RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CRYP-T10

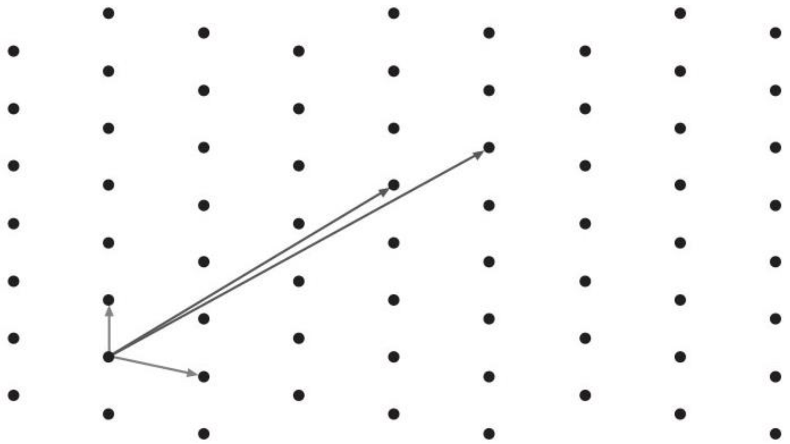
CRYPTANALYSIS OF COMPACT-LWE

Jonathan Bootle, Mehdi Tibouchi, Keita Xagawa

UCL
erc
uropean Research Council
stablished by the European Commission
NTT

Background Information

- Lattice-based cryptographic assumption



Based on the learning-with-errors (LWE) assumption

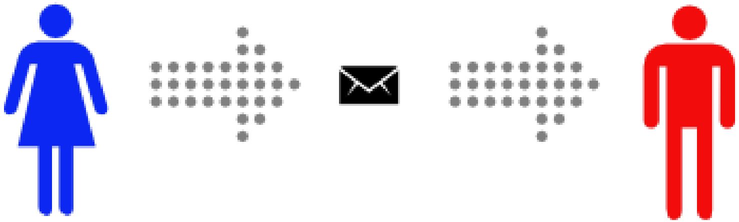
Compact-LWE

Hoped to achieve security for smaller parameters

Background Information

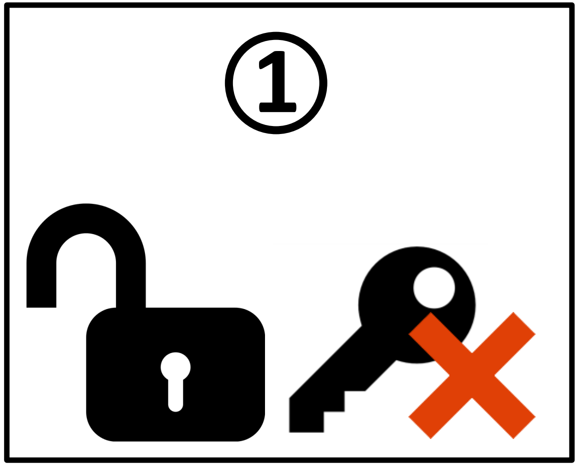
- Proposed by Liu, Li, Kim, and Nepal at ACISP'17 invited talk

- Gives lightweight encryption scheme for constrained devices

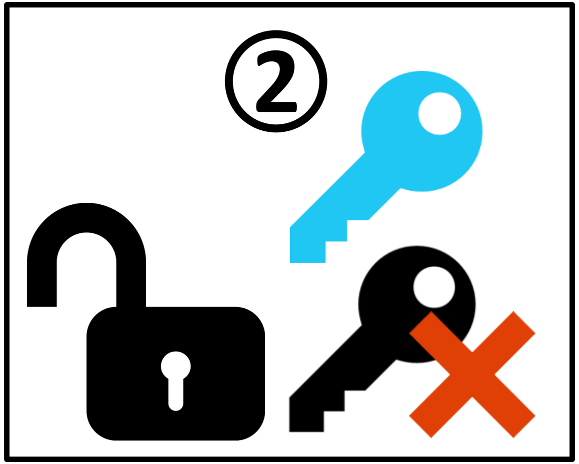


Background Information

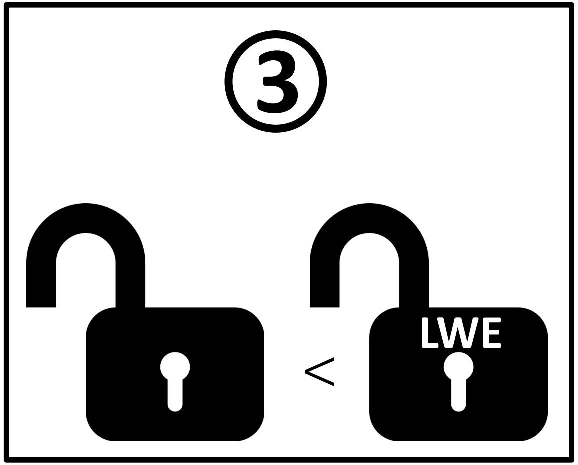
Basic Decryption Attack



Equivalent Secret Keys



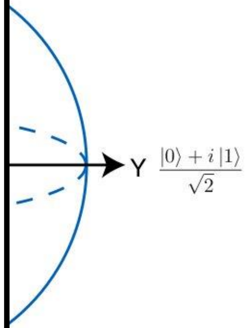
Parameter Choice



Honest Decryption:	500 ciphertexts per second
Our Decryption:	18,000 ciphertexts per second

