Polly Cracker, Revisited

Martin Albrecht[1] Pooya Farshim[2] Jean-Charles Faugère[1]
Ludovic Perret[1]

1 SALSA Project - INRIA, UPMC, Univ Paris 06
2 Information Security Group, Royal Holloway, University of London

MAYA Workshop
New Applications of New Computational Problems
25. May 2011

Outline

Outline

Notation & Definitions I

- $P = \mathbb{F}[x_1, \ldots, x_n]$ with some ordering on monomials.
- $P_{\leq b}$ elements in P of degree at most b.
- $\mathrm{LM}(f)$ is the leading monomial appearing in $f \in P$.
- $\mathrm{LC}(f)$ is the coefficient corresponding to $\mathrm{LM}(f)$ in f.
- $\mathrm{LT}(f)$ is $\mathrm{LC}(f)\mathrm{LM}(f)$.
- d is the degree of Gröbner bases in this talk.
- b is the degree of random ideal elements in this talk.

Notation & Definitions II

An example in $\mathbb{F}[x, y, z]$ with term ordering **deglex**:

$$f = 3yz + 2x + 1$$

- $\mathrm{LM}(f) = yz$,
- $\mathrm{LC}(f) = 3$ and
- $\mathrm{LT}(f) = 3yz$.

Notation & Definitions III

Definition (Generated Ideal)

Let $f_1, \ldots, f_m$ be polynomials in P. Define the set

$$\langle f_1, \ldots, f_m \rangle := \left\{ \sum_{i=1}^{m} h_i f_i : h_1, \ldots, h_m \in P \right\}.$$

This set $\mathcal{I}$ is an ideal called the ideal generated by $f_1, \ldots, f_m$.

Notation & Definitions IV

Definition (Gröbner Basis)

Let $\mathcal{I}$ be an ideal of $\mathbb{F}[x_1, \ldots, x_n]$ and fix a monomial ordering. A finite subset

$$G = \{g_1, \ldots, g_m\} \subset \mathcal{I}$$

is said to be a **Gröbner basis** of $\mathcal{I}$ if for any $f \in \mathcal{I}$ there exists $g_i \in G$ with

$$\mathrm{LM}(g_i) \mid \mathrm{LM}(f).$$

Notation & Definitions V

For each ideal $\mathcal{I}$ and monomial ordering there is a unique **reduced** Gröbner basis which can be computed in polynomial time from any Gröbner basis.

Gröbner bases allow to compute remainders modulo $\mathcal{I}$:

$$f \mod \mathcal{I} = f \mod G.$$

Informally speaking, if you know the (reduced) Gröbner basis you "understand" the ideal.

Generating Gröbner bases for Crypto I

Definition (S-Polynomial)

The S-polynomial of f and g is defined as

$$S(f,g) = \frac{\mathrm{LCM}(\mathrm{LM}(f),\mathrm{LM}(g))}{\mathrm{LT}(f)} \cdot f - \frac{\mathrm{LCM}(\mathrm{LM}(f),\mathrm{LM}(g))}{\mathrm{LT}(g)} \cdot g.$$

Theorem

A basis $G = \{g_1, \ldots, g_s\}$ for an ideal $\mathcal{I}$ is a Gröbner basis if and only if all $S(g_i, g_j)$ reduce to zero by polynomial division.

Generating Gröbner bases for Crypto II

1 **begin**
2 **for** $1 \leq i \leq n$ **do**
3 **for** $0 \leq j < |M_{<x_i^d}|$ **do**
4 $c_{ij} \leftarrow_\$ \mathbb{F}_q$;
5 $g_i \leftarrow x_i^d + \sum_j c_{ij} m_j$;
6 **return** $(g_1, \ldots, g_n)$;
7 **end**

Algorithm 1: $\mathsf{GBGen}_{\mathsf{dense}}(1^\lambda, P, d)$

Theorem

Let $f, g \in \mathbb{F}[x_0, \ldots, x_{n-1}]$ *with* $a = \mathrm{LM}(f)$ *and* $b = \mathrm{LM}(g)$ *and*

$$\mathrm{LCM}(a, b) = a \cdot b.$$

Then

$$S(f, g) \xrightarrow[\{f,g\}]{} 0.$$

Sampling Elements in $\mathcal{I}$

1 **begin**
2 $\quad f \leftarrow_{\$} P_{\leq b}$;
3 $\quad f \leftarrow f - f \mod G$;
4 $\quad$ **return** f;
5 **end**

Algorithm 2: **Sample**()

This sampling is uniform for elements $f \in \mathcal{I}$ with $\deg(f) \leq b$ because $P = \mathcal{I} \oplus P/\mathcal{I}$.

