# FIELD-SWITCHING IN HOMOMORPHIC ENCRYPTION


Craig Gentry
Shai Halevi
Chris Peikert
Nigel P. Smart

# HE Over Cyclotomic Rings

- Denote the field $K_m = Q(\zeta_m) \cong Q[X]/(\Phi_m(X))$
  - Its ring of integers is $R_m = Z(\zeta_m) \cong Z[X]/(\Phi_m(X))$
  - Mod-$q$ denoted $R_{m,q} = R_m/qR_m \cong Z_q[X]/(\Phi_m(X))$
- "Native plaintext space" is $R_{m,2}$
- Ciphertexts*, secret-keys are vectors over $R_{m,q}$
- $\boldsymbol{c}$ wrt $\boldsymbol{s}$ encrypts $a$ if (for representatives in $R_m$) we have $\langle \boldsymbol{s}, \boldsymbol{c} \rangle = a \cdot \frac{q}{2} + e \ (mod\ q)^*$ for small $e$
  - Decryption via $a := MSB(\langle \boldsymbol{s}, \boldsymbol{c} \rangle)^*$
  - Using "appropriate" $Z$-bases of $R_{m,2}, R_{m,q}$

* Not exactly

# HE Over Cyclotomic Rings

- "Native plaintexts" encode vectors of values
  - $a \in R_{m,2} \rightarrow (\alpha_1 \ldots \alpha_\ell) \in GF(2^d)^\ell$ (more on that later)
- Homomorphic Operations
  - Addition: $\boldsymbol{c} \boxplus \boldsymbol{c}'$ encrypts $a + a' \in R_{m,2}$, encoding $(\alpha_1 + \alpha_1' \ldots \alpha_\ell + \alpha_\ell')$
  - Multiplication: $\boldsymbol{c} \boxtimes \boldsymbol{c}'$ encrypts $a \times a' \in R_{m,2}$, encoding $(\alpha_1 \times \alpha_1' \ldots \alpha_\ell \times \alpha_\ell')$
  - Automorphism: $\boldsymbol{c}(X^t)$ encrypts $a(X^t) \in R_{m,2}$, encoding some permutation of $(\alpha_1 \ldots \alpha_\ell)$
    - Relative to key $\boldsymbol{s}(X^t)$

# HE Over Cyclotomic Rings

- Also a key-switching operation
- For any two $\mathbf{s}, \mathbf{s}' \in (R_{m,q})^2$ we can publish a key-switching gadget $W[\boldsymbol{s} \rightarrow \boldsymbol{s}']$
- $W$ used to translate valid $\mathbf{c}$ wrt $\mathbf{s}$ into $\mathbf{c}'$ wrt $\mathbf{s}'$
  - $\mathbf{c}, \mathbf{c}'$ encrypt the same plaintext

$$\langle \boldsymbol{s}, \boldsymbol{c} \rangle = \langle \boldsymbol{s}', \boldsymbol{c}' \rangle + e \ (mod\ q)$$

for some small $e$

# How Large are $m, q$?

- Ciphertexts are "noisy" (for security)
  - noise grows during homomorphic computation
  - Decryption error if noise grows larger than $q$

➔ Must set $q$ "much larger" than initial noise

➔ Security relies on LWE-hardness with very large modulus/noise ratio

➔ Dimension ($m$) must be large to get hardness

- Asymptotically $|q| = polylog(k), m = \tilde{\Omega}(k)$
  - For realistic settings, $|q| \approx 1000, m > 10000$

# Switching to Smaller $m$?

- As we compute, the noise grows
  - Cipehrtexts have smaller modulus/noise ratio
  - From a security perspective, it becomes permissible to switch to smaller values of $m$
- How to do this?
- Not even clear what outcome we want here:
  - Have $\boldsymbol{c}$ wrt $\boldsymbol{s} \in (R_{m,q})^2$, encrypting some $a \in R_{m,2}$
  - Want $\boldsymbol{c}'$ wrt $\boldsymbol{s}' \in (R_{m',q})^2$ for $m' < m$
    - Encrypting $a' \in R_{m',2}$ ??