Background Information

Cryptanalysed by us and others

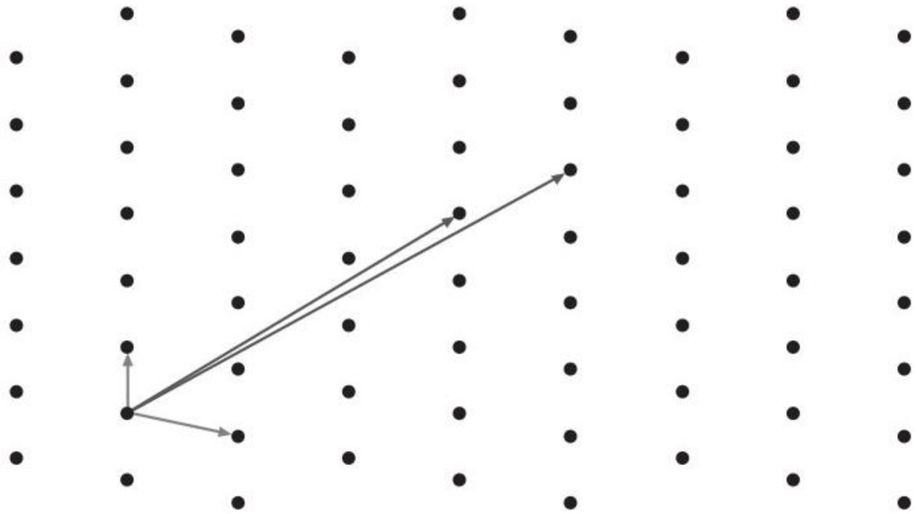


BACKGROUND

Lattices

An n-dimensional lattice $\mathcal{L}$ is

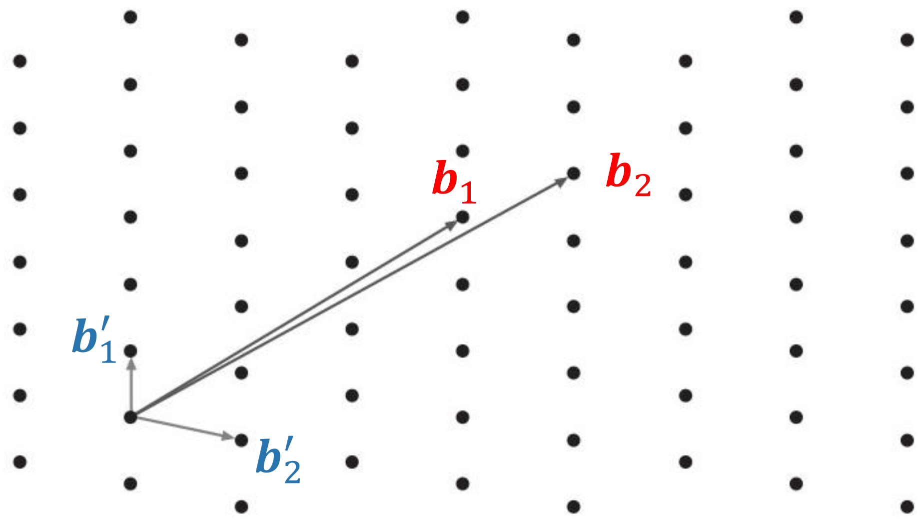
- A discrete additive subgroup of $\mathbb{R}^n$
- Generated by a basis $\mathcal{B} = \{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\}$
- $\mathcal{L} = \sum_{i=1}^{n}(\mathbb{Z} \cdot \boldsymbol{b}_i)$



Lattices

- Solve lattice problems by finding short vectors
- Example reduction algorithms are LLL and BKZ
- Add and subtract rows
- Find short basis vectors

$$\begin{pmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{pmatrix} \to \begin{pmatrix} \boldsymbol{b}'_1 \\ \boldsymbol{b}'_2 \end{pmatrix}$$

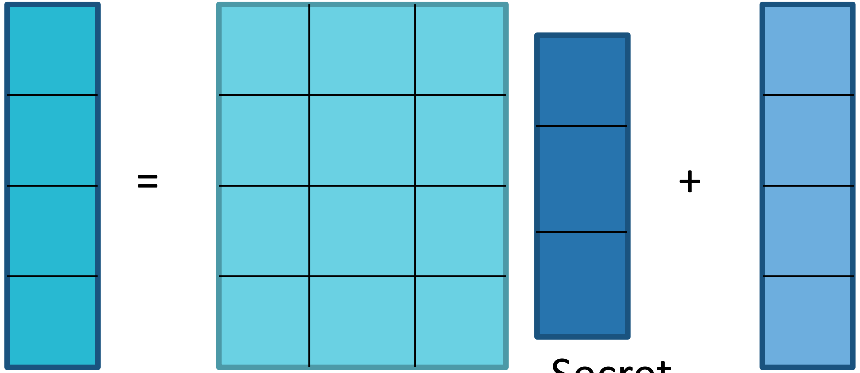


Learning with Errors

$$b_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + e_i$$

$$\boldsymbol{b} = A\boldsymbol{s} + \boldsymbol{e}$$

Samples b_i in $\boldsymbol{Z}_q$



Noise values $e_i \leftarrow \chi$

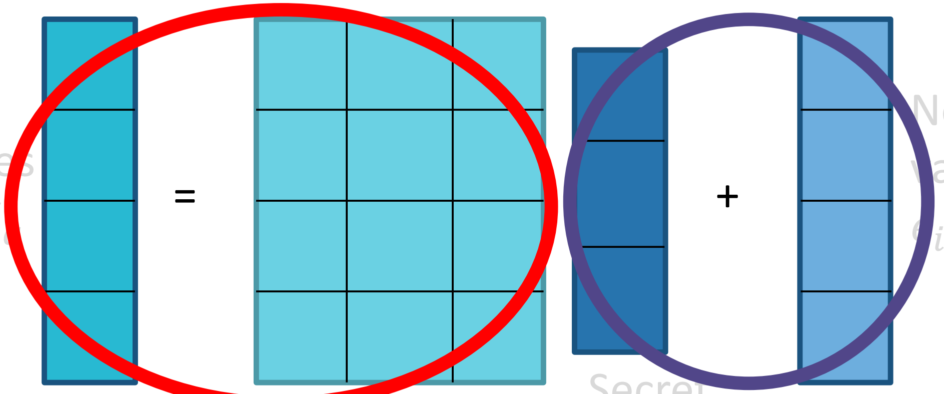
Uniformly random vectors $\boldsymbol{a}_i \in \boldsymbol{Z}_q^n$

Secret $\boldsymbol{s} \in \boldsymbol{Z}_q^n$

Learning with Errors

Decision: does $(\boldsymbol{b}, A)$ look random?
Search: given $(\boldsymbol{b}, A)$, find $\boldsymbol{s}$