Classical Computational Problems I

GB Given access to m samples from $\mathcal{I}$ recover G.

IR Given access to m samples from $\mathcal{I}$ and a challenge $f \in P$, recover $f \mod G$.

IM Given access to m samples from $\mathcal{I}$ and a challenge $f \in P$, decide if $f \mod G = 0$.

Hardness I

Lemma (IR <=> GB)

If we have an oracle which solves the IR *problem, we can construct an algorithm which solves the* GB *problem and vice versa.*

Proof for first direction.

Consider an arbitrary element g_i in the Gröbner basis G. We can write g_i as $m_i + \tilde{g}_i$ for some $\tilde{g}_i < g_i$ and $m_i = \mathrm{LM}(g_i)$.

Now, assume the normal form of m_i is r_i. Hence, $m_i = \sum_{j=1}^{n} h_j g_j + r_i$ for some $h_i \in P$. Thus, $m_i - r_i \in \langle G \rangle$ with leading monomial m_i.

Repeat this process for all monomials up to and including degree d and accumulate the results $m_i - r_i$ in a list $\tilde{G}$.

The list $\tilde{G}$ is a list of elements $\in \langle G \rangle$ with $\mathrm{LM}(\tilde{G}) \supseteq \mathrm{LM}(G)$ which implies $\tilde{G}$ is a Gröbner basis.

We cannot amplify our confidence since we only have a limited number of samples.

Hardness II

Lemma (IR <=> IM)

When the search space of remainders is $\mathrm{poly}(\lambda)$*, the* IM *and* IR *problems are equivalent, since the attacker can exhaustively search for the remainder using the* IM *oracle.*

Thus, we have decision (IM) to search (GB) reduction for the right choice of parameters . . . it remains to be established that GB is hard.

Hardness III

Assuming that $f_1, \ldots, f_m$ is a random system, the complexity of currently best known algorithms (i.e. with F_5) to solve the GB problem is given by

$$\mathcal{O}\left(\binom{n+D}{D}^{\omega}\right) = \mathcal{O}((n^D)^{\omega})$$

where $2 \leq \omega < 3$ is the linear algebra constant, and D is given by the index of the first non-positive coefficient of:

$$\sum_{k \geq 0} c_k z^k = \frac{(1-z^b)^m}{(1-z)^n}.$$

Thus Gröbner bases are exponential in n, if D is polynomial in n.

Hardness IV

Corollary

Let $c \geq 0$. Then for $m(\lambda) = c \cdot n(\lambda)$ or $m(\lambda) = c \cdot n(\lambda)^b$ polynomials of degree b in some ideal $\mathcal{I}$, the Gröbner basis of $\mathcal{I}$ can be computed in exponential or polynomial time in $n(\lambda)$ respectively.

Definition (GB/IR/IM Assumption)

Let $\mathcal{P}$ be such that $n(\lambda) = \Omega(\lambda)$. Assume $b - d > 0$, $b > 1$, and that $m(\lambda) = c \cdot n(\lambda)$ for a constant $c \geq 1$. Then the advantage of any ppt algorithm in solving the GB/IR/IM problem is negligible as function of λ.

Outline

Symmetric PollyCracker I

Algo. $\mathsf{Gen}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,b}(1^\lambda)$

begin
- $P \leftarrow_\$ \mathbf{P}_\lambda$;
- $G \leftarrow_\$ \mathsf{GBGen}(1^\lambda, P, d)$;
- $\mathsf{SK} \leftarrow (G, P, b)$;
- $\mathsf{PK} \leftarrow (P, b)$;
- **return** $(\mathsf{SK}, \mathsf{PK})$;

end

Algo. $\mathsf{Enc}(m, \mathsf{SK})$:

begin
- $f \leftarrow_\$ P_{\leq b}$;
- $f \leftarrow f - (f \mod G)$;
- $c \leftarrow m + f$;
- **return** c;

end

Algo. $\mathsf{Dec}(c, \mathsf{SK})$:

begin
- $m \leftarrow c \mod G$;
- **return** m;

end

Algo. $\mathsf{Eval}(c_0, \ldots, c_{t-1}, C, \mathsf{PK})$:

begin
- apply the Add and Mult gates of C over P;
- **return** the result;

end

Figure: The noise-free symmetric Polly Cracker scheme $\mathcal{SPC}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,b}$.