# Ring-Switching: The Goal

- We cannot get $a' = a$ since $a' \in R_{m',2}$, $a \in R_{m,2}$
- We want $a'$ to be "related" to $a$
  - $a \in R_{m,2}$ encodes $(\alpha_1 \dots \alpha_\ell) \in GF(2^d)^\ell$
  - $a' \in R_{m',2}$ encodes $(\alpha'_1 \dots \alpha'_{\ell'}) \in GF(2^{d'})^{\ell'}$
- May want $a'$ to encode a subset of the $\alpha_i$'s?
  - E.g., the first $\ell'$ of them
  - Not always possible, only if $d' = d$
- What relations between the ${\alpha'}_j, \alpha_i$'s are possible?

# Prior Work

- A limited ring-switching technique was described in [BGV'12]
  - Only for $m = 2^n, m' = 2^{n-1}$
- Transforms big-ring $\mathbf{c}$ into small-ring $\boldsymbol{c}_1', \boldsymbol{c}_2'$ s.t. $a$ (encrypted in $\mathbf{c}$) can be recovered from $a_1', a_2'$ (encrypted in $\boldsymbol{c}_1', \boldsymbol{c}_2'$).
- Used only for bootstrapping

# Our Transformation: Overview

- Work for any $m, m'$ as long as $m'|m$
- $\mathbf{c}$ wrt $\mathbf{s} \in (R_{m,q})^2$ ➔ $\mathbf{c}'$ wrt $\mathbf{s}' \in (R_{m',q})^2$
- $\mathbf{c}, \mathbf{c}'$ encrypt $a, a'$, that encode vectors:
  - $\boldsymbol{c} \rightarrow (\alpha_i) \in GF(2^d)^\ell$, $\mathbf{c}' \rightarrow (\alpha'_j) \in GF(2^{d'})^{\ell'}$
  - Necessarily $d'|d$, so $GF(2^{d'})$ a subfield of $GF(2^d)$
- Each $\alpha'_j$ is a $GF(2^{d'})$-linear function of some $\alpha_i$'s
  - We can choose the linear functions, but not the subset of $\alpha_i$'s that correspond to each $\alpha'_j$
  - If $d' = d$, can use projections (so $\alpha'_j$'s a subset of $\alpha_i$'s)

# Our Transformation: Overview

Denote $K = K_m, R = R_m,\ K' = K_{m'}, R' = R_{m'}$

1. Key-switching to map $\boldsymbol{c}$ wrt $\boldsymbol{s}$ ➔ $\boldsymbol{c}''$ wrt $\boldsymbol{s}'$
   - $\boldsymbol{s} \in R_q^2$ and $\boldsymbol{s}' \in {R'_q}^2 \subset R_q^2$
   - $\boldsymbol{c}'' = (c_0'', c_1'')$ over the big field, wrt subfield key
2. Compute a small $r \in R_q$ that depends only on the desired linear functions
3. Apply the trace function, $c_i' = \mathrm{Tr}_{K/K'}(r \cdot c_i'')$
4. Output $\boldsymbol{c}' = (c_0', c_1')$

# Algebra

# Geometry of $K$

- Use canonical-embedding to associate $u \in K$ with a $\phi(m)$-vector of complex numbers
  - Thinking of $u = u(X)$ as a polynomial, associate $u$ with the vector $\sigma(u) = \left(u(\rho^i)\right)_{i \in Z_m^*}$
    - $\rho = e^{2\pi i/m}$, the principal complex $m$'th root of unity
    - E.g., if $u \in Q \subset K$ then $\sigma(u) = (u, u, \ldots, u)$
- We can talk about the "size of $u$"
  - say the $l_2$ or $l_\infty$ norm of $\sigma(u)$
  - For decryption, the "noise element" must be $\ll q$

# Geometry of $K, K'$

- $K$ can be expressed as a vector-space over $K'$
  - Similarly $R$ over $R'$, $R_q$ over $R'_q$ , etc.
- Every $R'$ -basis $B$ induces a transformation