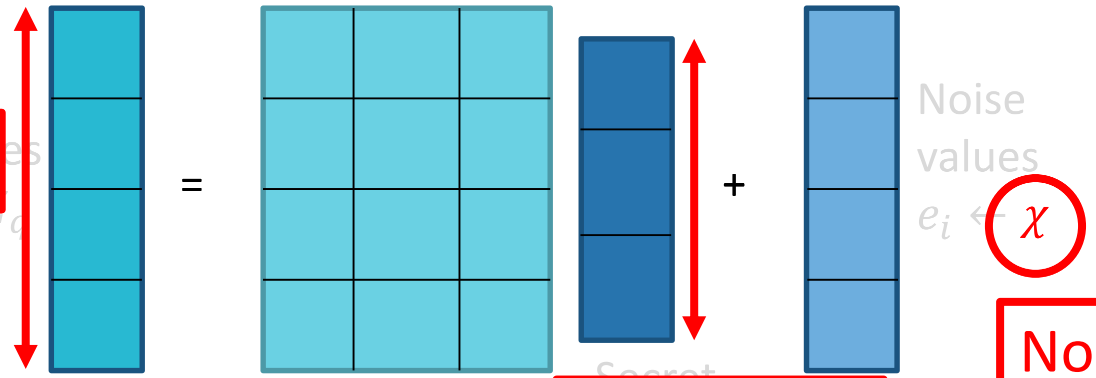
$\boldsymbol{b}$ and A are public



$\boldsymbol{s}$ and $\boldsymbol{e}$ are private

Learning with Errors

Decision: does $(\boldsymbol{b}, A)$ look random?
Search: given $(\boldsymbol{b}, A)$, find $\boldsymbol{s}$



m samples

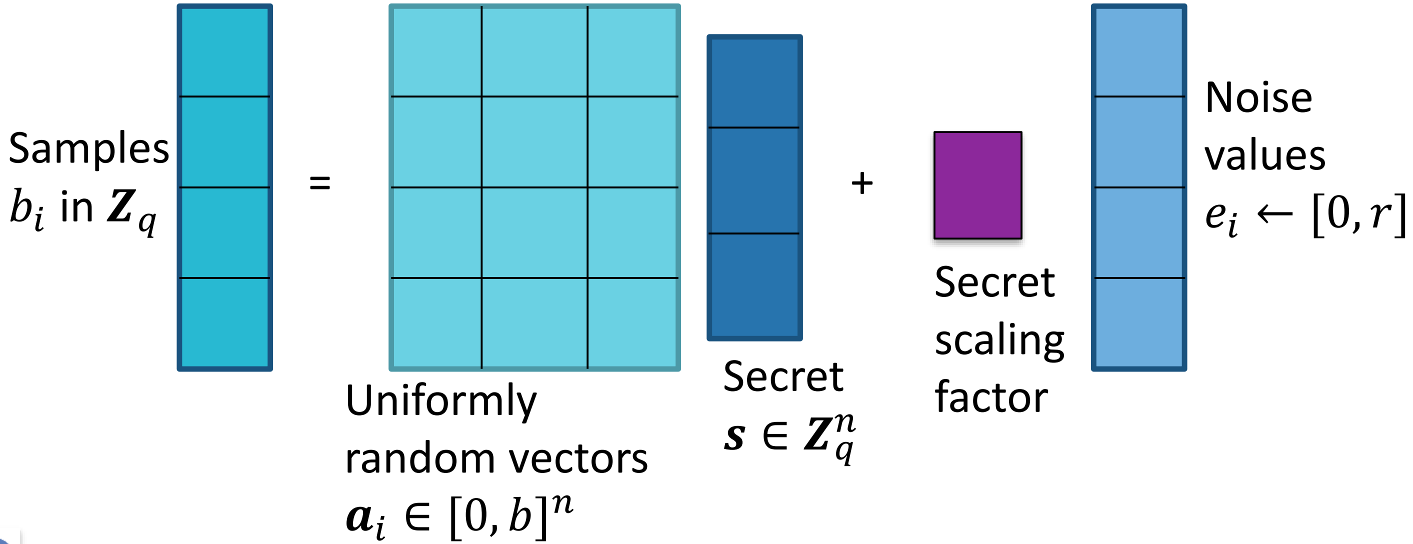
Noise distribution χ

Dimension
n secrets

Modulus q

Compact-LWE

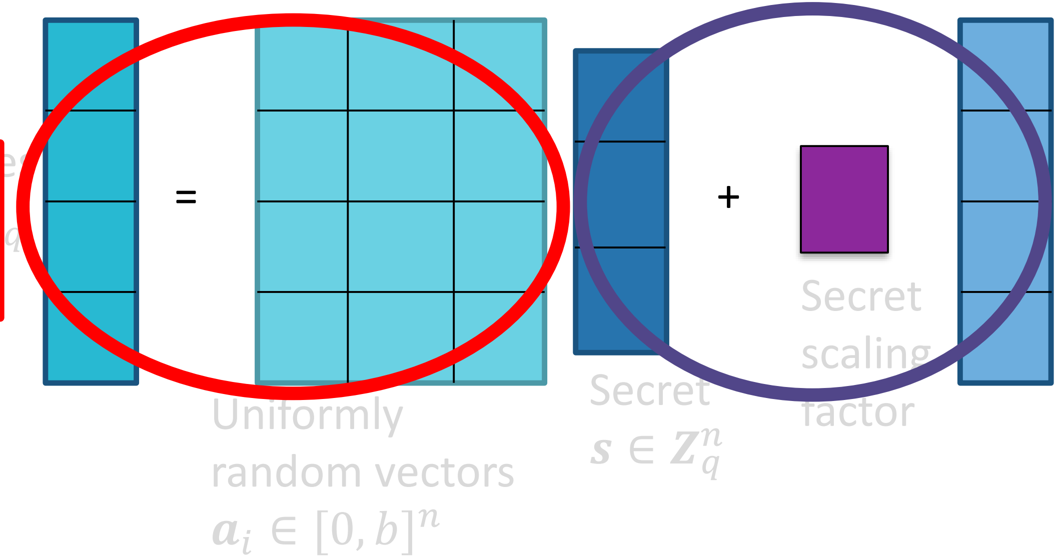
$$b_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + sk_q^{-1} \cdot p \cdot e_i$$



Compact-LWE

Decision: does $(\boldsymbol{b}, A)$ look random?
Search: given $(\boldsymbol{b}, A)$, find $\boldsymbol{s}$

$\boldsymbol{b}$ and A are public



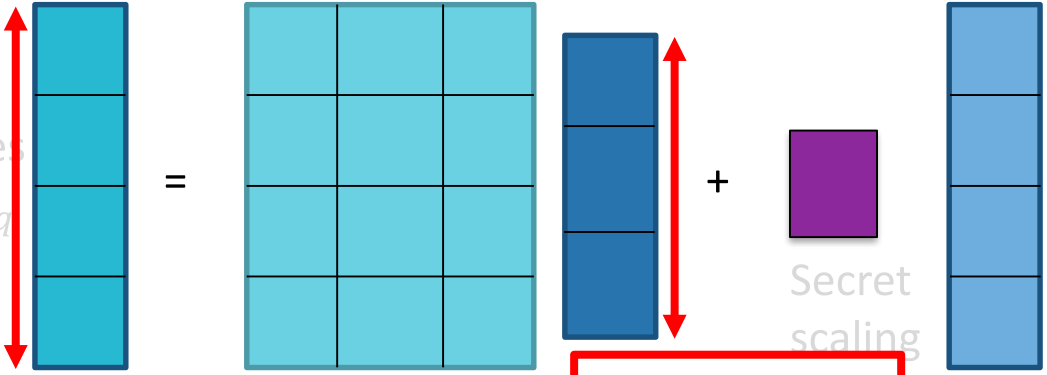
$\boldsymbol{s}$, $\boldsymbol{e}$ and the scaling factor are private

