When Homomorphism Becomes a Liability

Zvika Brakerski[*]

Abstract

We show that an encryption scheme cannot have a simple decryption circuit and be homomorphic at the same time. Specifically, if a scheme can homomorphically evaluate the majority function, then its decryption circuit cannot be a linear function of the secret key (or even a succinct polynomial), even if decryption error is allowed.

An immediate corollary is that known schemes that are based on the hardness of decoding in the presence of noise with *low hamming weight* cannot be fully homomorphic. This applies to known schemes such as LPN-based symmetric or public key encryption.

An additional corollary is that the recent candidate fully homomorphic encryption, suggested by Bogdanov and Lee (ePrint '11, henceforth BL), is insecure. In fact, we show two attacks on the BL scheme: One by applying the aforementioned general statement, and another by directly attacking one of the components of the scheme.

[*] Stanford University, `zvika@stanford.edu`. Supported by a Simons Postdoctoral Fellowship and by DARPA.

1 Introduction

An encryption scheme is called homomorphic if there is an efficient transformation that given $\mathsf{Enc}(m)$ for some message m, and a function f, produces $\mathsf{Enc}(f(m))$ using only public information. A scheme that is homomorphic w.r.t all efficient f is called fully homomorphic (FHE). Homomorphic encryption is a useful tool in both theory and practice and is extensively researched in recent years (see [Vai11] for survey), and a few candidates for full homomorphism are known.

Most of these candidates [Gen09, Gen10, SV10, BV11a, BV11b, GH11, BGV12, GHS12, Bra12] are based (either explicitly or implicitly) on *lattice assumptions* (the hardness of approximating short vectors in certain lattices). In particular, the learning with errors (LWE) assumption proved to be very useful in the design of such schemes. The one notable exception is [vDGHV10], but even that could be thought of as working over an appropriately defined lattice over the integers.

An important open problem is, therefore, to diversify and base fully homomorphic encryption on different assumptions (so as to not put all the eggs in one basket). One appealing direction is to try to use the learning parity with noise (LPN) problem, which is very similar in syntax to LWE: Making a vast generalization, LWE can be interpreted as a decoding problem for a linear code, where the noise comes from a family of *low norm* vectors. Namely, each coordinate in the code suffers from noise, but this noise is relatively small (this requires that the code is defined over a large alphabet). The LPN assumption works over the binary alphabet and requires that the noise has low hamming weight, namely that only a small number of coordinates are noisy, but in these coordinates the noise amplitude can be large. While similar in syntax, a direct connection between these two types of assumptions is not known.

While an LPN based construction is not known, recently Bogdanov and Lee [BL11] presented a candidate, denoted by BL throughout this manuscript, that is based on a different low-hamming-weight decoding problem: They consider a carefully crafted code over a large alphabet and assume that decoding in the presence of low-hamming-weight noise is hard.

In this work we show that not only that BL's construction is insecure, but rather the entire approach of constructing code based homomorphic encryption analogously to the LWE construction cannot work. We stress that we don't show that FHE cannot be based on LPN (or other code based assumptions), but rather that the decryption algorithm of such scheme cannot take the naïve form. (In particular it means that known schemes such as [Ale03, GRS08, ACPS09] cannot be fully homomorphic as is.)

1.1 Our Results

Our main result shows that encryption schemes with simple decryption circuits cannot be homomorphic, even if large decryption error is allowed. In particular, they cannot evaluate the majority function. The following theorem is proven in Section 3.

Theorem A. *An encryption scheme whose decryption circuit can be expressed as a constant degree polynomial in the elements of the secret key and ciphertext, and whose decryption error is* $< 1/2 - 1/\mathrm{poly}(n)$*, cannot homomorphically evaluate the majority function.*

As a byproduct we can deduce that known LPN based schemes such as [Ale03, GRS08, ACPS09] cannot be fully homomorphic unless their decryption circuit is changed. This may not seem obvious at first: the decryption circuit of the aforementioned schemes is not a constant degree polynomial, and their decryption error is negligible, so they seem to be out of the scope of our theorem. However, looking more closely, the decryption circuits consist of an inner product computation with the secret key, followed by additional post-processing. One can verify that if the post processing is not performed, then correct decryption is still achieved with probability $> 1/2 + 1/\text{poly}$. Thus we can apply our theorem and rule out homomorphism.

Furthermore, the same logic rules out the homomorphism of the BL candidate FHE. This contradiction implies that the BL scheme is insecure.

Theorem B. *There is a successful polynomial time CPA attack on the BL scheme.*

We further present a different attack on the BL scheme, targeting one of its building block. This allows us to not only distinguish between two messages like the successful CPA attack above, but rather decrypt any ciphertext with probability $1 - o(1)$.

Theorem C. *There is a polynomial time algorithm that decrypts the BL scheme.*

The BL scheme and the two breaking algorithms are presented in Section 4.

1.2 Our Techniques

Consider a simplified variant of Theorem A, where we require that the decryption function is an inner product between the ciphertext and the secret key (both represented as n dimensional vectors over some field), and we assume that the decryption error is at most, say, $1/(100n)$. Then the proof seems trivial: Sample $O(n)$ encryptions of 0. With good probability, all ciphertext decrypt correctly (using the union bound) and therefore we get $O(n)$ linear equations on the secret key. Using Gaussian elimination, the key can be reconstructed and the scheme is broken. (In fact, a somewhat more complicated argument is needed in case the $O(n)$ equations that we get are not linearly independent.) Note that we did not use the homomorphism of the scheme at all, indeed this simplified version is universally true even without assuming homomorphism.