  $T_B$: coefficients in $R'$ $\mapsto$ element of $R$
  - With canonical embedding on both sides, we have a $C$-linear transformation $T_B: C^{\phi(m)} \to C^{\phi(m)}$
- We want a "good basis", where $T_B$ is "short" and "nearly orthogonal"

# Geometry of $K, K'$

- Lemma 1: There exists $R'$-basis $B$ of R for which all the singular values of $T_B$ are nearly the same.
  - Specifically $s_1(T) = s_n(T) \cdot \sqrt{f}$ where $f \leq \frac{rad(m)}{rad(m')} = \prod$ primes that divide $m$ but not $m'$
- The proof follows techniques from [LPR13], the basis $B$ is essentially a tensor of DFT matrices

# The Trace Function

- For $u \in K$, $\mathrm{Tr}(u) = \sum_{i \in Z_m^*} \sigma(u)_i \in Q$
  - By definition: if $u$ is small then so is $\mathrm{Tr}(u)$
- $\mathrm{Tr}: K \to Q$ is $Q-$linear
  - $L: K \to Q$ is $Q$-linear if $\forall u, v \in K, q \in Q$,
    $L(u) + L(v) = L(u + v)$ and $L(q \cdot u) = q \cdot L(u)$
- The trace is a "universal" $Q$-linear function:
  - For every $Q$-linear function $L$ there exists $\kappa \in K$ such that $L(u) = \mathrm{Tr}(\kappa \cdot u)\ \forall u \in K$

# The Trace Function

- The trace Implies also a $Z$-linear map $\mathrm{Tr}: R \to Z$, and $Z_q$-linear map $\mathrm{Tr}: R_q \to Z_q$
- Every $Z$-linear map $\mathrm{L} : R \to Z$ can be written as $L(a) = \mathrm{Tr}(\kappa \cdot a)$
  - But $\kappa$ need not be in $R$
  - More on that later

# The Intermediate Trace Function

- $Tr_{K/K'}: K \to K'$ when $K$ is an extension of $K'$
  - Satisfies $Tr_{K/Q} = Tr_{K/K'} \circ Tr_{K'/Q}$
- <u>Lemma 2:</u> if $u$ is small then so is $\mathrm{Tr}_{K/K'}(u)$
  - Less trivial than for $\mathrm{Tr}_{K/Q}$ but still true
- $\mathrm{Tr}_{K/K'}$ is a "universal" $K'$-linear function:
  - $\mathrm{Tr}_{K/K'}: K \to K'$ is $K'-$linear
  - For every $K'$-linear function $L$ there exists $\kappa \in K_m$ such that $L(u) = \mathrm{Tr}_{\mathrm{K/K'}}(\kappa \cdot u)\ \forall u \in K_m$
- Similarly implies $R'$-linear map $Tr_{K/K'}: R \to R'$ and $R'_q$-linear map $Tr_{K/K'}: R_q \to R'_q$

# Some Complications

- Often we get $\mathrm{Tr}_{K/K'}(R) \subsetneq R'$
- Also for many linear functions we get $L(u) = \mathrm{Tr}_{\mathrm{K}/\mathrm{K}'}(\kappa \cdot u)$ where $\kappa$ is not in $R$
- In our setting this will cause problems when we apply the trace to ciphertext elements
  - That's (one reason) why ciphertexts are not really vectors over $\mathrm{R}$
  - Hence the *'s throughout the slides

# The Dual of $R$

- Instead of $R$, ciphertext are vectors over the dual $R^\vee = \{a \in K : \forall\, r \in R, \mathrm{Tr}(ar) \in Z\}$
  - $\mathrm{R}^\vee = \mathrm{R}/\mathrm{t}$, $\mathrm{R}'^\vee = \mathrm{R}'/\mathrm{t}'$ for some $\mathrm{t} \in R, t' \in R'$
- We have $\mathrm{Tr}_{K/K'}(\mathrm{R}^\vee) = {R'}^\vee$
  - Also every $\mathrm{R}'$-linear $L: R^\vee \to R'^\vee$ can be written as $L(a) = Tr_{K/K'}(r \cdot a)$ for some $r \in R$
- In the rest of this talk we ignore this point, and pretend that everything is over $R$