Security

The $m(\cdot)$-time IND-CPA security is defined by requiring that the advantage of any ppt $\mathcal{A}$

$$\mathbf{Adv}^{\text{ind-bcpa}}_{m(\cdot),\mathcal{SKE},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\text{IND-BCPA}^{\mathcal{A}}_{m(\cdot),\mathcal{SKE}}(\lambda) \Rightarrow \mathsf{T}\right] - 1$$

is negligible in λ. The difference with the usual CPA security is that the adversary can query the encryption oracle at most $m(\lambda)$ times.

Theorem

For any $\mathcal{A}$ against the m-time IND-BCPA *security of $\mathcal{SPC}$ there exists a $\mathcal{B}$ against the* IM *problem such that*

$$\mathbf{Adv}^{\text{ind-bcpa}}_{m,\mathcal{SPC},\mathcal{A}}(\lambda) = 2 \cdot \mathbf{Adv}^{\text{im}}_{\mathcal{P},\text{GBGen}(\cdot),d,b,m,\mathcal{B}}(\lambda).$$

Conversely, fory any $\mathcal{A}$ against the IM *problem there exists a $\mathcal{B}$ against the m-time* IND-BCPA *security of $\mathcal{SPC}$ such that*

$$\mathbf{Adv}^{\text{im}}_{\mathcal{P},\text{GBGen}(\cdot),d,b,m,\mathcal{A}}(\lambda) = \mathbf{Adv}^{\text{ind-bcpa}}_{m,\mathcal{SPC},\mathcal{B}}(\lambda).$$

Outline

Conversions in the Literature

- There are a few techniques in the literature, which convert an IND-CPA symmetric additive homomorphic scheme to an IND-CPA public-key additive homomorphic scheme.
- One such conversion is to publish N encryptions of zero $f_1, \ldots, f_N$ and to encrypt as

$$c = \sum_{s \in S} f_s + m$$

where S is a small subset of $\{1, \ldots, N\}$.

While PollyCracker is additive homomorphic and secure up to some bound, none of the proposed conversions give a secure scheme.

Impossibility Result I

Theorem (Dickenstein, Fitchas, Giusti, and Sessa)

Let $\mathcal{I} = \langle f_1, \ldots, f_m \rangle$ *be an ideal in* $P = \mathbb{F}[x_1, \ldots, x_n]$, h *be such that* $\deg(h) \leq D$, *and*

$$h - (h \mod \mathcal{I}) = \sum_{i=1}^{m} h_i f_i,$$

where $h_i \in P$ *and* $\deg(h_i f_i) \leq D$.

Let G *be the output of some Gröbner basis computation algorithm up to degree* D *(i.e., all computations with degree greater than* D *are ignored and dropped).*

Then $h \mod \mathcal{I}$ *can be computed by polynomial reduction of* h *via* G.

Impossibility Result II

Theorem

Let $\mathcal{I} = \langle f_1, \ldots, f_m \rangle$ be an ideal in $P = \mathbb{F}[x_1, \ldots, x_n]$. If there is a ppt algorithm $\mathcal{A}$ which samples elements from $\mathcal{I}$ uniformly given only $(f_1, \ldots, f_m) \in \mathcal{I}$, then there exists a ppt algorithm $\mathcal{B}$ which computes a Gröbner basis for $\mathcal{I}$.

Proof.

We can compute the normal forms of any f produced by $\mathcal{A}$ in polynomial time since we know $f_1, \ldots, f_m$.

If f is arbitrary in the ideal $\mathcal{I}$, we know that normals forms are equivalent to Gröbner basis computations.

Thus, we have a polynomial time algorithm for computing Gröbner bases.

Outline

Discrete Gaussian

Definition (Discrete Gaussian Distribution)

Let $\alpha > 0$ be a real number and $q \in \mathbb{N}$. The discrete Gaussian distribution $\chi_{\alpha,q}$, is a Gaussian distribution rounded to the nearest integer and reduced modulo q with mean zero and standard deviation αq.