The next step is to still consider linear decryption, but allow decryption error $1/2 - \epsilon$. Now we will need to use the homomorphism property. The idea is to use the homomorphism in order to reduce the decryption error and get back to the previous case. Consider a scheme that encrypts each message many times (say k), and then applies homomorphic majority on the ciphertexts. The security of this scheme directly reduces to that of the original scheme, and it has the same decryption function, however now the decryption error drops exponentially with k. This is because in order to get an error in the new scheme, at least $k/2$ out of the k encryptions need to have errors. Since the expected number is $(1/2 - \epsilon)k$, the Chernoff bound implies the result. Choosing k appropriately will bring us back to the previous case.

Finally, to generalize the result beyond linear functions, we notice that we can represent a succinct polynomial (i.e. one that only has polynomially many terms) as an inner product between two generalized vectors. For example, the polynomial $x_1x_2y_1^2 + x_1y_2 + 3x_2y_1y_2$, can be expressed as $\langle (x_1x_2, x_1, 3x_2), (y_1^2, y_2, y_1y_2) \rangle$. This simple observation concludes the proof.

While Theorem B follows immediately, as we explain above, for Theorem C we need to work a little harder. We notice that BL use homomorphic majority evaluation in one of the lower abstraction levels of their scheme. This allows us to break this abstraction level using similar reasoning to Theorem A. A complete break of BL follows.

1.3 Other Related Work

An independent work by Gauthier, Otmani and Tillich [GOT12] shows an interesting direct attack on BL's hardness assumption (we refer to it as the "GOT attack"). Their attack is very different from ours and takes advantage of the resemblance of BL's codes and Reed-Solomon codes as we explain below.

BL's construction relies on a special type of error correcting code. Essentially, they start with a Reed-Solomon code, and replace a small fraction of the columns of the generating matrix with a special structure. The homomorphic properties are only due to this small fraction of "significant" columns, and the secret key is chosen so as to nullify the effect of the other columns.

The GOT attack uses the fact that under some transformation (component-wise multiplication), the dimension of Reed-Solomon codes can grow by at most a factor of two. However, if a code contains "significant" columns, then the dimension can grow further. This allows to measure the number of significant columns in a given code. One can thus identify the significant columns by trying to remove one column at a time from the code and checking if the dimension drops. If yes then that column is significant. Once all significant columns have been identified, the secret key can be retrieved in a straightforward manner.

However, it is fairly easy to immunize BL's scheme against the GOT attack. As we explained above, the neutral columns do not change the properties of the encryption scheme, so they may as well be replaced by random columns. Since the dimension of random codes grows very rapidly under the GOT transformation, their attack will not work in such case.

Our attack, on the other hand, relies on the functional properties that BL use to make their scheme homomorphic. Thus a change in the scheme that preserves the homomorphic properties cannot help to overcome our attack. In light of our attack, it is interesting to investigate whether the GOT attack can be extended to the more general case.

2 Preliminaries

We denote scalars using plain lowercase (x), vectors using bold lowercase ($\mathbf{x}$ for column vector, $\mathbf{x}^T$ for row vector), and matrices using bold uppercase ($\mathbf{X}$). We let $\mathbf{1}$ denote the all-one vector (the dimension will be clear from the context). We let $\mathbb{F}_q$ denote a finite field of cardinality $q \in \mathbb{N}$, with efficient operations (we usually don't care about any other property of the field).

2.1 Properties of Encryption Schemes

A public key encryption scheme is a tuple of algorithms $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, such that: $\mathsf{Gen}(1^n)$ is the key generation algorithm that produces a pair of public and secret keys (pk, sk); $\mathsf{Enc}_{pk}(m)$ is a randomized encryption function that takes a message m and produces a ciphertext. In the context of this work, messages will only come from some predefined field $\mathbb{F}$; $\mathsf{Dec}_{sk}(c)$ is the decryption function that decrypts a ciphertext c and produces the message. Optimally, $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(\cdot))$ is the identity function, but in some schemes there are decryption errors.

The probability of decryption error is taken over the randomness used to generate the keys for the scheme and over the randomness used in the encryption function (we assume the decryption is deterministic). In this paper, our focus is on the error induced by the randomness of the encryption. We therefore allow a small fraction of the keys (one percent, for the sake of convenience) to have arbitrarily large decryption error, and define the decryption error ϵ to be the maximal error over the 99% best keys. This definition is not equivalent to other definitions of decryption error, and in fact our results can be generalized to other definitions (e.g. an additional parameter δ instead of 1%). However we feel that the improvement in the results does not justify the added complication. Our definition, therefore, is as follows.

Definition 2.1. *An encryption scheme is said to have decryption error $< \epsilon$ if with probability at least* 0.99 *over the key generation it holds that*

$$\max_m \{\Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) \neq m]\} < \epsilon \ ,$$

where the probability is taken over the random coins of the encryption function.

