Ring-LWE security in the case of FHE

Guillaume Bonnoron, Caroline Fontaine

Chair of Naval Cyber Defense

5 July 2016
Workshop HEAT – Paris

ECOLE NAVALE THALES DCNS TELECOM Bretagne Institut Mines-Télécom

Why worry?

Which algorithm performs best depends on the concrete parameters considered.

For small n, DEC may be favourable. For large n, BKW may be fastest when considering PKE but not when considering HE schemes which require large q

Albrecht, Player, Scott, '15

Roadmap

1. Definitions: Ring-LWE and HE schemes
2. Our special-purpose attack
3. Some experimental results

Ring-Learning With Errors

Parameters:

- n, q positive integers
- R a ring of degree n over $\mathbb{Z}$
  e.g. $R = \mathbb{Z}[x]/(f(x))$ with $f(x)$ cyclotomic
  R_q denotes the ring of degree $n-1$ polynomials with coefficients in $[0, q-1]$
- χ probability distribution over R with std deviation σ

Problem: given $s \in R_q$, we sample several

- $a_i \leftarrow \mathcal{U}(R_q)$
- $e_i \leftarrow \chi$

and provide to the attacker the pairs: $(a_i, [a_i s + e_i]_q)$

She aims at recovering s.

FHE + Ring-LWE = FV

Proposal from Fan-Vercauteren (2012)

KeyGen

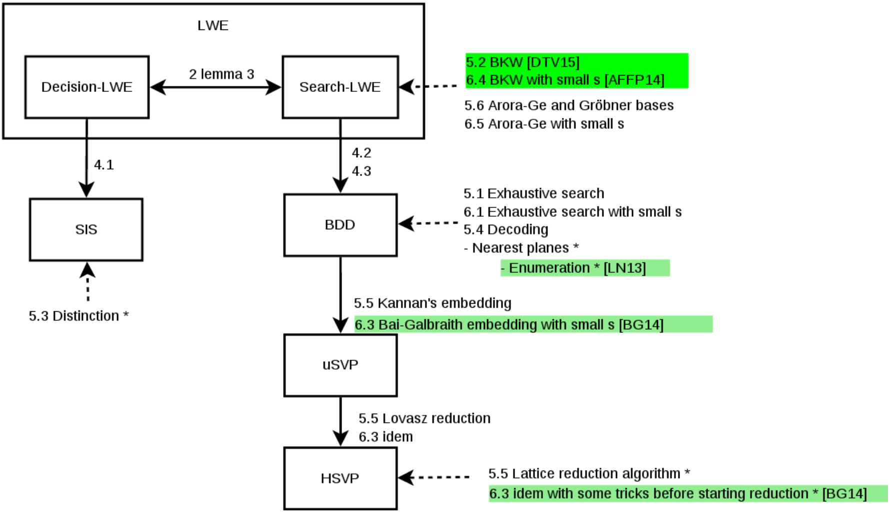
1. $\texttt{FV.ES.SecretKeyGen}(n, \sigma, q)$:
   Sample $\mathbf{s} \leftarrow R_2$ and return $\texttt{sk} = \mathbf{s}$
2. $\texttt{FV.ES.PublicKeyGen(sk)}$:
   With $\mathbf{s} = \texttt{sk}$, sample $\mathbf{a} \leftarrow \mathcal{U}(R_q)$, $\mathbf{e} \leftarrow \chi$ and return

$$\texttt{pk} = ([-(\mathbf{a} \cdot \mathbf{s} + \mathbf{e})]_q, \mathbf{a})$$

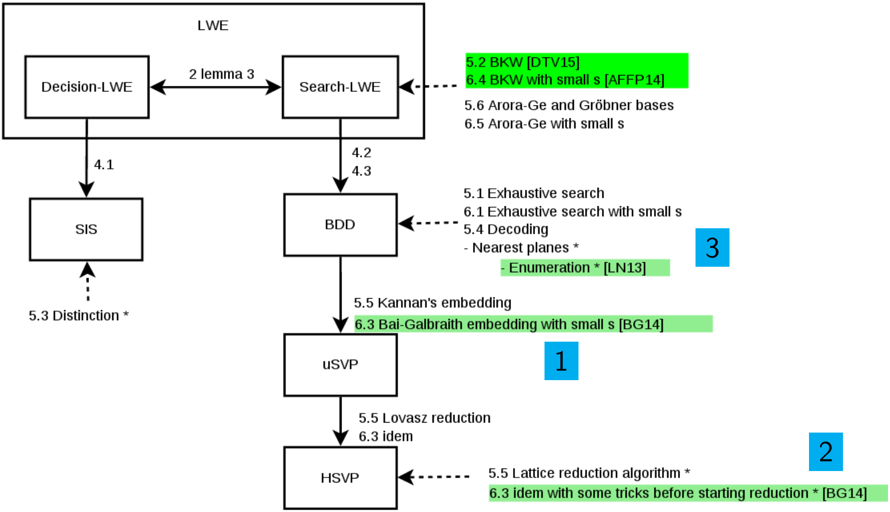
The public key pk is, up to the sign, a Ring-LWE sample. With properties:

- $R = \mathbb{Z}[x]/(x^n + 1)$
- σ minimum, $\sigma = 2\sqrt{n}$
- $||s|| \leq 1$

Attack – State-of-the-art [Albrecht-Player-Scott, 15]



Attack – State-of-the-art [Albrecht-Player-Scott, 15]