# Prime Splitting

- The integer 2 splits over $R$ as $2 = \prod_i \boldsymbol{p}_i^e$
  - $i$ ranges over $G = Z_m^*/(2)$
  - $\boldsymbol{p}_i$ is generated by $\left(2, F_i(X) = \prod_j \left(X - \zeta_m^{i\cdot 2^j}\right)\right)$
  - In this talk we assume $e$=1 (i.e., $m$ is odd)
  - $\ell = |G|$ prime ideals, each $R/\boldsymbol{p}_i \cong GF(2^d)$
  - $\mathrm{R}_2 = R/(2) \cong \bigoplus_i R/\boldsymbol{p}_i \cong \bigoplus_i GF(2^d)$
- Using CRT, each $a \in R_2$ encodes the vector

$$(\underbrace{a \bmod \boldsymbol{p}_{i_1}}_{\alpha_1}, \ldots, \underbrace{a \bmod \boldsymbol{p}_{i_\ell}}_{\alpha_\ell}) \in GF\left(2^d\right)^\ell$$

# Prime Splitting

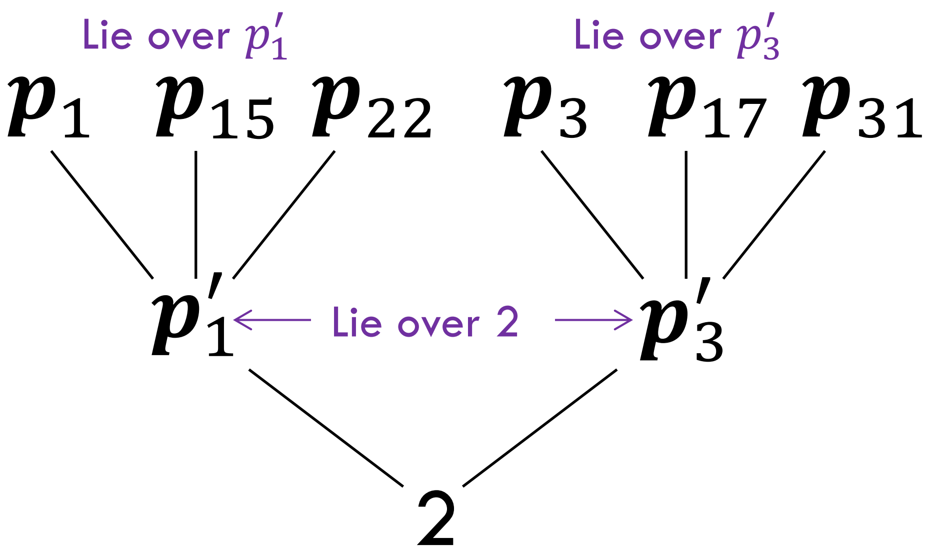
- Similarly 2 splits over $R'$ as $2 = \prod_j \boldsymbol{p}'_j{}^{e'}$
  - Again we assume $e' = 1$
  - Using CRT, each $a' \in R'_2$ encodes the vector

$$(\underbrace{a' \bmod \boldsymbol{p}'_{j_1}}_{\alpha'_1}, \dots, \underbrace{a' \bmod \boldsymbol{p}'_{j_{\ell'}}}_{\alpha'_\ell}) \in GF\left(2^{d'}\right)^{\ell'}$$

- When $m'|m$ then also $d'|d$, $\ell'|\ell$, and each $\boldsymbol{p}'_j$ split over $R$ as a product of some of the $\boldsymbol{p}_i$ 's

# Prime Splitting

- Example for $m = 91, m' = 7$




$$\begin{array}{ll} R & d = 12 \\ \cup\!| & \ell = 6 \\ R' & d' = 3 \\ \cup\!| & \ell' = 2 \\ Z & \end{array}$$