We use the standard definition of security against chosen plaintext attacks (CPA): The attacker receives a public key and chooses two values m_0, m_1. The attacker then receives a ciphertext $c = \mathsf{Enc}_{pk}(m_b)$, where $b \in \{0,1\}$ is a random bit that is unknown to the attacker. The attacker needs to decide on a guess $b' \in \{0,1\}$ as to the value of b. We say that the scheme is broken if there is a polynomial time attacker for which $\Pr[b' = b] \geq 1/2 + 1/\mathrm{poly}(n)$ (where n is the security parameter). Recall that this notion is equivalent to the notion of semantic security [GM82].

In addition, we will say that a scheme is *completely broken* if there exists an adversary that upon receiving the public key and $\mathsf{Enc}_{pk}(m)$ for *arbitrary value of* m, returns m with probability $1 - o(1)$.

In this work, we consider schemes with linear decryption circuits, as defined below.

Definition 2.2. *An encryption scheme is n-linearly decryptable if its secret key is of the form $sk = \mathbf{s} \in \mathbb{F}^n$, for some field $\mathbb{F}$, and its decryption function is*

$$\mathsf{Dec}_{sk}(\mathbf{c}) = \langle \mathbf{s}, \mathbf{c} \rangle \ .$$

We discuss the homomorphic properties of encryption schemes. However the only notion of homomorphism that we use is w.r.t the majority function. We define the notion of k-majority-homomorphism below.

Definition 2.3. *A public-key encryption scheme is k-*majority-homomorphic *(where k is a function of the security parameter) if there exists a function* $\mathsf{MajEval}$ *such that with probability* 0.99 *over the key generation, the following holds: For any sequence of ciphertexts output by $\mathsf{Enc}_{pk}(\cdot)$: $c_1, \ldots, c_k$, it holds that*

$$\mathsf{Dec}_{sk}(\mathsf{MajEval}_{pk}(c_1, \ldots, c_k)) = \mathsf{Majority}(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_k)) \ .$$

Again we allow some "slackness" by allowing some of the keys to not abide the homomorphism.

We note that Definition 2.3 above is a fairly strong notion of homomorphism in two aspects: First, it requires that homomorphism holds even for ciphertexts with decryption error. Second, we do not allow $\mathsf{MajEval}$ to introduce error for "good" key pairs. Indeed, known homomorphic encryption schemes have these properties, but it is interesting to try to bypass our negative results by finding schemes that do not have them.

2.2 Spanning Distributions over Low Dimensional Spaces

We will use a lemma that shows that any distribution over a low dimensional space is easy to span in the following sense: Given sufficiently many samples from the distribution (a little more than the dimension of the support), we are guaranteed that any new vector falls in the span of previous samples. This lemma will allow us to derive a very general result that only depends on the decryption function, irrespective to the way encryption is performed.

We speculate that this lemma is already known, since it is fairly general and very robust to the definition of dimension (e.g. it also applies to non-linear spaces).

Lemma 2.1. *Let $\mathcal{S}$ be a distribution over a linear space S of dimension s. For all k, define*

$$\delta_k \triangleq \Pr_{\mathbf{v}_1,\ldots,\mathbf{v}_k \xleftarrow{\$} \mathcal{S}} \left[\mathbf{v}_k \notin \mathrm{Span}\left\{\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}\right\}\right] \ .$$

Then $\delta_k \le s/k$.

Proof. Notice that by symmetry $\delta_i \ge \delta_{i+1}$ for all i. Let D_i denote the (random variable) dimension of $\mathrm{Span}\left\{\mathbf{v}_1, \ldots, \mathbf{v}_i\right\}$. Note that always $D_i \le s$.

Let E_i denote the event $\mathbf{v}_i \notin \mathrm{Span}\left\{\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}\right\}$, note that $\delta_i = \Pr[E_i]$. By definition,

$$D_k = \sum_{i=1}^{k} \mathbb{1}_{E_i} \ .$$

Therefore

$$s \ge \mathbb{E}[D_k] = \mathbb{E}\left[\sum_{i=1}^{k} \mathbb{1}_{E_i}\right] = \sum_{i=1}^{k} \Pr[E_i] = \sum_{i=1}^{k} \delta_i \ge k \cdot \delta_k \ ,$$

and the lemma follows. □

3 Homomorphism is a Liability When Decryption is Simple

This section shows our main result. We show that schemes with simple decryption circuits are very limited in terms of their correctness probability and their homomorphic properties. We start by showing that a scheme with linear decryption cannot have decryption error smaller than $\Omega(1/n)$ and be secure (regardless of homomorphism). We then show that if the scheme can homomorphically evaluate the majority function, then the above amplifies dramatically and security can not be guaranteed for *any* reasonable decryption error: We rule out $1/2 - \epsilon$ error for any noticeable ϵ. The latter is then extended to non-linear decryption functions such as low-degree polynomials.

For the sake of simplicity, we only focus on the public key case. However, we note that our proofs easily extend also to symmetric encryption, since our attacks only use the public key in order to generate ciphertexts for known messages.