Compact-LWE

$$b_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + sk_q^{-1} \cdot p \cdot$$

Scaling factor ingredients

m samples



Noise values $e_i \leftarrow [0, r]$

Secret scaling

Noise bound r

Uniformly ... m vectors ... $[0, b]^T$

maximum size of A elements

Dimension n secrets modulo q

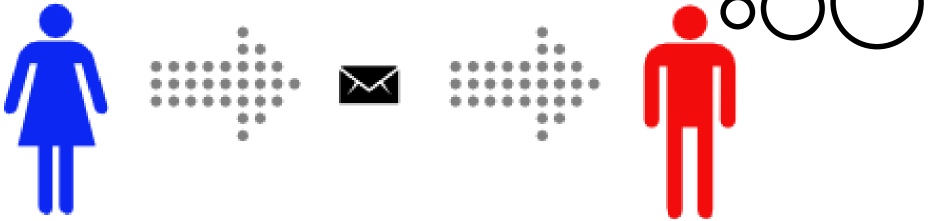
Parameters

Public Parameters

- $pp = (q, n, m, t, w, b)$
- t, maximum plaintext size
- w, knapsack weight for encryption
- $\boldsymbol{PK} = (A, \boldsymbol{b})$

Secret Parameters

- $\boldsymbol{K} = (\boldsymbol{s}, sk, r, p)$



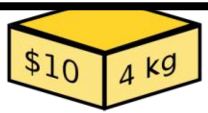
$$n + 1 < m < n^2$$
$$2b(b\,log_2 b + 1) < q$$
$$2log_2 b < n$$
$$t \leq p$$
$$sk \cdot (t - 1) + wrp < q$$
$$b < r$$

Encryption Idea

- $\boldsymbol{PK}$ contains random-looking samples $(\boldsymbol{a}_i, b_i)$ from $(A, \boldsymbol{b})$
- Add knapsack of b_i to hide message
- Include same knapsack of $\boldsymbol{a}_i$ to allow decryption

Enc$(\boldsymbol{PK}, v)$:

- Randomly pick w samples $(\boldsymbol{a}_{i_j}, b_{i_j})$ from $\boldsymbol{PK}$
- $(\boldsymbol{a}, b) = \sum_{j=1}^{w} (\boldsymbol{a}_{i_j}, b_{i_j})$
- Return c $= (\boldsymbol{a}, v - b)$



Comparison of Parameters

Compact-LWE Parameters

- Claims 138-bit security
- $q = 2^{32}$
- $n = 13$
- $m = 74$
- $t = 2^{16}, w = 86, b = 16$

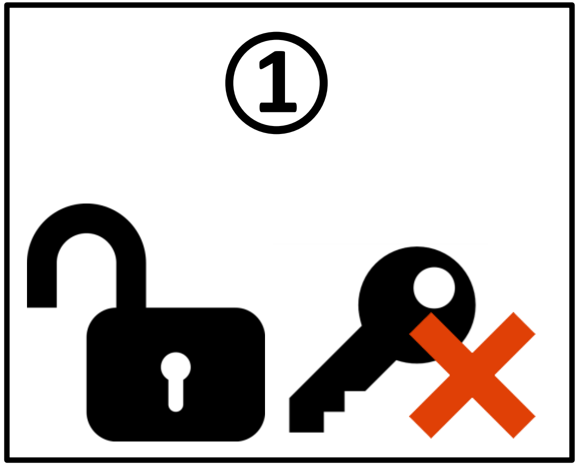
Lizard, Classical Parameters, 2016

- Claims 128-bit security
- $q \approx 2^{10}$
- $n = 544$
- $m = 840$

Implementation Results

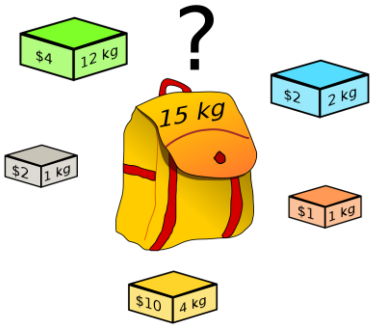
- Implemented on MTM-CM5000-MSP device
- Contiki OS
- 50 encryptions per second
- 500 decryptions per second

BASIC DECRYPTION ATTACK



Attack Strategy

- $\mathrm{c} = (\boldsymbol{a}, v - b) = (\boldsymbol{a}, b')$
- $(\boldsymbol{a}, b) = \sum_{j=1}^{w} (\boldsymbol{a}_{i_j}, b_{i_j})$
- Create lattice encoding knapsack
- Find a short vector with lattice reduction



$$\begin{pmatrix} 1 & \mathbf{0} & \mathbf{0} & v \end{pmatrix}$$

$$\begin{pmatrix} 1 & \mathbf{0} & \kappa \boldsymbol{a} & b' \\ \mathbf{0} & t I_m & -\kappa A & \boldsymbol{b} \\ 0 & \mathbf{0} & \mathbf{0} & q \end{pmatrix}$$

Solves knapsack

Recovers plaintext

Experimental Results

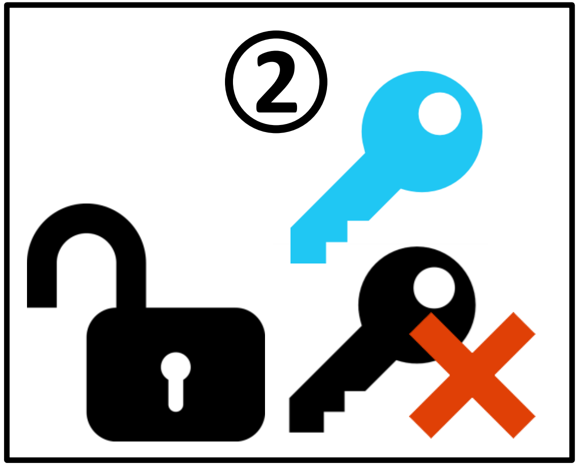
- Correctly decrypted 9998/10,000 random ciphertexts
- Roughly 16 decryptions per second
- 3.4 GHz Core i7-3770 desktop
- Sagemath, LLL in fplll