Sampling Noisy Elements in $\mathcal{I}$

1 **begin**
2 $\quad f \leftarrow_{\$} P_{\leq b}$;
3 $\quad f \leftarrow f - (f \mod G)$;
4 $\quad e \leftarrow_{\$} \chi$;
5 $\quad$ **return** $f + e$;
6 **end**

Algorithm 3: **Sample**()

Noisy Variants of Classical Computational Problems I

GBN Given access to m noisy samples from $\mathcal{I}$ recover G.

IRN Given access to m noisy samples from $\mathcal{I}$ and a challenge $f \in P$, recover $f \mod G$.

IMN Given access to m noisy samples from $\mathcal{I}$ and a challenge $f \in P$, recover if $f \mod G \approx 0$.

Our Ideal Membership with Noise (IMN) is essentially Gentry's Ideal Coset problem for noisy polynomials.

Lemma (IRN Hard $\Leftrightarrow$ GBN Hard)

For any ppt adversary $\mathcal{A}$ against the IRN *problem, there exists a ppt adversary $\mathcal{B}$ against the* GBN *problem such that*

$$\mathbf{Adv}^{\mathsf{irn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathcal{B}}(\lambda).$$

. . . and vice versa.

Proof Sketch.

The proof proceeds exactly as in the GB $\Leftrightarrow$ IR case (query IR for all leading monomials) except that we can amplify our confidence in the output.

Lemma (IMN Hard $\Leftrightarrow$ IRN Hard)

For any ppt adversary $\mathcal{A}$ against the IMN *problem, there exists a ppt adversary $\mathcal{B}$ against the* IRN *problem such that*

$$\mathbf{Adv}^{\mathsf{imn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathcal{A}}(\lambda) \leq \mathbf{Adv}^{\mathsf{irn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,b,\chi,\mathcal{B}}(\lambda),$$

if $q(\lambda)^{\dim_{\mathbb{F}_q}(\mathcal{P}(\lambda)/\mathsf{GBGen}(\cdot))}$ *is polynomial in* λ.

. . . and vice versa.

Proof Sketch.

The proof proceeds exactly as in the IR $\Leftrightarrow$ IM case (exhaustive search of $P/\mathcal{I}$) except that

- we can amplify our confidence in the output and
- we can also "discover" $P/\mathcal{I}$ during the execution of the algorithm.

Security I

Lemma (LWE Hard $\Rightarrow$ GBN Hard for $d = 1, b = 1$)

Let q be a prime number. Then for any ppt adversary $\mathcal{A}$ against the GBN *problem with $b = d = 1$, there exists a ppt adversary $\mathcal{B}$ against the* LWE *problem such that*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),1,1,\chi,\mathcal{A}}(\lambda) = \mathbf{Adv}^{\mathsf{lwe}}_{n,q,\chi,\mathcal{B}}(\lambda).$$

Proof.

Whenever $\mathcal{A}$ calls its **Sample** oracle, $\mathcal{B}$ queries its own **Sample** oracle to obtain (a, b) where $a = (a_0, \ldots, a_{n-1})$. It returns $\sum a_i x_i - b$ to $\mathcal{A}$. When $\mathcal{A}$ calls its **Finalize** on G, since $d = 1$, we may assume that G is of the form $[x_0 - s_0, \ldots, x_{n-1} - s_{n-1}]$ with $s_i \in \mathbb{F}_q$. Algorithm $\mathcal{B}$ terminates by calling its **Finalize** oracle on $s = (s_0, \ldots, s_{n-1})$.

Security II

Lemma (GBN Hard for $2b \Rightarrow$ GBN Hard for b)

For any ppt adversary $\mathcal{A}$ against the GBN *problem at degree b with noise $\chi_{\alpha,q}$, there exists a ppt adversary $\mathcal{B}$ against the* GBN *problem at degree $2b$ with noise $\chi_{\sqrt{N}\alpha^2 q,q}$ such that*

$$\mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,b,\chi_{\alpha,q},\mathcal{A}}(\lambda) = \mathbf{Adv}^{\mathsf{gbn}}_{\mathcal{P},\mathsf{GBGen}(\cdot),d,2b,\chi_{\sqrt{N}\alpha^2 q,q},\mathcal{B}}(\lambda)$$

for $N = \binom{n+b}{b}$.