3.1 Linear Decryption without Homomorphism

We start by showing that an n-linearly decryptable scheme has to have decryption error $\Omega(1/n)$, otherwise it is insecure. The basic idea is very simple: Consider a ciphertext that correctly decrypts to 0. Such ciphertext corresponds to a linear constraint on the secret key, since $\langle \mathbf{c}, \mathbf{s} \rangle = 0$. Given

n such constraints, we hope to be able to find $\mathbf{s}$ using Gaussian elimination. Therefore, if the decryption error is under $\Omega(1/n)$, then we can get such n equations by just encrypting 0 ourselves n times. This argument is a little too simplistic since the encryptions of 0 can be linearly dependent. For example, think of a scheme that encrypts 0 using the zero vector: this scheme is obviously insecure, but the above method cannot retrieve the secret key. We therefore use Lemma 2.1 to argue that no matter what distribution the encryption algorithm produces, it can still be learned using sufficiently many samples in a way that breaks the security of the scheme. A formal statement and proof follow.

Theorem 3.1. *An n-linearly decryptable scheme with decryption error $< 1/(36n)$ is insecure.*

Proof. We start by observing that given a key pair, the set of ciphertexts that decrypt to 0 is a homogenous linear space (the orthogonal space to $\mathbf{s}$). Denote this space by $Z \subseteq \mathbb{F}^n$. The dimension of Z is at most n (in fact, at most $n-1$). The ciphertexts that decrypt to m for some $m \in \mathbb{F}$ are a coset of Z, that we denote by Z_m. Our proof works by finding an approximate basis for Z: a set of vectors that with high probability span Z. For a given key pair, we let $\mathcal{Z}$ denote the probability distribution $\mathbf{c} = \mathsf{Enc}_{pk}(0) | \mathsf{Dec}_{sk}(\mathbf{c}) = 0$. Namely the distribution induced by encrypting 0, conditioned on the ciphertext being correctly decryptable.

We will show a polynomial time CPA attacker that succeeds with probability noticeably higher than 1/2 to distinguish between encryptions of 0 and encryptions of nonzero as follows: Given a public key pk, the attacker lets $m_0 = 0, m_1 = 1$ (or any other nonzero value). The attacker generates $6n$ encryptions of 0, denoted $\mathbf{v}_1, \ldots, \mathbf{v}_{6n}$, and lets V denote their span. Then, given a challenge ciphertext $\mathbf{c} = \mathsf{Enc}_{pk}(m_b)$, the attacker outputs 0 if and only if $\mathbf{c} \in V$.

We will first analyze the attacker's success probability on "good" key pairs, ones for which the decryption error is indeed bounded:

- If $b = 0$ then $\mathbf{c} = \mathsf{Enc}_{pk}(0)$. With probability $(6n+1)/(36n)$ it holds that all $\mathbf{v}_1, \ldots, \mathbf{v}_{6n}, \mathbf{c}$ indeed decrypt to 0, in which case we can apply Lemma 2.1 to conclude that $\mathbf{c} \in V$ with probability at least $(n-1)/(6n)$. It follows that in this case $b' = b$ with probability at least $1 - (6n+1)/(36n) - (n-1)/(6n) \geq 2/3$.
- If $b = 1$ then with probability $(6n+1)/(36n)$ it holds that all $\mathbf{v}_1, \ldots, \mathbf{v}_{6n}$ indeed decrypt to 0 and $\mathbf{c}$ indeed decrypts to 1. In such case it must be that $\mathbf{c} \notin V$. It follows that $b' = b$ with probability at least $1 - (6n+1)/(36n) \geq 2/3$.

It follows that for "good" key pairs, the attacker succeeds with probability at least 2/3. Since the probability of a bad key pair is at most 0.01, the total success probability of the attacker is at least $2/3 - 0.01 > 1/2$. □

We note that tighter versions of the theorem are possible, and also ones with better advantage for the attacker, however the above is sufficient for our needs.

3.2 Simple Decryption with Majority Homomorphism

Theorem 3.1 does not seem very restrictive. Specifically, it is not directly applicable to attacking any known scheme. Indeed, known schemes with linear decryption have sufficiently high decryption error. In this section, we show that if homomorphism is required as a property of the scheme, then no reasonable decryption error can be achieved. Furthermore, we extend the result beyond linear decryption.

Consider Definition 2.3 for majority homomorphism. The next theorem states that majority-homomorphic schemes cannot have linear decryption for any reasonable decryption error.

Theorem 3.2. *An n-linear decryptable scheme with $(1/2-\epsilon)$ decryption error cannot be $O(\log n/\epsilon^2)$-majority-homomorphic.*

Let us first outline the proof before formalizing it. Our goal is the same as in the proof of Theorem 3.1, to generate $O(n)$ properly decryptable encryptions of 0, which will enable to break security. However, unlike before, taking $O(n)$ independent encryptions of 0 will surely introduce decryption errors. We thus use the majority homomorphism: To generate a good encryption of 0, i.e. one that is decryptable with high probability, we will generate $O(\log n/\epsilon^2)$ random encryptions of 0, and apply majority homomorphically. Chernoff bound guarantees that with high probability, more than half of the ciphertexts are properly decryptable, and therefore the output of the majority evaluation is with high probability a decryptable encryption of 0. At this point, we can apply the same argument as in the proof of Theorem 3.1. The formal proof follows.

Proof. Consider an encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as in the theorem statement. We will construct a new scheme $(\mathsf{Gen}' = \mathsf{Gen}, \mathsf{Enc}', \mathsf{Dec}' = \mathsf{Dec})$ (with the same key generation and decryption algorithms) whose security relates to that of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Then we will use Theorem 3.1 to render the latter scheme insecure.

