Electronic Colloquium on Computational Complexity, Report No. 157 (2012)

On the depth complexity of homomorphic encryption schemes

Andrej Bogdanov* Chin Ho Lee†

Abstract

We show that secure homomorphic evaluation of any non-trivial functionality of sufficiently many inputs with respect to any CPA secure encryption scheme cannot be implemented by constant depth, polynomial size circuits, i.e. in the class AC^0. In contrast, we observe that certain previously studied encryption schemes (with quasipolynomial security) can be implemented in AC^0. We view this as evidence that encryption schemes that support homomorphic evaluation are inherently more complex than ordinary ones.

1 Introduction

A central objective in the theory of cryptography is to classify the relative complexity of various cryptographic tasks. One common way of arguing that task B is of comparable easiness to task A is to give a black-box implementation of B using A as a primitive. Notable examples include the construction of pseudorandom generators from one-way permutations [GL89] and one-way functions [HILL99, HRV10].

But how should we argue that task B is "more complex" than task A? In the generic setting, one looks for the existence of a black-box separation [IR89, RTV04], or a lower bound on the query complexity of a black-box reduction [GT00]. However such black box impossibility results are not always a good indicator of the relative complexity of the two tasks in the real world (under suitable complexity assumptions). For example, although collision-resistant hash functions cannot be constructed from one-way functions in a black-box manner [Sim98], both objects have simple, local (NC^0) implementations under standard assumptions [AIK07].

An alternative way to argue that task B is more complex than task A is to provide a concrete complexity model in which one can implement A (under plausible assumptions), but not B. For example, Applebaum et al. [AIK07] show that under plausible complexity assumptions, nontrivial pseudorandom generators can be implemented in the complexity class NC^0. However, it is not difficult to see that this class does not contain pseudorandom functions; in fact, Linial, Mansour, and Nisan [LMN93] show that pseudorandom functions cannot be implemented even in AC^0. Taken together, these results may be viewed as concrete evidence that pseudorandom functions are more complex than pseudorandom generators, despite the existence of a black-box reduction [GGM86] and the lack of lower bounds on the complexity of such reductions [MV11].

In this work we give concrete complexity-theoretic evidence that encryption schemes that support homomorphic evaluation of essentially any non-trivial functionality are more complex than

*andrejb@cse.cuhk.edu.hk. Dept. of Computer Science and Engineering and Institute for Theoretical Computer Science and Communications, Chinese University of Hong Kong. Work supported by grants RGC GRF CUHK410309 and CUHK410111.

†chlee@cse.cuhk.edu.hk. Dept. of Computer Science and Engineering, Chinese University of Hong Kong.

ISSN 1433-8092

ordinary encryption schemes. Our main result (Theorem 3.1) shows that homomorphic evaluation of any non-trivial functionality (for example the AND function) that depends on sufficiently many inputs cannot be implemented by circuits of constant depth and subexponential size with respect to any CPA secure encryption scheme. In Section 4 we review some proposals of CPA secure private key encryption schemes of quasipolynomial security that can be implemented in this model. In the public key setting, we observe that the cryptosystem of Applebaum, Barak, and Wigderson [ABW10] can be implemented in constant depth.

Thus constant-depth circuits provide sufficient computational power for implementing both private and public-key encryption schemes (under previously studied assumptions), but not variants of such schemes that support homomorphic evaluation of any non-trivial functionality.

2 Definitions

In this section we give a definition of what it means for an algorithm E to homomorphically evaluate a given functionality f. A fairly weak requirement is that a homomorphic evaluator for $f(m_1, \ldots, m_k)$ should take as inputs encryptions of $m_1, \ldots, m_k$ and output a ciphertext that decrypts to $f(m_1, \ldots, m_k)$.

We will allow for the evaluation algorithm to err on some fraction of the encryptions. This takes into account the possibility that the encryption scheme itself may produce incorrect encryptions with some probability.

Definition 2.1. Let $(\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ be a private-key encryption scheme over message set Σ with ciphertexts in $\{0,1\}^n$. We say a circuit E is a *homomorphic evaluator* of $f : \Sigma^k \to \Sigma$ with error δ if for all $m_1, \ldots, m_k \in \Sigma$,

$$\Pr[\mathbf{Dec}_{SK}(E(\mathbf{Enc}_{SK}(m_1, R_1), \ldots, \mathbf{Enc}_{SK}(m_k, R_k))) = f(m_1, \ldots, m_k)] \geq 1 - \delta,$$

where $SK \sim \mathbf{Gen}$ is a uniformly chosen secret key and $R_1, \ldots, R_k$ are independent random seeds.