- Honest decryption: 500 decryptions per second, constrained device

- One lattice reduction per ciphertext
- Relies on low dimension n = 13

SECRET KEY RECOVERY

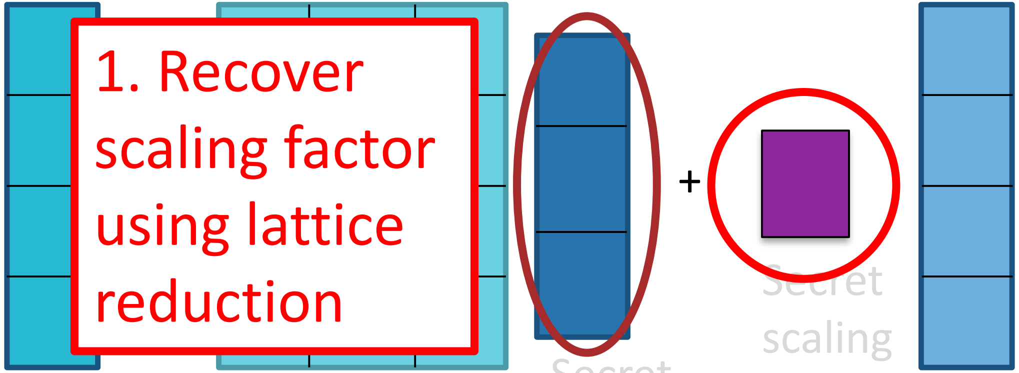
*equivalent secret key



Attack Strategy

$$b_i = \langle \boldsymbol{a}_i, \boldsymbol{s} \rangle + sk_q^{-1} \cdot p + e_i$$

2. Find other secret values by brute force



1. Recover scaling factor using lattice reduction

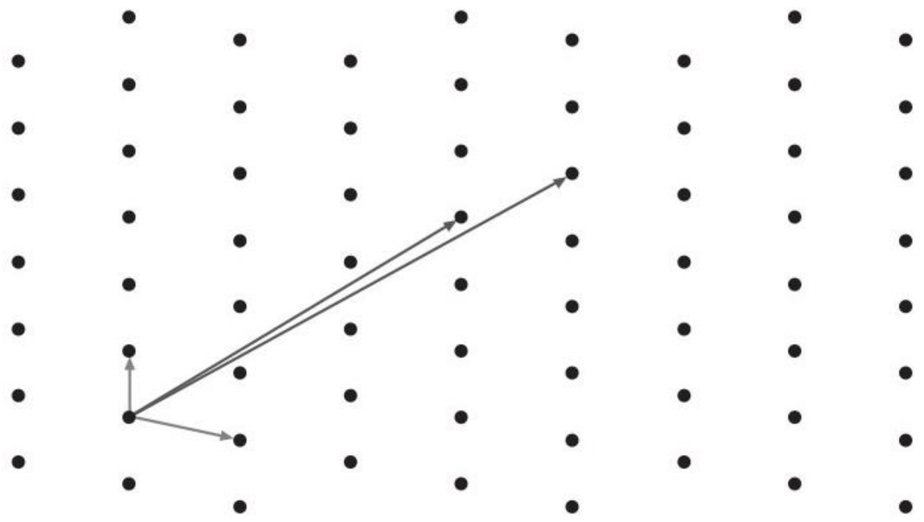
3. Compute equivalent secret using lattice reduction

Step 1: Scale-factor Recovery

- $\boldsymbol{b} = A\boldsymbol{s} + k\boldsymbol{e}$
- Compute short U such that $U^T A = 0 \bmod q$
- $U\boldsymbol{b} = k \cdot U\boldsymbol{e} \bmod q$

Public

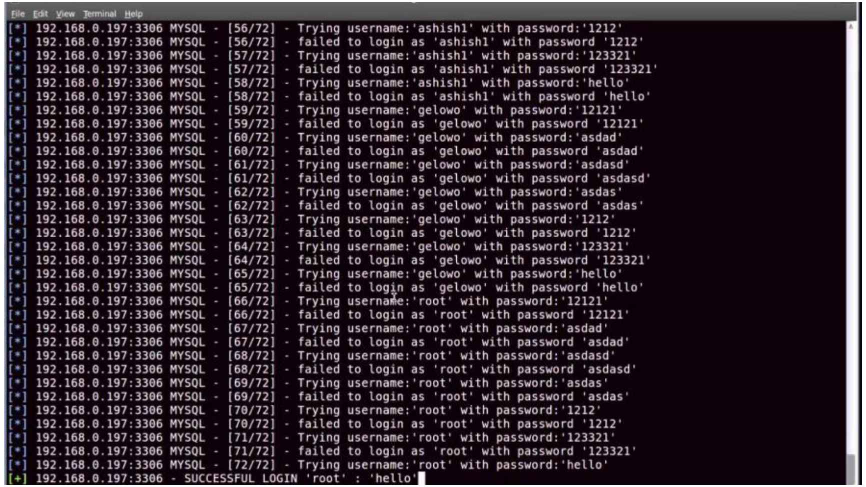
Short vector in $\begin{pmatrix} (U\boldsymbol{b})^T \\ qI \end{pmatrix}$



Step 2: Recovering Secret Key Parameters

- Secret scale-factor is $k = sk_q^{-1} \cdot p$
- Brute force search for sk and p
- Use the values which maximise r

$$sk \cdot (t - 1) + wrp < q$$



Step 3: Find an Equivalent Secret

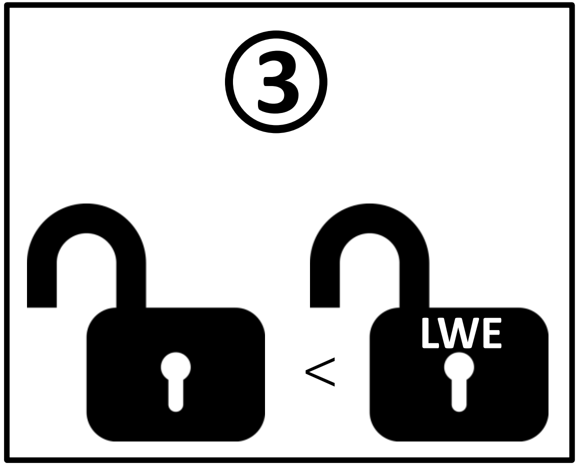
- Secret is a short lattice vector
- Use with modified decryption algorithm

$$\begin{pmatrix} A^T & 0 \\ qI_m & 0 \\ k^{-1} & t \end{pmatrix}$$

Experimental Results

- Correctly decrypted 10,000/10,000 random ciphertexts
- 1.28 seconds to get a key
- 53 microseconds per ciphertext
- Over 18,000 decryptions per second