The new encryption algorithm $\mathsf{Enc}'_{pk}(m)$ works as follows: To encrypt a message m, invoke the original encryption $\mathsf{Enc}_{pk}(m)$ for $k = 100(\ln n + 1)/\epsilon^2$ times, thus generating k ciphertexts. Apply $\mathsf{MajEval}$ to those k ciphertexts and output the resulting ciphertext.

The security of the new scheme is related to that of the original by a straightforward hybrid argument. We will show that the new scheme has decryption error at most $1/(36n)$, but in a slightly weaker sense then Definition 2.1: We will allow 2% of the keys to be "bad" instead of just 1% as before. However one can easily verify that the proof of Theorem 3.1 works in this case as well.

Our set of good key pairs for Enc' is those for which $\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(\cdot))$ indeed have decryption error at most $1/2 - \epsilon$ and in addition $\mathsf{MajEval}$ is correct. By the union bound this happens with probability at least 0.98.

To bound the decryption error of $\mathsf{Dec}_{sk}(\mathsf{Enc}'_{pk}(\cdot))$, assume that we have a good key pair as described above. We will bound the probability that more than a $1/2 - \epsilon/2$ fraction of the k ciphertexts generated by Enc' are decrypted incorrectly. Clearly if this bad event does not happen, then by the correctness of $\mathsf{MajEval}$, the resulting ciphertext will decrypt correctly.

Recalling that the expected fraction of falsely decrypted ciphertexts is at most $1/2 - \epsilon$, the Chernoff bound implies that the aforementioned bad event happens with probability at most

$$e^{-2(\epsilon/2)^2 k} < 1/(36n) ,$$

and the theorem follows. □

From the proof it is obvious that even "approximate-majority homomorphism" is sufficient for the theorem to hold. Namely, even if $\mathsf{MajEval}$ only computes the majority function correctly if the fraction of identical inputs is more than $1/2 + \epsilon/2$.

Extension to Non-Linear Decryption. We can extend Theorem 3.2 beyond linear decryption in a straightforward manner: So long as we can interpret the decryption function $\mathsf{Dec}_{sk}(c)$ as an inner product between two generalized vectors $\langle \tilde{\mathbf{s}}, \tilde{\mathbf{c}} \rangle$, where $\tilde{\mathbf{s}}$ depends only on sk and $\tilde{\mathbf{c}}$ depends only on c, we can apply Theorem 3.2.

This can obviously be done if the decryption function is a constant degree polynomial in $\mathbf{s}, \mathbf{c}$: each monomial will be translated into a component of $\tilde{\mathbf{s}}$ and a component of $\tilde{\mathbf{c}}$, and the decryption value is the inner product between these generalized vectors. The only restriction is that the dimension of the generalized vectors must remain polynomial in the security parameter.

4 Attacks on the BL Scheme

In this section we use our tools from above to show that the BL scheme (outlines in Section 4.1 below) is broken. We present two attacks: one that follows immediately from Theorem 3.2 (Section 4.2); and one that directly attacks one of the subcomponents of the scheme (Section 4.3) and allows to decrypt any ciphertext. In fact, the latter attack follows the same principles as the former and exploits a "built-in" evaluation of majority that exists in one of the sub-components of BL.

4.1 Essentials of the BL Scheme

In this section we present the properties of the BL scheme. We only concentrate on the properties that are required for our break. In fact, for our first break, the last paragraph of this section is all that is really needed. We refer the reader to [BL11] for further details.

The BL scheme has a number of layers of abstraction, which are all instantiated based on a global parameter $0 < \alpha < 0.25$ as explained below.

The Scheme $\mathbf{K}_q(n)$. BL introduce $\mathbf{K}_q(n)$, a public-key encryption scheme with imperfect correctness. For security parameter n, the public key is a matrix $\mathbf{P} \in \mathbb{F}_q^{n \times r}$, where $r = n^{1-\alpha/8}$, and the secret key is a vector $\mathbf{y} \in \mathbb{F}_q^n$ in the kernel of $\mathbf{P}^T$ (namely, $\mathbf{y}^T \cdot \mathbf{P} = 0$). The keys are generated in a highly structured manner in order to support homomorphism, but their structure is irrelevant to us. An encryption of a message $m \in \mathbb{F}_q$ is a vector $\mathbf{c} = \mathbf{P} \cdot \mathbf{x} + m \cdot \mathbf{1} + \mathbf{e}$, where $\mathbf{x} \in \mathbb{F}_q^r$ is some vector, and where $\mathbf{e} \in \mathbb{F}_q^n$ is a low hamming weight vector. Decryption is performed by taking the inner product $\langle \mathbf{y}, \mathbf{c} \rangle$, and succeeds so long as $\langle \mathbf{y}, \mathbf{e} \rangle = 0$ (the vector $\mathbf{y}$ is chosen such that $\langle \mathbf{y}, \mathbf{1} \rangle = 1$). It is shown how the structure of the keys implies that decryption succeeds with probability at least $\left(1 - n^{-(1-\alpha/2)}\right)$. Finally, BL show that $\mathbf{K}_q(n)$ is homomorphic with respect to a single addition or multiplication.[1]

Re-Encryption. In order to enable homomorphism, BL introduce the notion of re-encryption. Consider an instantiation of $\mathbf{K}_q(n)$, with keys $(\mathbf{P}, \mathbf{y})$, and an instantiation of $\mathbf{K}_q(n')$ with keys $(\mathbf{P}', \mathbf{y}')$, for $n' = n^{1+\alpha}$. Let $\mathbf{H}_{n':n} \in \mathbb{F}_q^{n' \times n}$ be an element-wise encryption of $\mathbf{y}$ using the public key $\mathbf{P}'$.[2] Namely $\mathbf{H}_{n':n} = \mathbf{P}' \cdot \mathbf{X}' + \mathbf{1} \cdot \mathbf{y}^T + \mathbf{E}'$. Due to the size difference between the schemes, it holds

[1] Homomorphic operations (addition, multiplication) are performed element-wise on ciphertext vectors, and the structure of the key guarantees that correctness is preserved.