# Plaintext-Slot Representation

- Recall that $R/\boldsymbol{p}_i \cong GF(2^d)$ for all the $\boldsymbol{p}_i$'s
  - But the isomorphisms are not unique
- To fix the isomorphisms:
  - Fix a primitive $m$-th root of unity $\omega \in GF(2^d)$
  - Fix representatives $u_i \in Z_m^*$ for all $i \in Z_m^*/(2)$
  - $h_i: R/\boldsymbol{p}_i \to GF(2^d)$ defined via $h_i(\zeta_m) = \omega^{u_i}$
- Same for isomorphisms $R'/\boldsymbol{p}'_j \cong GF(2^{d'})$
  - Define $h'_j: R'/\boldsymbol{p}'_j \to GF(2^{d'})$ by fixing $\rho'$ and $u'_j$

# Plaintext-Slot Representation

- Making the $h_i$'s and $h'_j$'s "consistent"
  - Fix $\omega \in GF(2^d)$ and set $\omega' = \rho^{m/m'} \in GF(2^{d'})$
  - Fix $u'_j \in j \cdot (2) \subset Z^*_{m'}$ $\forall j$, then $\forall$ $\boldsymbol{p}_i$ that lies over $\boldsymbol{p}'_j$, choose $u_i \in i \cdot (2)$ s.t. $u_i = u'_j \bmod m'$

- Fact: if $\boldsymbol{p}_i$ lies over $\boldsymbol{p}'_j$ and $r' \in R' \subset R$, then

$$h_i(r' \bmod \boldsymbol{p}_i) = h'_j(r' \bmod \boldsymbol{p}'_j) \in GF(2^{d'})$$

  - In words: for a sub-ring plaintext, the slots mod $\boldsymbol{p}'_j$ and all the $\boldsymbol{p}_i$'s lie over it, hold the same value

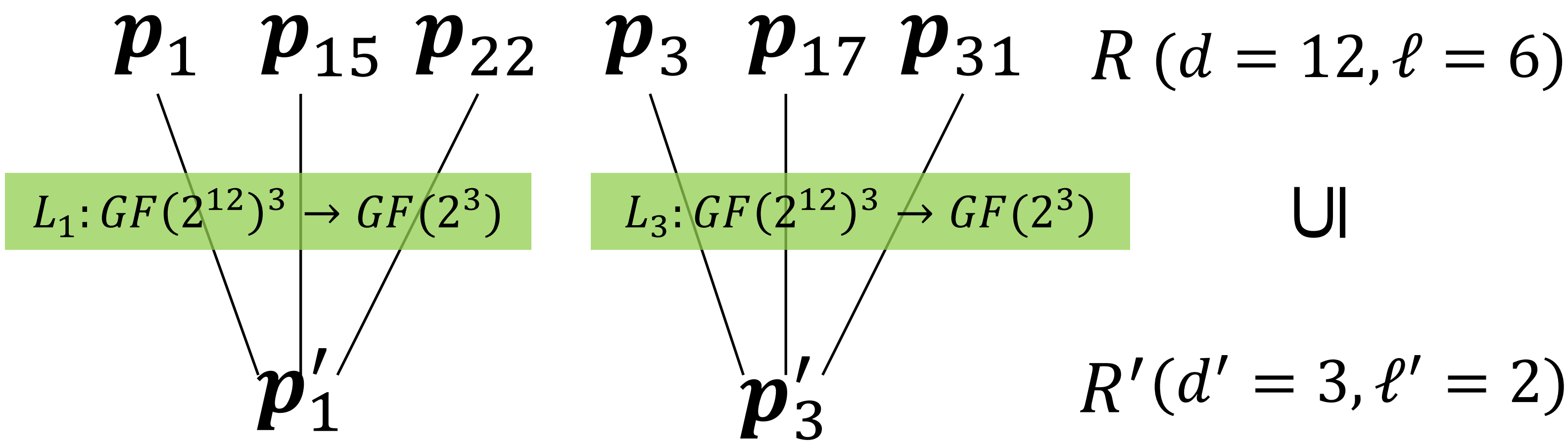
# Plaintext-Slot Representation

- Lemma 3: $\forall$ collection of $GF(2^{d'})$-linear functions
$\left\{L_j\colon\ GF\left(2^d\right)^{\frac{\ell}{\ell'}} \to\ GF\left(2^{d'}\right)\right\}_{j\in Z^*_{m'}/2}$
$\exists$ a unique $R'_2$-linear function $L\colon R_2 \to R'_2$ s.t.