In the public-key setting, we are given an encryption scheme $(\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ and require that

$$\Pr[\mathbf{Dec}_{SK}(E(PK, \mathbf{Enc}_{PK}(m_1, R_1), \ldots, \mathbf{Enc}_{PK}(m_k, R_k))) = f(m_1, \ldots, m_k)] \geq 1 - \delta.$$

where $(PK, SK) \sim \mathbf{Gen}$ is a random key pair.

We point out one challenge that this natural definition poses in the context of ruling out the existence of homomorphic evaluators. When k is much smaller than n, the definition allows for plausible encryption schemes that admit trivial homomorphic evaluators, by "outsourcing" the homomorphic evaluation to the decryption algorithm. For example suppose that the meaningful portion of an encryption is only captured in the first n/k bits of the ciphertext. Then the homomorphic evaluator can simply copy the meaningful portion of its k encryptions in non-overlapping parts of the output. Upon seeing a ciphertext of this form, the decryption algorithm can easily compute the value $f(m_1, \ldots, m_k)$ by first decrypting the ciphertext corresponding to each of the k encryptions and then evaluating f.

Thus our negative result will only apply to functions whose number of relevant inputs k is sufficiently large in terms of n. Beyond this requirement, we do not make any assumption on f.

The requirement we make on the encryption scheme is CPA message indistinguishability. A private-key encryption scheme is (s, d, ε) CPA message indistinguishable if for every pair of messages

$m, m' \in \Sigma$ and every distinguishing oracle circuit $D^?$ of size s and depth d,

$$|\Pr_{SK,R}[D^{\mathbf{Enc}(SK,\cdot)}(\mathbf{Enc}_{SK}(m,R)) = 1] - \Pr_{SK,R}[D^{\mathbf{Enc}(SK,\cdot)}(\mathbf{Enc}_{SK}(m',R)) = 1]| \leq \varepsilon.$$

In the public key setting CPA security follows from ordinary message indistinguishability:

$$|\Pr_{PK,R}[D(PK, \mathbf{Enc}_{PK}(m,R)) = 1] - \Pr_{PK,R}[D(PK, \mathbf{Enc}_{PK}(m',R)) = 1]| \leq \varepsilon.$$

3 Homomorphic evaluation requires depth

Theorem 3.1. *Suppose* $(\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ *is an* $(2s + k + O(1), d + 1, 1/6)$ *CPA message indistinguishable private-key (resp. public-key) encryption scheme. Let* E *be a homomorphic evaluator of size* s *and depth* d *with error at most* $1/3$ *for some* $f : \Sigma^k \to \Sigma$ *that depends on all of its inputs with respect to this scheme. Then* $s > 2^{\Omega((k/6n)^{1/(d-1)})}$.

For notational simplicity, we present the proof for the private key variant. Since f depends on all its inputs, for every $i \in [k]$ there is a pair of messages m and m' that differ only in coordinate i such that $f(m) \neq f(m')$. Now suppose E is a homomorphic evaluator for f with error $1/3$. Then

$$\Pr[\mathbf{Dec}(E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m_i, R_i), \ldots, \mathbf{Enc}(m_k, R_k))) \neq f(m)] \leq 1/3 \quad \text{and}$$
$$\Pr[\mathbf{Dec}(E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m'_i, R'_i), \ldots, \mathbf{Enc}(m_k, R_k))) \neq f(m')] \leq 1/3,$$

where the probability is taken over the choice of secret key SK (which we omit to simplify notation), the random choice of $i \sim [k]$, and the randomness $R_1, \ldots, R_i, R'_i, \ldots, R_k$ used in the encryption. Since $f(m) \neq f(m')$, it follows that

$$\Pr[\mathbf{Dec}(E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m_i, R_i), \ldots, \mathbf{Enc}(m_k, R_k))) \neq \mathbf{Dec}(E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m'_i, R'_i), \ldots, \mathbf{Enc}(m_k, R_k)))] \geq 1/3.$$

Therefore it must be that

$$\Pr[E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m_i, R_i), \ldots, \mathbf{Enc}(m_k, R_k)) \neq E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m'_i, R'_i), \ldots, \mathbf{Enc}(m_k, R_k))] \geq 1/3.$$