[2] A note on notation: In [BL11], the re-encryption parameters are denoted by I (as opposed to our $\mathbf{H}$). We feel that their notation ignores the important linear algebraic structure of the re-encryption parameters, and therefore we switched to matrix notation, which also dictated the change of letter.

that with probability $\left(1 - n^{-\Omega(1)}\right)$, all of the columns of $\mathbf{H}_{n':n}$ are simultaneously decryptable and indeed $\mathbf{y'}^T \cdot \mathbf{H}_{n':n} = \mathbf{y}^T$. In such case, for any ciphertext $\mathbf{c}$ of $\mathbf{K}_q(n)$, we get $\langle \mathbf{y}', \mathbf{H}_{n':n}\mathbf{c}\rangle = \langle \mathbf{y}, \mathbf{c}\rangle$. The matrix $\mathbf{H}_{n':n}$ therefore re-encrypts ciphertexts of $\mathbf{K}_q(n)$ as ciphertexts of $\mathbf{K}_q(n')$.

The critical idea for our second break is that a re-encrypted ciphertext always belongs to an n-dimensional linear subspace (recall that $n \ll n'$), namely to the span of $\mathbf{H}_{n':n}$.

The Scheme BASIC. Using re-encryption, BL construct a ladder of schemes of increasing lengths that allow for homomorphic evaluation. They define the scheme **BASIC** which has an additional depth parameter $d = O(1)$ (BL suggest to use $d = 8$, but our attack works for any $d > 1$). They consider instantiations of $\mathbf{K}_q(n_i)$, where $n_i = n^{(1+\alpha)^{-(d-i)}}$, for $i = 0, \ldots, d$, so $n_d = n$. They generate all re-encryption matrices $\mathbf{H}_{n_{i+1}:n_i}$ (with success probability $\left(1 - n^{-\Omega(1)}\right)$) and can thus homomorphically evaluate depth d circuits.

The homomorphic evaluation works by performing a homomorphic operation at level i of the evaluated circuit (with i going from 0 to $d-1$), and then using re-encryption with $\mathbf{H}_{n_{i+1}:n_i}$ to obtain a fresh ciphertext for the next level.

For the purposes of our (second) break, we notice that in the last step of this evaluation is re-encryption using $\mathbf{H}_{n_d:n_{d-1}}$. This means that homomorphically evaluated ciphertexts all come from a linear subspace of dimension $n_{d-1} = n^{1/(1+\alpha)}$.

Error Correction and the Matrix $\mathbf{H}_{n:n}$. The scheme **BASIC** only allows homomorphism at the expense of increasing the instance size (namely n). BL show next that it is possible to use **BASIC** to generate a re-encryption matrix without a size increase.

They generate an instance of **BASIC**, with public keys $\mathbf{P}_0, \ldots, \mathbf{P}_d$, secret key $\mathbf{y}_d = \mathbf{y}^*$, and re-encryption matrices $\mathbf{H}_{n_{i+1}:n_i}$. An additional independent instance of $\mathbf{K}_q(n)$ is generated, whose keys we denote by $(\mathbf{P}, \mathbf{y})$. Then, a large number of encryptions of the elements of $\mathbf{y}$ under public key $\mathbf{P}_0$ are generated.[3] While some of these ciphertexts may have encryption error, BL show that homomorphically evaluating a depth-d correction circuit ($CORR$ in their notation), one can obtain a matrix $\mathbf{H}_{n:n}$, whose columns are encryptions of $\mathbf{y}^*$ that are decryptable under $\mathbf{y}$ without error. This process succeeds with probability $\left(1 - n^{-\Omega(1)}\right)$.

The resemblance to the proof of our Theorem 3.2 is apparent. In a sense, the public key of **BASIC** is ready-for-use attacker.

To conclude this part, BL generate a re-encryption matrix $\mathbf{H}_{n:n}$ that takes ciphertexts under $\mathbf{y}$ and produces ciphertexts under $\mathbf{y}^*$. Since $\mathbf{H}_{n:n}$ is produced using homomorphic evaluation, its rank is at most $n_{d-1} = n^{1/(1+\alpha)}$. We will capitalize on the fact that re-encryption using $\mathbf{H}_{n:n}$ produces ciphertexts that all reside in a low-dimensional space.

