache $\mathbf{A} \leftarrow \text{seed}_{\mathbf{A}}$ for pk lifetime: save $\approx 40\%$ in Encaps/Decaps

Parting Thought

FrodoKEM's security derives from plain *Learning With Errors*
on algebraically unstructured lattices,
parameterized cautiously to avoid known risk categories,
and to conform to a worst-case/average-case reduction.

`https://FrodoKEM.org`

Thanks!