By CPA message indistinguishability,

$$\Pr[E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m'_i, R'_i), \ldots, \mathbf{Enc}(m_k, R_k)) \neq E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m_i, R'_i), \ldots, \mathbf{Enc}(m_k, R_k))] \leq 1/6$$

and therefore we can replace m'_i by m_i in the last inequality to obtain

$$\Pr[E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m_i, R_i), \ldots, \mathbf{Enc}(m_k, R_k)) \neq E(\mathbf{Enc}(m_1, R_1), \ldots, \mathbf{Enc}(m_i, R'_i), \ldots, \mathbf{Enc}(m_k, R_k))] \geq 1/6. \quad (1)$$

Lemma 3.2. *Let* $D_1, \ldots, D_k$ *be any distributions over* $\{0,1\}^n$. *Let* $g : (\{0,1\}^n)^k \to \{0,1\}$ *be a circuit of size* s *and depth* d *where* $s \leq 2^{(\varepsilon k)^{1/(d-1)}/K}$ *for some absolute constant* K. *Then*

$$\Pr[g(X_1, \ldots, X_i, \ldots, X_k) \neq g(X_1, \ldots, X'_i, \ldots, X_k)] < \varepsilon$$

where the randomness is taken over the choice of $i \sim [k]$ *and independent samples* $X_1 \sim D_1, \ldots, X_i, X'_i \sim D_i, \ldots, X_k \sim D_k$.

We apply this Lemma with D_i equal to the distribution of encryptions of m_i and $\varepsilon = 1/6n$ to each of the n outputs of E and take a union bound to conclude that (1) is violated unless $s > 2^{\Omega((k/6n)^{1/(d-1)})}$.

Proof of Lemma 3.2. Fix any pair $Z, Z' \in (\{0,1\}^n)^k$. For any $w \in \{0,1\}^k$, let $Z_w \in (\{0,1\}^n)^k$ be the string such that

$$\text{the } i\text{-th block of } Z_w = \begin{cases} \text{the } i\text{-th block of } Z, & \text{if } w_i = 0 \\ \text{the } i\text{-th block of } Z', & \text{if } w_i = 1. \end{cases}$$

Let $h_{Z,Z'}(w) = g(Z_w)$. Then h is of size at most s and depth at most d. By Boppana [Bop97], for every Z and Z' we have

$$\Pr_{W,i}[h_{Z,Z'}(W) \neq h_{Z,Z'}(W + e_i)] \leq (K \log s)^{d-1}/k$$

for some constant K. Therefore for Z, Z' sampled independently from $D_1 \times \cdots \times D_k$ we have

$$\begin{aligned} \Pr[g(X_1, \ldots, X_i, \ldots, X_k) \neq g(X_1, \ldots, X'_i, \ldots, X_k)] &= \mathrm{E}_{Z,Z'}[\Pr_{W,i}[h_{Z,Z'}(W) \neq h_{Z,Z'}(W + e_i)]] \\ &= \mathrm{E}_{Z,Z'}[(K \log s)^{d-1}/k] \\ &= (K \log s)^{d-1}/k. \end{aligned}$$

It follows that if this probability is at most ε, then $s \leq 2^{(\varepsilon k)^{1/(d-1)}/K}$. □

A similar lemma was proved by Blais, O'Donnell, and Wimmer [BOW10] for noise sensitivity of boolean functions. Here we adapted their argument to influence.

4 On CPA secure encryption schemes in AC^0

In this section we review the depth complexity of some studied candidate CPA secure encryption schemes. To begin with, we observe that asymptotically superpolynomial security cannot be achieved by NC^0 decryption circuits: If every output of the decryption circuit depends on at most d bits of the ciphertext, then for any message m the decryption circuit on the distribution of encryptions of m can be PAC-learned in time $O_d(n^d)$, violating CPA security.

Kharitonov [Kha93] implicitly shows the existence of a "weakly pseudorandom" function family in AC^0 that is $2^{\mathrm{poly}\log n}$ hard to predict on a uniformly random input even from membership queries (assuming Blum integers are sufficiently hard to factor). This function family can be used to obtain a CPA secure symmetric key encryption scheme whose encryption and decryption algorithms are in AC^0. However, we do not know if key generation (which involves generating random Blum integers of magnitude $2^{\mathrm{poly}\log n}$) can be performed in AC^0. Gilbert et al. [GRS08] give a probabilistic CPA secure symmetric key encryption scheme whose security can be reduced to the hardness of the Learning Parity with Noise (LPN) problem. The current best known algorithm [BKW00] for the LPN problem over $\{0,1\}^m$ runs in time $2^{\Theta(m/\log m)}$. Assuming this is optimal, by setting $m = (\log n)^d$ one can implement all components of this scheme using circuits of size $\mathrm{poly}(n)$ and depth $d + O(1)$, and the scheme has security $2^{\Theta((\log n)^d/\log\log n)}$.