PARAMETER CHOICE



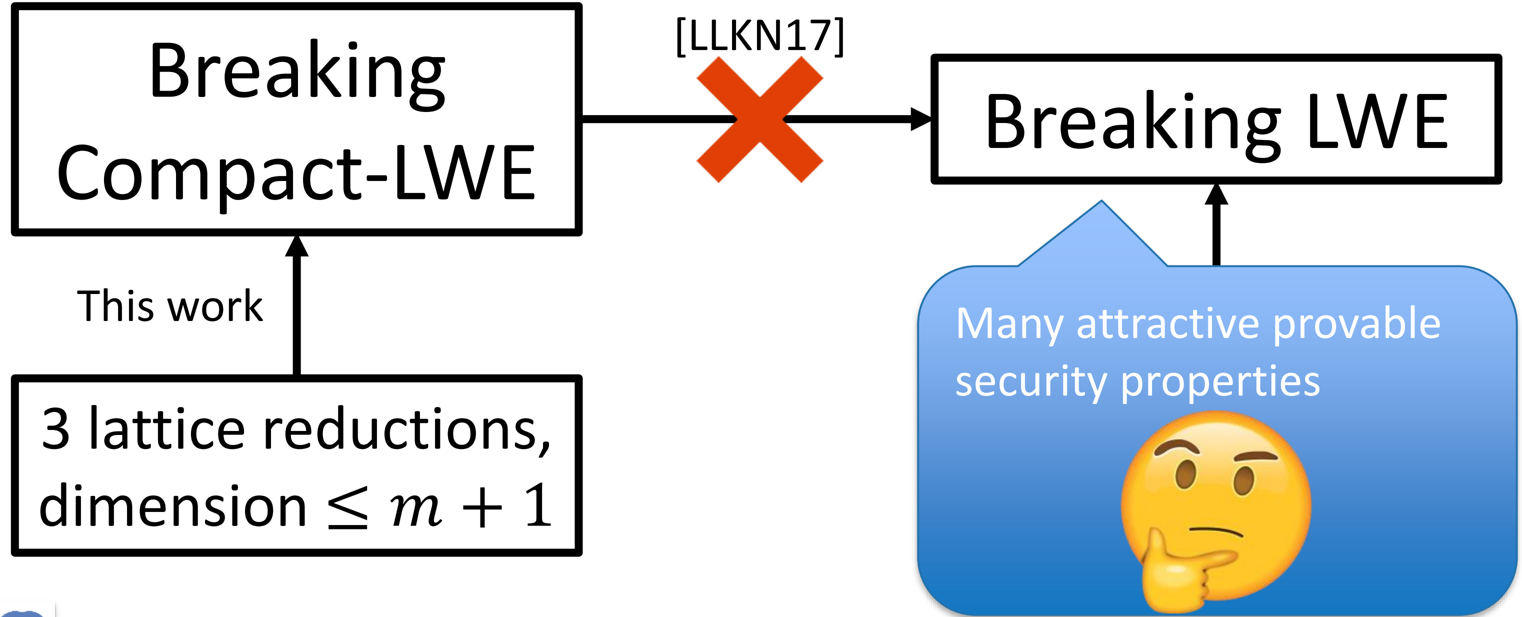
Hardness Reductions

Breaking Compact-LWE $\xrightarrow{\text{[LLKN17]}}$ Breaking LWE

Dimension 13 | Dimension 13

Easy...

Hardness Reductions



THANKS!

NIST Version Attack Paper: https://eprint.iacr.org/2018/020.pdf

NIST Version Attack Code: https://goo.gl/2Vo3T7

RSAConference2018
San Francisco | April 16 – 20 | Moscone Center

SESSION ID: CRYP-T10

TWO-MESSAGE KEY EXCHANGE WITH STRONG SECURITY FROM IDEAL LATTICES

Yu Chen

Associate Professor
Institute of Information Engineering, Chinese Academy of Sciences

Two-message Key Exchange with Strong Security from Ideal Lattices

Zheng Yang (University of Helsinki)
Yu Chen (Chinese Academy of Sciences)
Song Luo (Chongqing University of Technology)

April 17th, CT-RSA 2018

Background: Two-message Key Exchange (TMKE)

- Two-message Key exchange
  - Two messages: m_{id1}, m_{id2}——derived from party's (ephemeral) secrets.
  - Shared session key *K*——computed from party's (ephemeral) secrets and exchanged messages
  - Appealing to practice: low bandwidth and asynchronous communication

id_1		id_2
Message generation function	**Protocol Execution**	
Session key generation function	$\xrightarrow{m_{\mathsf{ID}_1}}$ $\xleftarrow{m_{\mathsf{ID}_2}}$	
accept K		accept K

The Simplest Example of TMKE

- Seminal TMKE: Diffie-Hellman key exchange (DHKE) [DH76]
  - Cyclic group G =< g > of prime order p
  - Two messages: X, Y
  - Passively secure; active attacker can implement man-in-the middle attack

id_1		id_2
	Protocol Execution	
$x \xleftarrow{\$} \mathbb{Z}_p^*$ $X := g^x$		$y \xleftarrow{\$} \mathbb{Z}_p^*$ $Y := g^y$
	$\xrightarrow{\quad X \quad}$	
	$\xleftarrow{\quad Y \quad}$	
accept $K := Y^x$		accept $K := X^y$

Motivation

- Quantum computers are about to get real
- DL, factoring, …., not hard against quantum algorithms
- Lattice-based Cryptography
  - Quantum secure
  - Simple, efficient, and highly parallel
- Existing Lattice-based AKE, e.g.:
  - AsiaCCS'13, Fujioka et al. ,

    —— standard model, CK+ model without perfect forward secrecy (PFS)
  - Eurocrypt'15, Zhang et al.,

    —— random oracle, BR model without PFS and leakage of ephemeral secret key
  - **CT-RSA'14, Kurosawa and Furukawa (KF scheme)**

    ——standard model, eCK model without PFS **Is it Secure?**

Overview of Our Results

- Revisit the security of the KF scheme (CT-RSA’14)
  - finding an attack
- Propose a new generic TMKE scheme
  - New cryptographic primitive: One-time CCA-secure KEM
  - Without random oracles
  - eCK-PFS model: known session key (KSK) , key compromise impersonation (KCI) , chosen identity and public key (CIDPK), ephemeral secret key leakage (ESKL), and perfect forward secrecy (PFS)
- Instantiation of TMKE from ideal lattices