Proof.

Multiply samples f_i, f_j to get $f_{i,j} = f_i \cdot f_j$. To ensure sufficient randomness, sum up N such products.

Security III

Approximate GCD:

- The GBN problem for $n = 1$ is the approx. GCD problem over $\mathbb{F}_q[x]$.
- This problem has not yet received much attention, and hence it is unclear under which parameters it is hard.
- However, the notion of a Gröbner basis can been extended to $\mathbb{Z}[x_0, \ldots, x_{n-1}]$.
- This implies a version of the GBN problem over $\mathbb{Z}$.
- This can be seen as a direct generalisation of the approximate GCD problem in $\mathbb{Z}$.

Security IV

GBN over $\mathbb{F}_2$:

- For $d = 1$ and $q = 2$ we can reduce Max-3SAT instances to GBN instances by translating each clause individually to a Boolean polynomial.
- The Gröbner basis returned by an arbitrary algorithm $\mathcal{A}$ solving GBN using a **bounded number** of samples will provide a solution to the Max-3SAT problem.
- Vice versa, we may convert a GBN problem for $d = 1$ to a Max-SAT problem (more precisely Partial Max-Sat) by running an ANF to CNF conversion algorithm.

Security V

Best known attack (for $d = 1$):

- We reduce GBN to a larger LWE instance.
- Denote by $N = \binom{n+b}{b}$ the number of monomials up to degree b.
- Let $\mathcal{M} : P \to \mathbb{F}_q^N$ be a function which maps polynomials in P to vectors in $\mathbb{F}_q^N$ by assigning the i-th component of the image vector the coefficient of the i-th monomial $\in M_{\leq b}$.
- Reply to each **Sample** query by the LWE oracle by calling the GBN **Sample** oracle to retrieve f, compute $v = \mathcal{M}(f)$ and return (a, b) with $a = (v_{N-1}, \ldots, v_1)$ and $b = -v_0$.
- When the LWE oracle queries its **Finalize** with s query the GBN **Finalize** with $[x_0 - s_0, \ldots, x_{n-1} - s_{n-1}]$.

Outline

Polly Cracker with Noise

- GBN/IRN/IMN allow to construct a noisy version of our symmetric Polly Cracker scheme: $\mathcal{SPCN}$.
- $\mathcal{SPCN}$ is IND-CPA under the GBN assumption.
- Using any symmetric-to-asymmetric conversion from literature this leads to a public-key Polly Cracker scheme.
- This scheme is somewhat homomorphic and can support a fixed but arbitrary number of multiplications.
- This also implies that Regev's public-key scheme based on LWE is multiplicative homomorphic under some choice of parameters.

$$\begin{aligned} c_0 \cdot c_1 &= (m_0 + 2e_1 + \sum h_{0i} g_i) \cdot (m_1 + 2e_1 + \sum h_{1i} g_i) \\ &= m_0 m_1 + 2e_0 e_1 + 2\tilde{e} + \sum \tilde{h}_i g_i \end{aligned}$$

Trading Noise for Degree I

- The product of two polynomials of degree b is a polynomial of degree $2b$, and hence the size of the ciphertext squares if multiplied.
- We can reduce polynomials of degree $2b$ to polynomials of degree b by performing a proxy re-encryption.
- Let $P = \mathbb{F}[x_1, \ldots, x_n]$ and suppose $G_A = \{g_1, \ldots, g_n\}$ and $G_B = \{h_1, \ldots, h_n\}$ are Gröbner bases for ideals $\mathcal{I}_A$ and $\mathcal{I}_B$.
- Suppose $P/\mathcal{I}_A = P/\mathcal{I}_B$.

Trading Noise for Degree II

- To re-encrypt a ciphertext intended for G_A under key G_B we generate the re-encryption key $G_{A \to B}$.
- For this the central idea is the equivalence between different representations of elements in $P/\mathcal{I}$.
- That is, we make use of different representations of elements in $P/\mathcal{I}$.
- For example, if $x + 1$ is an element of a Gröbner basis G_A both $f = x$ and $r = -1$ represent the same element in $P/\mathcal{I}_A$ since $f \mod G_A = r$, i.e., $x \mod G_A = -1$.
- Hence, if we are interested in $P/\mathcal{I}_A$ we can use f and r interchangeably.