We are not aware of any implementation of a public key encryption scheme with all but negligible security all of whose components are in AC^0. Here we show that the cryptosystem proposed by

Applebaum, Barak and Wigderson [ABW10] can be implemented using circuits of polynomial size and constant depth in the security parameter. The variant of the cryptosystem we discuss is conjectured to have security $n^{\Omega(\log n)}$.[1]

First we review the key generation, encryption and decryption in the ABW encryption scheme. One can refer to [ABW10] for further details. Then we show how to implement each operation in constant depth.

The public key is a random bipartite graph $G = ((U, V), E)$, where $|U| = n$ and $|V| = r = n^{0.9}$, generated in the following way. First choose a random subset $S \subseteq U$ and $T \subseteq V$ of size s and $s/3$ respectively, where $s = O(\log n)$. Each vertex in S is connected to d (possibly repeated) random vertices in T and each vertex outside S is connected to d random vertices in V. The secret key SK is an odd size subset of S such that each vertex in T has an even number of neighbors in SK.

To encrypt a message $m \in \{0, 1\}$, choose a random subset T' of V and output $y + e + m\mathbf{1}$, where each coordinate of $y \in \{0, 1\}^n$ is the degree of the corresponding vertex in U restricted to T' mod 2, $e \in \{0, 1\}^n$ is a vector with each coordinate sampled from a distribution $\hat{\eta}$ with $\Pr[\hat{\eta} = 0] = \eta$ independently, and $\mathbf{1} \in \{0, 1\}^n$ is the all ones vector.

To decrypt a ciphertext $c \in \{0, 1\}^n$, output $\sum_{i \in SK} c_i$. Now we give an AC^0 implementation of the cryptosystem.

Implementation of the ABW cryptosystem in AC^0

Key Generation: Sample

1. $y_1, y_2, \ldots, y_s$ from $[n]$ and $w_1, w_2, \ldots, w_{s/3}$ from $[r]$ to represent the subsets $S \subseteq U$ and $T \subseteq V$, respectively;

2. $v_{i,1}, \ldots, v_{i,d}$ from $[r]$ for every i from 1 to n. These are the random neighbors of each vertex i in $U \setminus S$;

3. $\hat{v}_{i,1}, \ldots, \hat{v}_{i,d}$ from $[s/3]$ for every i from 1 to s. These become the random neighbors of the vertices in S after being mapped to the w_i's by the index function $\iota : [s/3] \to [r]$ such that $\iota(i) = w_i$. This function can be written as

$$\iota(i) = \bigvee_{j=1}^{s/3} [(i = j) \wedge w_j].$$

The key generation circuit outputs $v_{i,1}, \ldots v_{i,d}$ if the vertex i is not in S, and outputs $\iota(\hat{v}_{i,1}), \iota(\hat{v}_{i,2}), \ldots, \iota(\hat{v}_{i,d})$ otherwise. Now we can output the jth random neighbor of each vertex $i \in U$ by

$$\left[\delta_i \wedge \bigvee_{k=1}^{s} [(i = y_k) \wedge \iota(\hat{v}_{k,j})]\right] \vee (\overline{\delta_i} \wedge v_{i,j}),$$

where $\delta_i := \bigvee_{k=1}^{s} (i = y_k)$ indicates whether i belongs to S.

To come up with the secret key SK, we enumerate all the possible subsets of S (recall that $s = O(\log n)$) and output the first one that satisfies the linear dependency. Given an odd size

[1] Owing to the existence of a quasipolynomial time algorithm for learning from random examples [LMN93], if ciphertexts are computationally indistinguishable from the uniform distribution, any AC^0 decryption algorithm can be broken in time $2^{\mathrm{poly}\log n}$.

subset of S indicated by the support of the vector $a \in \{0,1\}^s$. It is not difficult to see that the formula

$$f_a = \bigvee_{j=1}^{s/3} \bigoplus_{i:a_i=1} \bigoplus_{k=1}^{d} (\hat{v}_{i,k} = j)$$

outputs 0 if every vertex in T has an even number of neighbors in the support of a and outputs 1 otherwise. (Since the XOR involves only $O(d \log n)$ inputs, it can be implemented in depth two and size $n^{O(d)}$.) Thus we can enumerate all the possible $a \in \{0,1\}^s$ of odd hamming weight and output the first subset a with $f_a = 0$. The secret key is represented by a vector z containing s entries in $[n]$, where each nonzero entry corresponds to a vertex in SK. More precisely, we output the ith entry as

$$z_i = \iota \left(\bigvee\nolimits_{a \in \{0,1\}^s : wt(a) \text{ is odd}} \left[\overline{f_a} \wedge \left(\bigwedge\nolimits_{b<a} f_b \right) \wedge (a_i \wedge i) \right] \right).$$