The KF scheme

- Building Blocks: Twisted PRF (TPRF), Signature (SIG), IND-CPA KEM (wKEM)

id_1		id_2
$(sk_{\mathsf{id}_1}, pk_{\mathsf{id}_1}) = ((ssk_{\mathsf{id}_1}, s_{\mathsf{id}_1}), spk_{\mathsf{id}_1})$		$(sk_{\mathsf{id}_2}, pk_{\mathsf{id}_2}) = ((ssk_{\mathsf{id}_2}, s_{\mathsf{id}_2}), spk_{\mathsf{id}_2})$

Protocol Execution

ElGamal

$r_1, r_2 \xleftarrow{\$} \{0,1\}^*$		$r_3, r_4 \xleftarrow{\$} \{0,1\}^*$
$R_1 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_1)$		$R_3 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_3)$
$R_2 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_2)$		$R_4 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_4)$
$(epk = a,\ epk = g^a) \leftarrow \mathsf{wKEM.Gen}(1^\kappa, R_1)$		$(C = (g^r, g^{ra} \cdot K),\ K) \leftarrow \mathsf{wKEM.Gen}(epk_{\mathsf{id}_1}, R_3)$
$X = (\mathsf{id}_1, epk)$	$\xrightarrow{X, \sigma_X}$	$Y = (\mathsf{id}_2, C)$
$\sigma_X \leftarrow \mathsf{SIG.Sign}(ssk_{\mathsf{id}_1}, X, R_2)$	$\xleftarrow{Y, \sigma_{YX}}$	$\sigma_{YX} \leftarrow \mathsf{SIG.Sign}(ssk_{\mathsf{id}_2}, Y \| X, R_4)$
$K := \mathsf{wKEM.Dec}(esk_{\mathsf{id}_1}, C)$		

Insecurity of the KF scheme: Attack

id_1	$\mathcal{A}$	id_2
$(sk_{\mathsf{id}_1}, pk_{\mathsf{id}_1}) = ((ssk_{\mathsf{id}_1}, s_{\mathsf{id}_1}), spk_{\mathsf{id}_1})$	(Adversary)	$(sk_{\mathsf{id}_2}, pk_{\mathsf{id}_2}) = ((ssk_{\mathsf{id}_2}, s_{\mathsf{id}_2}), spk_{\mathsf{id}_2})$

Protocol Execution

test oracle

$\pi^s_{\mathsf{id}_1}$	Forward X, σ_X Intercept $Y, \sigma_{Y,X}$	$\pi^t_{\mathsf{id}_2}$
	Corrupt ssk_{id_2}	
	$C_{\mathcal{A}} := (g^{r\beta}, g^{ar\beta} \cdot K^{\beta})$ for arbitrary β	
$R_1 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_1)$	$Y_{\mathcal{A}} := (\mathsf{id}_2, C_{\mathcal{A}})$	$R_3 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_3)$
$R_2 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_2)$		$R_4 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_4)$
$(a, g^a) \leftarrow \mathsf{wKEM.Gen}(1^{\kappa}, R_1)$	$\sigma_{\mathcal{A}} := \mathsf{SIG.Sign}(ssk_{\mathsf{id}_2}, Y_{\mathcal{A}} \Vert X, R')$	$((g^r, g^{ar} \cdot K), K) \leftarrow \mathsf{wKEM.Gen}(g^a, R_3)$
$X = (\mathsf{id}_1, g^a)$	$\xrightarrow{X, \sigma_X}$ Drop Y, σ_{YX} $\xrightarrow{X, \sigma_X}$	$Y = (\mathsf{id}_2, g^r, g^{ar} \cdot K)$
$\sigma_X \xleftarrow{\$} \mathsf{SIG.Sign}(ssk_{\mathsf{id}_1}, X, R_2)$	$\xleftarrow{Y_{\mathcal{A}}, \sigma_{\mathcal{A}}}$ Reveal $K_{\mathcal{A}}$ from $\pi^s_{\mathsf{id}_1}$ $\xleftarrow{Y, \sigma_{YX}}$	$\sigma_{YX} \xleftarrow{\$} \mathsf{SIG.Sign}(ssk_{\mathsf{id}_2}, Y \Vert X, R_4)$
$K_{\mathcal{A}} := K^{\beta} = \frac{g^{ar\beta} \cdot K^{\beta}}{g^{r\beta a}}$	Compute $K := {K_{\mathcal{A}}}^{\beta^{-1}}$	

test oracle is fresh and has no partner oracle at id_1

Insecurity of the KF scheme: Problems

id_1	$\mathcal{A}$	id_2
$(sk_{\mathsf{id}_1}, pk_{\mathsf{id}_1}) = ((ssk_{\mathsf{id}_1}, s_{\mathsf{id}_1}), spk_{\mathsf{id}_1})$	(Adversary)	$(sk_{\mathsf{id}_2}, pk_{\mathsf{id}_2}) = ((ssk_{\mathsf{id}_2}, s_{\mathsf{id}_2}), spk_{\mathsf{id}_2})$

Protocol Execution

Not tied to session info
unknown key share

$\pi^s_{\mathsf{id}_1}$	Forward X, σ_X Intercept $Y, \sigma_{Y,X}$	$\pi^t_{\mathsf{id}_2}$
	Corrupt ssk_{id_2}	
	$C_{\mathcal{A}} := (g^{r\beta}, g^{ar\beta} \cdot K^{\beta})$ for arbitrary β	
$R_1 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_1)$	$Y_{\mathcal{A}} := (\mathsf{id}_2, C_{\mathcal{A}})$	$R_3 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_3)$
$R_2 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_2)$		$R_4 \leftarrow \mathsf{TPRF}(s_{\mathsf{id}_1}, r_4)$
$(a, g^a) \leftarrow \mathsf{wKEM.Gen}(1^{\kappa}, R_1)$	$\sigma_{\mathcal{A}} := \mathsf{SIG.Sign}(ssk_{\mathsf{id}_2}, Y_{\mathcal{A}} \Vert X, R')$	$((g^r, g^{ar} \cdot K), K) \leftarrow \mathsf{wKEM.Gen}(g^a, R_3)$
$X = (\mathsf{id}_1, g^a)$	$\xrightarrow{X, \sigma_X}$ Drop Y, σ_{YX} $\xrightarrow{X, \sigma_X}$	$Y = (\mathsf{id}_2, g^r, g^{ar} \cdot K)$
$\sigma_X \xleftarrow{\$} \mathsf{SIG.Sign}(ssk_{\mathsf{id}_1}, X, R_2)$	$\xleftarrow{Y_{\mathcal{A}}, \sigma_{\mathcal{A}}}$ Reveal $K_{\mathcal{A}}$ from $\pi^s_{\mathsf{id}_1}$ $\xleftarrow{Y, \sigma_{YX}}$	$\sigma_{YX} \xleftarrow{\$} \mathsf{SIG.Sign}(ssk_{\mathsf{id}_2}, Y \Vert X, R_4)$
$K_{\mathcal{A}} := K^{\beta} = \frac{g^{ar\beta} \cdot K^{\beta}}{g^{r\beta a}}$	Compute $K := {K_{\mathcal{A}}}^{\beta^{-1}}$	

One-time CCA attack against the KEM, CPA-secure KEM does not suffice

How to remedy the KF scheme?