Trading Noise for Degree III

- That is, for some $f = \sum c_i m_i$ with monomials m_i and coefficients $c_i \in \mathbb{F}_q$, we can compute the first decryption step, i.e.,

$$m + 2e = f \mod \mathcal{I}_A, \text{ as } \sum (c_i m_i \mod \mathcal{I}_A).$$

- Furthermore, since $P/\mathcal{I}_A = P/\mathcal{I}_B$, we may encrypt the encoded message $m + 2e$ for G_B by computing

$$\begin{aligned} f' &= (f \mod \mathcal{I}_A) + \tilde{f} \\ &= \sum (c_i m_i \mod \mathcal{I}_A) + \tilde{f} \\ &= m + 2e + \tilde{f} \text{ for } \tilde{f} \in \mathcal{I}_B. \end{aligned}$$

- Hence, we get that $f' \mod \mathcal{I}_B = f \mod \mathcal{I}_A$.

Trading Noise for Degree IV

begin
  $G_{A\to B} \leftarrow \varnothing$;
  for $m \in M_{\leq b'}$ **do**
    $m' \leftarrow m \mod G_A$;
    for $0 \leq j < \lceil \log_2(q/2) \rceil$ **do**
      $s \leftarrow_{\$}$ a sparse subset of $[0, \ldots, m-1]$;
      $f \leftarrow \sum_s f_s$;
      $G_{A\to B}[2^j \cdot m] \leftarrow f + 2^j \cdot m'$;
      $G_{A\to B}[-2^j \cdot m] \leftarrow f - 2^j \cdot m'$;
  return $G_{A\to B}$;
end

Algorithm 4: Generating the re-encryption key

Trading Noise for Degree V

- Now, using the key $G_{A \to B}$ we may re-encrypt a ciphertext f under G_A to a ciphertext f' under G_B.
- All elements in $G_{A \to B}$ are of degree at most b. Hence, the degree of the output is at most b.

Trading Noise for Degree VI

1 **begin**
2 $f' \leftarrow 0$;
3 **for** $m \in f$ **do**
4 $c \leftarrow$ the coefficient in f of m represented as an integer in $(-\lfloor \frac{q}{2} \rfloor, \lfloor \frac{q}{2} \rfloor]$;
5 $m' \leftarrow 0$;
6 **for** $0 \leq j < \lceil \log_2(q/2) \rceil$ **do**
7 **if the** j**-th bit of** $|c|$ **is set then**
8 $m' \leftarrow m' + G_{A \rightarrow B}[2^j \cdot m]$;
9 **if** $c < 0$ **then**
10 $m' \leftarrow -1 \cdot m'$;
11 $f' \leftarrow f' + m'$;
12 **return** f';
13 **end**

Algorithm 5: Re-encryption

Trading Noise for Degree VII

- Consider re-encryption under the same key, i.e., $G_A = G_B$.
- If $b' = b$, the key $G_{A \to A}$ can be constructed publicly given access to encryptions of zero by requesting a fresh encryption of zero f and storing $G_{A \to A}[2^j \cdot m] = 2^j \cdot m + f$.
- Since $(f \mod \mathcal{I}) = 2e$ for some small error term e it holds that $f + 2^j \cdot m \mod \mathcal{I} = (2^j \cdot m \mod \mathcal{I}) + 2e$.
- Hence, $G_{A \to A}$ is a correct re-encryption key which can be generated given only access to encryptions zero, i.e., no additional information is leaked.
- However, this argument does not go through for $b' > b$.
- While it is easy to construct elements f which satisfy $f \mod \mathcal{I} \approx 2^j \cdot m \mod \mathcal{I}$ for m a monomial of degree $> b$ for anyone with access to encryptions of zero, it is not easy to produce such elements with degree $\leq b$ and small noise.

Recent Related Work

- **Efficient Fully Homomorphic Encryption from (Standard) LWE** by Brakerski and Vaikuntanathan available on ePrint provides an independent construction closely related to ours.
- Their first part – **relineraisation** – is essentially identitical to ours.
- However, in contrast to our work, their perspective is explicitly non-algebraic: "In contrast, all previous schemes relied on complexity assumptions related to ideals in various rings."
- Their **dimension reduction technique** goes beyond our construction and provides a powerful technique for the construction of homomorphic encryption schemes.
- Note that it also applies to our construction.

Thank you for your attention

Questions?