Special-purpose attack – High-level steps

1. FV key $\rightarrow$ lattice
2. Embedding [Bai-Galbraith, 14]
3. Lattice reduction [LLL, 82], [BKZ, 94]
4. Enumeration for BDD [Liu-Nguyen, 13]

Special-purpose attack – Step 1

- $([-(\mathbf{a} \cdot \mathbf{s} + \mathbf{e})]_q, \mathbf{a}) \rightarrow (\mathbf{A}, \mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e} \bmod q)$

$$\text{avec } \mathbf{A}^T = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & a_n \\ -a_n & a_1 & a_2 & \cdots & a_{n-1} \\ -a_{n-1} & -a_n & a_1 & \cdots & a_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -a_2 & -a_3 & -a_4 & \cdots & a_1 \end{pmatrix}$$

- Rewriting: $\mathbf{b} = \mathbf{A}' \left(\begin{smallmatrix} \mathbf{s} \\ \mathbf{e} \end{smallmatrix} \right) \bmod q$ with $\mathbf{A}' = (\mathbf{A}|\mathbf{I_n})$

Particular solution: $\mathbf{w} = \left(\begin{smallmatrix} 0 \\ \mathbf{b} \end{smallmatrix} \right) \longrightarrow \mathbf{A}'\mathbf{w} = \mathbf{b} \bmod q$
Not the one expected...

Special-purpose attack – Step 2

In $\mathcal{L}' = \left\{ \mathbf{v} \in \mathbb{Z}^{2n} : \mathbf{A}'\mathbf{v} = 0 \bmod q \right\}$, we want to approximate $\mathbf{w}$.

Let $\mathbf{v_0} \in \mathcal{L}'$ be the closest to $\mathbf{w}$, the difference $\mathbf{w} - \mathbf{v_0}$ is small and $\mathbf{A}'(\mathbf{w} - \mathbf{v_0}) = \mathbf{b} \bmod q$

By embedding we get a basis of $\mathcal{L}'$

$$\mathbf{A}^T \longrightarrow \mathbf{B}^T = \begin{pmatrix} \mathbf{I_n} & \mathbf{0} \\ -\mathbf{A} & q\mathbf{I_n} \end{pmatrix} \in \mathbb{Z}^{2n \times 2n}$$

It remains to solve BDD in $\mathcal{L}'$ for the point $\mathbf{w} = \binom{0}{\mathbf{b}}$

Special-purpose attack – Step 3

To hope to solve BDD, we need a *good* basis of the lattice

- Several quality conditions
  - size : $\forall i < j$, $||(\mathbf{b_j}|\mathbf{b_i^\star})|| \leq \eta \,.\, ||\mathbf{b_i^\star}||^2$
  - LLL : size-reduced and

$$\forall i, \delta||\mathbf{b_i^\star}||^2 \leq \left(||\mathbf{b_{i+1}^\star}||^2 + \frac{(\mathbf{b_{i+1}}|\mathbf{b_i^\star})^2}{||\mathbf{b_i}||^{\star 2}} \right)$$

  - BKZ : LLL-reduced and
    For all j, $\mathbf{b_j^\star}$ is the shortest vector of the sub-lattice generated by $(\mathbf{b_j}, \ldots, \mathbf{b_k})$ with $k = \min(j + \beta - 1, n)$
- Several algorithms:
  - LLL, polynomial time
  - BKZ, better quality

Usually, lattice reduction behavior drives the parameter choice.

→ In our experiments, weak LLL reduction was sufficient.

Special-purpose attack – Step 4

Algorithms for BDD

- Nearest Plane(s) [Babai, 1986], [Lindner-Peikert, 2010]
- Pruned enumeration [Liu-Nguyen, 2013]

Idea

- Construct the solution component by component
- At each depth, bound the distance between the current (partial) solution and the target

Heuristic complexity, very good in practice

Results – Some benchmarks

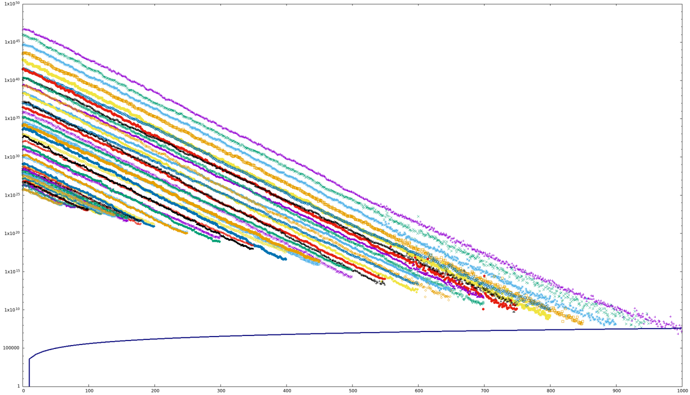
Our attack...

- works!
- has its cost dominated by the reduction step, polynomial
- lasts 29 hours for $(n, \sigma, q) = (320, 34, 2^{68})$

Have we broken FHE or Ring-LWE security?

- Does it scale up?
- Does it somehow work in other settings?

Results – Does it scale?



Bigger n implies

- Bigger error
- Smaller smallest GS coefficient

Conclusion

How good is this attack?

- We broke ($n = 320$, $\sigma = 34$, $q = 2^{68}$) in 1 day.
- Estimator from [APS15] predicts one month of computation.
- Last year [LL15] broke ($n = 350$, $\sigma = 8$, $q = 2^{52}$) in 3.5 days

We still need more cryptanalysis, especially in specific settings!

Thanks for your attention!