$$h'_j(a' \bmod \boldsymbol{p}'_j) = L_j((h_i(a \bmod \boldsymbol{p}_i))_i)$$

holds $\forall\ a \in R_2$ and $a' = L(a)$, and $\forall\ j$
  - The $i$'s range over all the $\boldsymbol{p}_i$'s that lie over $\boldsymbol{p}'_j$

# Illustration of Lemma 3




- $\exists L: R_2 \to R'_2$ s.t. $\forall\, a \in R_2$ and $a' = L(a) \in R'_2$
  - $h'_1(a') = L_1(h_1(a), h_{15}(a), h_{22}(a))$
  - $h'_3(a') = L_2(h_3(a), h_{17}(a), h_{31}(a))$
- Can express $L(a) = Tr_{K/K'}(r \cdot a)$ for some $r \in R_2$*

* Not exactly

# The Transformation

# Step 1, Key Switching

- Let $\boldsymbol{s} \in R_q^2, \boldsymbol{s}' \in {R'_q}^2 \subset R_q^2$ (chosen at keygen)
- Publish a key-switching matrix $W[\boldsymbol{s} \rightarrow \boldsymbol{s}']$
- Given ctxt $\boldsymbol{c}$ wrt $\boldsymbol{s}$, use W to get $\boldsymbol{c}''$ wrt $\boldsymbol{s}'$
  - Just plain key-switching in the big ring
  - $\boldsymbol{c}''$ still over the big ring, but wrt a sub-ring key
  - $\boldsymbol{c}''$ encrypts the same $R_2$-element as $\boldsymbol{c}$

# Security of Key-Swicthing

- Security of usual big-ring key-switching relies on the secret $\boldsymbol{s}'$ being drawn from $R_q$
  - Then $W$ constrains only LWE-instance over $R_q$
  - What can we say when it is drawn from $R'_q$?
- We devise LWE instances over $R_q$ with secret from $R'_q$, with security relying on LWE in $R'_q$
  - Instead of one small error element in $R_q$, choose many small elements in $R'_q$, use an $R'_q$-basis of $R_q$ to combine them into a single error element in $R_q$

# $R_q$-LWE With Secret in $R'_q$

- Let $B = (\beta_1, \ldots, \beta_n)$ be any $R'_q$-basis of $R_q$
- Given the LWE secret $s' \in R'_q \subset R_q$
  - Choose uniform $a \leftarrow R_q$ and small $e'_1, \ldots, e'_n \leftarrow R'_q$
  - Set $e = \sum_i e'_i \beta_i \in R_q$ and output $(a, b = as' + e)$
- If the basis B is "good" (short, orthogonal) then $e$ is not much larger than the $e'_i$'s
  - This is where we use Lemma 1 ($\exists$ good basis)

# $R_q$-LWE With Secret in $R'_q$

- <u>Theorem</u>: If decision-LWE is hard in $R'_q$, then $(a, b)$ is indistinguishable from uniform in $R_q^2$
- Proof:
  - We can consider $a = \sum_i a'_i \beta_i$ for uniform $a'_i \leftarrow R'_q$
    - Induces the same uniform distribution on $a$
  - Then we would get $b = \sum_i (a'_i s' + e'_i) \beta_i$.
  - If the $(a'_i s' + e'_i)$ were uniform in $R'_q$, then $b$ would be uniform in $R_q$. □