Encryption: Given a public key represented by the neighbors $v_{i,1}, \ldots, v_{i,d}$ of each vertex i in U. To encrypt a message $m \in \{0,1\}$, choose a random vector x in $\{0,1\}^r$ whose support forms the subset T' of V, a noise vector $e \in \{0,1\}^n$ by choosing each of its entries independently from $\hat{\eta}$. The ith bit of the encryption can be written as

$$c_i = \bigvee_{\substack{k_i \neq k_j, 1 \leq i < j \leq d, k_i \in [r] \\ a_1, \ldots, a_d:\ a_1 + \cdots + a_d = 1}} [\bigwedge\nolimits_{j=1}^{d} (v_{i,j} = k_j) \wedge (x_{k_1} = a_1) \wedge \cdots \wedge (x_{k_d} = a_d)] \oplus e_i \oplus m.$$

Decryption: Given a ciphertext c and the secret key SK represented by the vector $z \in \{0,1\}^{s \times \log n}$, output

$$\bigoplus_{i=1}^{s} \bigvee_{k=1}^{n} [(z_i = k) \wedge c_k].$$

Reducing the encryption error The ABW cryptosystem (as well as the LPN-based system of Gilbert et al.) has noticeable encryption error. The encryption error can be made negligible by encrypting the message independently multiple times. While some of the multiple encryptions may be erroneous, with all but negligible probability at least 2/3 of them will be correct. The errors can be corrected by taking approximate majority at the decryption stage, which can be implemented using circuits of depth 3 [Ajt83], thereby preserving the constant depth complexity of the implementation.

References

[ABW10] B. Applebaum, B. Barak, and A. Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the 42th ACM Symposium on Theory of Computing*, pages 171–180, 2010.

[AIK07] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography with constant input locality. In *Proceedings of the 27th annual international cryptology conference on Advances in cryptology*, CRYPTO'07, pages 92–110, Berlin, Heidelberg, 2007. Springer-Verlag.

[Ajt83] M. Ajtai. Σ_1^1-formulae on finite structures. In *Annals of Pure and Applied Logic*, volume 24, pages 607–620, 1983.

[BKW00] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 435–440, 2000.

[Bop97] Ravi B. Boppana. The average sensitivity of bounded-depth circuits. *Inf. Process. Lett.*, 63(5):257–261, September 1997.

[BOW10] Eric Blais, Ryan ODonnell, and Karl Wimmer. Polynomial regression under arbitrary product distributions. *Machine Learning*, 80:273–294, 2010.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, August 1986.

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, STOC '89, pages 25–32, New York, NY, USA, 1989. ACM.

[GRS08] Henri Gilbert, Matthew J. Robshaw, and Yannick Seurin. How to encrypt with the LPN problem. In *Proceedings of the 35th international colloquium on Automata, Languages and Programming, Part II*, ICALP '08, pages 679–690, Berlin, Heidelberg, 2008. Springer-Verlag.

[GT00] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, pages 305–, Washington, DC, USA, 2000. IEEE Computer Society.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, March 1999.

[HRV10] Iftach Haitner, Omer Reingold, and Salil Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 437–446, New York, NY, USA, 2010. ACM.

[IR89] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, STOC '89, pages 44–61, New York, NY, USA, 1989. ACM.

[Kha93] Michael Kharitonov. Cryptographic hardness of distribution-specific learning. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 372–381, New York, NY, USA, 1993. ACM.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant-depth circuits, fourier transform and learnability. *Journal of the ACM*, 40(3), 1993.

[MV11] Eric Miles and Emanuele Viola. On the complexity of non-adaptively increasing the stretch of pseudorandom generators. In *Proceedings of the 8th conference on Theory of cryptography*, TCC'11, pages 522–539, Berlin, Heidelberg, 2011. Springer-Verlag.

[RTV04] Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In Moni Naor, editor, *Theory of Cryptography*, volume 2951 of *Lecture Notes in Computer Science*, pages 1–20. Springer Berlin Heidelberg, 2004.

[Sim98] Daniel Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In Kaisa Nyberg, editor, *Advances in Cryptology EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 334–345. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0054137.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de