Achieving Full Homomorphism – The Scheme HOM. The basic idea is to generate a sequence of matrices $\mathbf{H}_{n:n}$, thus creating a chaining of the respective secret keys that will allow homomorphism of any depth. However, generating an arbitrarily large number of such re-encryption matrices will eventually cause an error somewhere down the line. Therefore, a more sophisticated solution is required. BL suggest to encrypt each message a large number of times, and generate a large number of re-encryption matrices per level. Then, since the vast majority of matrices per level

[3] To be absolutely precise, BL encrypt a bit decomposition of $\mathbf{y}^*$, but this is immaterial to us.

are guaranteed to be correct, one can use shallow approximate majority computation to guarantee that the fraction of erroneous ciphertexts per level does not increase with homomorphic evaluation.

Decryption is performed as follows: Each ciphertext is a set of ciphertexts $c_1, \ldots, c_k$ of $\mathbf{K}_q(n)$ (all with the same secret key). The decryption process first uses the $\mathbf{K}_q(n)$ key to decrypt the individual ciphertexts and obtain $m_1, \ldots, m_k$, and then outputs the majority between the values m_i. BL show that a majority of the ciphertexts (say more than 0.6 fraction) are indeed correct, which guarantees correct decryption.

BL can thus achieve a (leveled) fully homomorphic scheme which they denote by **HOM**, which completes their construction.

4.2 A Generic Attack on BL

We can use Theorem 3.2 to break BL as stated below.

Theorem 4.1. *There is a polynomial time CPA attack on BL.*

Proof. The scheme **HOM** is fully homomorphic so it can evaluate any function, in particular majority. However, it is not linearly decryptable.[4] We thus consider a related decryption procedure for **HOM**, one which is linear but has decryption error: Recalling the decryption of **HOM** described above, if instead of decrypting all of the c_i's and taking majority, we will just decrypt c_1 (this is $\mathbf{K}_q(n)$ decryption and thus linear). This simple (and linear) decryption will produce the correct response with probability at least 0.6 (for all but negligibly few key pairs). This is sufficient for Theorem 3.2 to render the scheme insecure. □

4.3 A Specific Attack on BL

We noticed that the scheme **BASIC** which is a component of **HOM** contains by design homomorphic evaluation of majority: this is how the matrix $\mathbf{H}_{n:n}$ is generated. We thus present an attack that only uses the matrix $\mathbf{H}_{n:n}$ and allows to completely decrypt BL ciphertexts with probability $1 - n^{-\Omega(1)}$. We recall that an attack *completely breaks* a scheme if it can decrypt any given ciphertext with probability $1 - o(1)$.

Theorem 4.2. *There exists a polynomial time attack that completely breaks* **BASIC***, and thus also BL.*

Proof. We consider the re-encryption matrix $\mathbf{H} = \mathbf{H}_{n:n} \in \mathbb{F}_q^{n \times n}$ described in Section 4.1, which re-encrypts ciphertexts under $\mathbf{y}$ into ciphertexts under $\mathbf{y}^*$. The probability that $\mathbf{H}$ was successfully generated is at least $1 - n^{-\Omega(1)}$, in which case it holds that

$$\mathbf{y}^{*T} \cdot \mathbf{H} = \mathbf{y}^T \ .$$

In addition, as we explained in Section 4.1, the rank of $\mathbf{H}$ is at most $h = n^{1/(1+\alpha)}$.

Our breaker will be given $\mathbf{H}$ and the public key $\mathbf{P}$ that corresponds to $\mathbf{y}$, and will be able to decrypt any vector $\mathbf{c} = \mathsf{Enc}_{\mathbf{P}}(m)$ with high probability, namely compute $\langle \mathbf{y}, \mathbf{c} \rangle$.

[4] Note that **HOM** has negligible decryption error, so if it was linearly decryptable it would have been trivial to break.

Breaker Code. As explained above, the input to the breaker is $\mathbf{H}, \mathbf{P}$ and challenge $\mathbf{c} = \mathsf{Enc}_{\mathbf{P}}(m)$. The breaker will execute as follows:

1. Generate $k = h^{1+\epsilon}$ encryptions of 0, denoted $\mathbf{v}_1, \ldots, \mathbf{v}_k$, for $\epsilon = \frac{\alpha(1-\alpha)}{4}$ (any positive number smaller than $\frac{\alpha(1-\alpha)}{2}$ will do).

   Note that this means that with probability $1 - n^{-\Omega(1)}$, all $\mathbf{v}_i$ are decryptable encryptions of 0. Intuitively, these vectors, once projected through $\mathbf{H}$, will span all decryptable encryptions of 0.

2. For all $i = 1, \ldots, k$, compute $\mathbf{v}_i^* = \mathbf{H} \cdot \mathbf{v}_i$ (the projections of the ciphertexts above through $\mathbf{H}$). Also compute $\mathbf{o}^* = \mathbf{H} \cdot \mathbf{1}$ (the projection of the all-one vector).

3. Find a vector $\tilde{\mathbf{y}}^* \in \mathbb{F}_q^n$ such that $\langle \tilde{\mathbf{y}}^*, \mathbf{v}_i^* \rangle = 0$ for all i, and such that $\langle \tilde{\mathbf{y}}^*, \mathbf{o}^* \rangle = 1$. Such a vector necessarily exists if all $\mathbf{v}_i$'s are decryptable, since $\mathbf{y}^*$ is an example of such a vector.

4. Given a challenge ciphertext $\mathbf{c}$, compute $\mathbf{c}^* = \mathbf{H} \cdot \mathbf{c}$ and output $m = \langle \tilde{\mathbf{y}}^*, \mathbf{c}^* \rangle$ (namely, $m = \tilde{\mathbf{y}}^{*T} \cdot \mathbf{H} \cdot \mathbf{c}$).