- Main idea
  - Enhance the security of KEM
    - CPA to one-time CCA
  - Employ Key Derivation function
    - bind the session key with session specific information to defend active attacks

Our New Generic TMKE Protocol

- Building blocks
  - **One-time KEM (OTKEM):** encapsulate the session key
  - **Signature (SIG):** authenticate exchanged messages
  - **IND-CPA KEM (wKEM)**: implement NAXOS trick against ephemeral key leakage (generate the pk for OTKEM)
  - **Pseudo-random function (PRF)**: act as KDF to bind session key with session specific information

A New Generic TMKE Protocol

id_1		id_2
$(sk_{\mathsf{id}_1}, pk_{\mathsf{id}_1}) = ((ssk_{\mathsf{id}_1}, dk_{\mathsf{id}_1}), spk_{\mathsf{id}_1})$		$(sk_{\mathsf{id}_2}, pk_{\mathsf{id}_2}) = ((ssk_{\mathsf{id}_2}, dk_{\mathsf{id}_2}), spk_{\mathsf{id}_2})$

Protocol Execution

$c_{\mathsf{id}_1} \xleftarrow{\$} \mathcal{C}_{\mathsf{wKEM}},\ rs_{\mathsf{id}_1} \xleftarrow{\$} \mathcal{RS}_{\mathsf{SIG}}$		
$rpg_{\mathsf{id}_1} \leftarrow \mathsf{wKEM.Dec}(dk_{\mathsf{id}_1}, c_{\mathsf{id}_1})$		
$(epk_{\mathsf{id}_1}, esk_{\mathsf{id}_1}) \xleftarrow{\$} \mathsf{OTKEM.Gen}(1^\kappa, rpg_{\mathsf{id}_1})$		
$\sigma_{\mathsf{id}_1} \leftarrow \mathsf{SIG.Sign}(ssk_{\mathsf{id}_1}, epk_{\mathsf{id}_1}, rs_{\mathsf{id}_1})$		
$m_{\mathsf{id}_1} := (\mathsf{id}_1, epk_{\mathsf{id}_1}, \sigma_{\mathsf{id}_1})$		
	$\xrightarrow{m_{\mathsf{id}_1}}$	reject if $\mathsf{SIG.Vfy}(spk_{\mathsf{id}_1}, \sigma_{\mathsf{id}_1}, epk_{\mathsf{id}_1}) \neq 1$
		$c_{\mathsf{id}_2} \xleftarrow{\$} \mathcal{C}_{\mathsf{wKEM}},\ rs_{\mathsf{id}_2} \xleftarrow{\$} \mathcal{RS}_{\mathsf{SIG}}$
		$erk_{\mathsf{id}_2} \leftarrow \mathsf{wKEM.Dec}(dk_{\mathsf{id}_2}, c_{\mathsf{id}_2})$
		$(k, C_{\mathsf{id}_2}) \xleftarrow{\$} \mathsf{OTKEM.Enc}(epk_{\mathsf{id}_1}, erk_{\mathsf{id}_2})$
		$T := \mathsf{id}_1 \| pk_{\mathsf{id}_1} \| epk_{\mathsf{id}_1} \| \sigma_{\mathsf{id}_1} \| \mathsf{id}_2 \| pk_{\mathsf{id}_2} \| C_{\mathsf{id}_2}$
		$\sigma_{\mathsf{id}_2} \leftarrow \mathsf{SIG.Sign}(ssk_{\mathsf{id}_2}, T, rs_{\mathsf{id}_2})$
	$\xleftarrow{m_{\mathsf{id}_2}}$	$m_{\mathsf{id}_2} := (\mathsf{id}_2, C_{\mathsf{id}_2}, \sigma_{\mathsf{id}_2})$
$T := \mathsf{id}_1 \| pk_{\mathsf{id}_1} \| epk_{\mathsf{id}_1} \| \sigma_{\mathsf{id}_1} \| \mathsf{id}_2 \| pk_{\mathsf{id}_2} \| C_{\mathsf{id}_2}$		
reject if $\mathsf{SIG.Vfy}(spk_{\mathsf{id}_2}, \sigma_{\mathsf{id}_2}, T) \neq 1$		
$k \leftarrow \mathsf{OTKEM.Dec}(esk_{\mathsf{id}_1}, C_{\mathsf{id}_2})$		
$\mathsf{sid} := T \| \sigma_{\mathsf{id}_2}$		$\mathsf{sid} := T \| \sigma_{\mathsf{id}_2}$
accept $K := \mathsf{PRF}(k, \mathsf{sid})$		accept $K := \mathsf{PRF}(k, \mathsf{sid})$

All protocol messages

Instantiations from Ideal Lattices

- Building blocks' instantiations from existing works:
  - **Signature (SIG):** Ruckert (PQCrypto'10)
  - **IND-CPA KEM (wKEM):** Peikert (PQCrypto'14).
  - **Pseudo-random function (PRF)**: Banrjee et al. (Eurocrypt'12)
  - **One-time KEM (OTKEM):** q-bounded IND-CCA KEM (q=1), Cramer et al. Asiacrypt'07 (less efficient)

Can we build efficient OTKEM from ideal lattices?

Efficient OTKEM from Ideal Lattices

- Direct construction
  - **Ring-Learning with Errors (RLWE):**

$$a \in_R R_q, (s, e) \in_R \mathcal{X}, V_0 := a \cdot s + e, V_1 \in_R$$

  - **Target collision resistant hash function (TCRHF):**

$$\mathsf{TCRHF} : hk_{\mathsf{TCRHF}} \times R_q \to \{0,1\}^n$$

Similar to construction of OTS from OWF

Thank you very much for your attention!