# Steps 2,3: Ring Switching

- $\boldsymbol{c}''$ encrypts $a \in R_2$ wrt $\boldsymbol{s}'$
  - $a$ encodes a vector $\boldsymbol{\alpha} = (\alpha_i)_i \in GF\left(2^d\right)^\ell$
  - We view it as $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_{\ell'}) \in \left(GF\left(2^d\right)^{\ell/\ell'}\right)^{\ell'}$
- $\ell'$ target functions, $L_j: GF\left(2^d\right)^{\ell/\ell'} \to GF(2^{d'})$
  - Want small-ring ciphertext $\boldsymbol{c}'$ encrypting $a \in R_2'$ that encodes $\boldsymbol{\alpha}' = \left(\alpha_1', \dots, \alpha_{\ell'}'\right) \in GF\left(2^{d'}\right)^{\ell'}$
  - For each $j$, $\alpha_j' = L_j(\boldsymbol{\alpha}_j)$

# Steps 2,3: Ring Switching

- By Lemma 2, $\exists L: R_2 \to R_2'$ that induces the $L_j$'s
  - Expressed as $L(a) = Tr_{K/K'}(r \cdot a)$ for $r \in R_2'$*
  - We identify $r$ with a short representative in $R'$
    - One must exists since 2 is "short"
    - Thus identify $L$ with $L(a) = Tr_{K/K'}(r \cdot a)$ over $R$
  - Further identify $r$ as a representative of $r \in R_q'$
- Apply the trace, $c_i' = Tr_{K/K'}(r \cdot c_i'')$
  - Recall that $\boldsymbol{c}''$ is valid wrt $\boldsymbol{s}' \in R_q' \subset R_q$

* Not exactly

# Correctness

- Recall $\langle \mathbf{s}', \mathbf{c}'' \rangle = k \cdot q + \ a \cdot \frac{q}{2} + e$ over $K$
  - For some $k, e \in R$ (with $e$ small) and $\mathbf{s}'$ over $R'$
- Thus we have the equalities (over $K$):
  - $\langle \mathbf{s}', \mathbf{c}' \rangle = \langle \mathbf{s}', Tr_{K/K'}\,(r \cdot \mathbf{c}'') \rangle = Tr_{K/K'}\,(r \cdot \langle \mathbf{s}', \mathbf{c}'' \rangle)$
    $= L\left(q \cdot k + \ a \cdot \frac{q}{2} + e\right) = L(k) \cdot q + L(a) \cdot \frac{q}{2} + L(e)$
    $= k' \cdot q + a' \cdot \frac{q}{2} + e'$
  - $a'$ encodes the $\alpha_j'$'s that we want

# Correctness

- We have $\langle \boldsymbol{s}', \boldsymbol{c}' \rangle = k' \cdot q + a' \cdot \frac{q}{2} + e'$
  - This looks like a valid encryption of $a'$
  - It remains to show that $e'$ is short
- $e' = L(e) = Tr_{K/K'}(r \cdot e)$
  - $e$ is short (from the input), $r$ is short (reduced mod 2)
  - So $r \cdot e$ is short
  - By Lemma 3 also $Tr_{K/K'}(r \cdot e)$ is short

# Conclusions

- We have a general ring-switching technique
  - Converts $\boldsymbol{c}$ over $R_m$ to $\boldsymbol{c}'$ over $R_{m'}$ for $m'|m$
  - The plaintext slots in $\boldsymbol{c}'$ can contain any linear functions of the slots in $\boldsymbol{c}$
    - A $\boldsymbol{c}'$-slot is a function of the $\boldsymbol{c}$-slots that lie above it
  - We may choose projection functions to have $\boldsymbol{c}'$ contain subset of the slots of $\boldsymbol{c}$
- Lets us to speed up computation by switching to a smaller ring

# Epilog: The [AP13] Work

Alperin-Sheriff & Peikert described a clever use of ring-switching for efficient homomorphic computation of DFT-like transformations:

1. Decompose it to an FFT-like network of “local” linear functions
2. Use ring-switching for each level
3. Then switch back up before the next level

Yields fastest bootstrapping procedure to date