Correctness. To analyze the correctness of the breaker, we first notice that the space of ciphertexts that decrypt to 0 under $\mathbf{y}$ is linear (this is exactly the orthogonal space to $\mathbf{y}$). We denote this space by Z. Since $\mathbf{1} \notin Z$, we can define the cosets $Z_m = Z + m \cdot \mathbf{1}$. We note that all legal encryptions of m using $\mathbf{P}$ reside in Z_m.

We let Z^* denote the space $\mathbf{H} \cdot Z$ (all vectors of the form $\mathbf{H} \cdot \mathbf{z}$ such that $\mathbf{z} \in Z$). This is a linear space with dimension at most h. Similarly, define $Z_m^* = Z^* + m \cdot \mathbf{o}^*$.

Consider the challenge ciphertext $\mathbf{c} = \mathsf{Enc}_{\mathbf{P}}(m)$. We can think of $\mathbf{c}$ as an encryption of 0 with an added term $m \cdot \mathbf{1}$. We therefore denote $\mathbf{c} = \mathbf{c}_0 + m \cdot \mathbf{1}$. Again this yields a $\mathbf{c}_0^*$ such that $\mathbf{c}^* = \mathbf{c}_0^* + m \cdot \mathbf{o}^*$.

Now consider the distribution $\mathcal{Z}$ over Z, which is the distribution of decryptable encryptions of 0 (i.e. the distribution $\mathbf{c} = \mathsf{Enc}_{\mathbf{P}}(0)$, conditioned on $\langle \mathbf{y}, \mathbf{c} \rangle = 0$). The distribution $\mathcal{Z}^*$ is defined by projecting $\mathcal{Z}$ through $\mathbf{H}$. With probability $\left(1 - n^{-\Omega(1)}\right)$, it holds that $\mathbf{v}_1^*, \ldots, \mathbf{v}_k^*$, and $\mathbf{c}_0^*$ are uniform samples from $\mathcal{Z}^*$.

By Lemma 2.1 below, it holds that $\mathbf{c}_0^* \in \mathrm{Span}\left\{\mathbf{v}_1^*, \ldots, \mathbf{v}_k^*\right\}$, with probability $\left(1 - n^{-\Omega(1)}\right)$. In such case

$$\langle \tilde{\mathbf{y}}^*, \mathbf{c}^* \rangle = \langle \tilde{\mathbf{y}}^*, \mathbf{c}_0^* \rangle + m \cdot \langle \tilde{\mathbf{y}}^*, \mathbf{o}^* \rangle = m \ .$$

We conclude that with probability $1 - n^{-\Omega(1)}$, our breaker correctly decrypts $\mathbf{c}$ as required. □

Acknowledgements

The author wishes to thank the MIT-BU cryptography reading group for introducing the BL scheme to him, and especially to Stefano Tessaro and Shafi Goldwasser for various discussions. We further thank Andrej Bogdanov for his comments on a preprint of this manuscript, and for the discussions that followed, leading to the general argument about homomorphic schemes.

References

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *FOCS*, pages 298–307. IEEE Computer Society, 2003.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. ITCS, 2012. See also `http://eprint.iacr.org/2011/277`.

[BL11] Andrej Bogdanov and Chin Ho Lee. Homomorphic encryption from codes. Cryptology ePrint Archive, Report 2011/622, 2011. `http://eprint.iacr.org/`.

[Bra12] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. Cryptology ePrint Archive, Report 2012/078, 2012. `http://eprint.iacr.org/`.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In *CRYPTO*, volume 6841, page 501, 2011.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. Efficient fully homomorphic encryption from (standard) LWE. In Rafail Ostrovsky, editor, *FOCS*, pages 97–106. IEEE, 2011. References are to full version: `http://eprint.iacr.org/2011/344`.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.

[Gen10] Craig Gentry. Toward basing fully homomorphic encryption on worst-case hardness. In *CRYPTO*, pages 116–137, 2010.

[GH11] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In Kenneth G. Paterson, editor, *EUROCRYPT*, volume 6632 of *Lecture Notes in Computer Science*, pages 129–148. Springer, 2011.

[GHS12] Craig Gentry, Shai Halevi, and Nigel P. Smart. Fully homomorphic encryption with polylog overhead. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 465–482. Springer, 2012.

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In Harry R. Lewis, Barbara B. Simons, Walter A. Burkhard, and Lawrence H. Landweber, editors, *STOC*, pages 365–377. ACM, 1982.

[GOT12] Valérie Gauthier, Ayoub Otmani, and Jean-Pierre Tillich. A distinguisher-based attack of a homomorphic encryption scheme relying on reed-solomon codes. Cryptology ePrint Archive, Report 2012/168, 2012. `http://eprint.iacr.org/`.

[GRS08] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. How to encrypt with the lpn problem. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 679–690. Springer, 2008.

[SV10] Nigel P. Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 420–443. Springer, 2010.

[Vai11] Vinod Vaikuntanathan. Computing blindfolded: New developments in fully homomorphic encryption. In Rafail Ostrovsky, editor, *FOCS*, pages 5–16. IEEE, 2011.

[vDGHV10] Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, pages 24–43, 2010. Full Version in `http://eprint.iacr.org/2009/616.pdf`.