# Homomorphic Encryption: from Private-Key to Public-Key

Ron Rothblum*

Weizmann Institute of Science, Rehovot, Israel
ron.rothblum@weizmann.ac.il

**Abstract.** We show how to transform any additively homomorphic private-key encryption scheme that is compact, into a public-key encryption scheme. By compact we mean that the length of a homomorphically generated encryption is independent of the number of ciphertexts from which it was created. We do not require anything else on the distribution of homomorphically generated encryptions (in particular, we do not require them to be distributed like real ciphertexts).
Our resulting public-key scheme is homomorphic in the following sense. If the private-key scheme is $i+1$-hop homomorphic with respect to some set of operations then the public-key scheme we construct is $i$-hop homomorphic with respect to the same set of operations.

## 1 Introduction

Homomorphic encryption is a paradigm that refers to the ability, given encryptions of some messages, to generate an encryption of a value that is related to the original messages. Specifically, this ability means that from encryptions of $k$ messages $m_1, \ldots, m_k$ it is possible to generate an encryption of $m^* = f(m_1, \ldots, m_k)$ for some (efficiently computable) function $f$. Ideally, one may want the homomorphically generated encryption of $m^*$ to be distributed identically (or statistically close) to a standard encryption of $m^*$ (even given the original ciphertexts). We call schemes that have this property distribution-preserving homomorphic encryption schemes. Indeed, some proposed homomorphic encryption schemes are distribution-preserving w.r.t some algebraic operations such as addition or multiplication (e.g. Goldwasser-Micali [10], El-Gamal [5]).

For some applications, it seems as though distribution-preserving homomorphic encryption is an overkill. There are weaker notions of homomorphic encryption that might be easier to construct and still suffice for these applications. The very minimal requirement is that a homomorphically generated encryption decrypts correctly to the corresponding message. Alas, this minimalistic requirement does not seem to be useful as is, because it captures schemes that we do not really consider to be homomorphic: Actually, *any* encryption scheme can

* This research was partially supported by the Israel Science Foundation (grant No. 1041/08).

be slightly modified to satisfy this requirement w.r.t *any* efficient operation[1]. A more meaningful notion is obtained by restricting the length of the homomorphically generated encryption. Specifically, we call a homomorphic encryption scheme compact if homomorphically generated encryptions properly decrypt to the correct message *and* their lengths depend *only* on the security parameter and the message length (and not on the number of input ciphertexts). Note that every distribution preserving homomorphic scheme is compact whereas the converse does not hold. Thus, the compactness property is strictly weaker than being distribution-preserving.

### 1.1 Private-Key vs. Public-Key

When discussing homomorphic encryption, we did not specify whether we consider private-key or public-key encryption schemes. Indeed, one can define homomorphic encryption in both settings (with only minor differences). The focus of this paper is showing the connection between public-key and private-key homomorphic encryption.

The easy direction is showing that a public-key homomorphic encryption scheme can be transformed into a private-key homomorphic scheme. This transformation is quite simple and involves only a minor issue. Intuitively, it seems as though any public-key homomorphic scheme *is* a private-key homomorphic scheme. The only problem is that in the public-key setting (in contrast to the private-key one), the homomorphic evaluation algorithm is also given the encryption-key. A simple transformation that addresses this issue is to append the encryption-key to each ciphertext. The resulting private-key scheme clearly retains the homomorphic properties of the public-key scheme (this holds for both distribution-preserving or merely compact homomorphic schemes).

The harder direction is showing that a private-key homomorphic encryption scheme implies a public-key one. This direction will be addressed by our main result, Theorem 2, which basically states that any *compact* additively homomorphic *private-key* encryption scheme can be transformed into a public-key encryption scheme. We present two such constructions both of which partially retain the homomorphic properties of the underlying private-key scheme (see Section 1.2).

We note that it is quite easy to transform a *distribution preserving* homomorphic private-key scheme into a distribution preserving homomorphic public-key one. In fact, this transformation was used by Barak [1] in his exposition of the work of van Dijk et al. [3]. For further discussion, see Section 1.4.

---

[1] Consider implementing the homomorphic evaluation algorithm as the identity function. That is, given ciphertexts and a description of an operation, just output both. Then, modify the decryption algorithm to first decrypt all the ciphertexts and then apply the operation to the decrypted messages. Thus, homomorphic evaluation is delegated to the decryption algorithm that, using the decryption key, can trivially evaluate the required operation.

### 1.2 Homomorphic Properties of the Public-Key Scheme

So far we have described homomorphic evaluation as a one-shot process, however one can consider repeated applications of the homomorphic evaluation algorithm. For *distribution-preserving* homomorphic encryption it is possible to do this because homomorphically generated values are identical (or statistically close) to real ciphertexts. For *compact* homomorphic encryption, the homomorphically generated encryptions can completely differ from actual ciphertexts, hence it is unclear that it is possible to keep computing on such homomorphically generated data. Gentry et al. [7] called a scheme that supports $i$ such repeated applications an $i$-hop homomorphic encryption scheme.

The public-key schemes that we construct are homomorphic in the following sense. If the original private-key scheme is $(i+1)$-hop homomorphic w.r.t some set of operations (which must include addition modulo 2), then the public-key schemes are $i$-hop homomorphic w.r.t the same set of operations. That is, we lose one application of the homomorphic operation in the construction.

### 1.3 Connection to Prior Work

It is possible to combine previous results to construct a public-key encryption scheme from a compact additively homomorphic private-key scheme. However, the resulting public-key scheme does not (necessarily) retain the homomorphic properties of the private-key scheme. The indirect construction works as follows.

Kushilevitz and Ostrovsky [12] show that a compact additively homomorphic *public-key* scheme can be used to construct a two-message private information retrieval (PIR) protocol but their construction also works when using a private-key scheme (that is compact additively homomorphic). Di Crescenzo et al. [2] show that such a PIR protocol implies a two-message oblivious transfer (OT) protocol which in turn easily implies a public-key encryption scheme.

The public-key scheme constructed by combining these results is not necessarily homomorphic. Our simpler direct constructions retain the homomorphic properties of the private-key scheme (in the sense outlined in Section 1.2). Additionally, our schemes are fairly efficient and do require the amplification step used in [2] (to construct OT from PIR).

### 1.4 Technique

The intuition for how to move from private to public-key can be seen in a more straightforward manner in the case of *distribution preserving* homomorphic encryption. The following construction was suggested implicitly in [1].

Let $E$ and $D$ be the respective encryption and decryption algorithm of a private-key encryption scheme. Suppose that the scheme is distribution-preserving homomorphic w.r.t the identity function. That is, it is possible to "re-randomize" ciphertexts[2]. Such a scheme can be used to construct a public-key bit-encryption

[2] This means that there exists an algorithm $RR$ such that for *any* encryption $c$ of a bit $b$, the output of $RR(c)$ is distributed identically to $E_e(b)$.

scheme[3] as follows. The (private) decryption-key is a key $k$ of the private-key scheme and the (public) encryption-key consists of an encryption of 0 and an encryption of 1 (i.e. $E_k(0)$ and $E_k(1)$). To encrypt a bit $\sigma$ just re-randomize the ciphertext corresponding to $\sigma$. To decrypt, apply the private-key decryption algorithm using $k$ (i.e. $D_k$).

The security of this construction follows from the fact that after re-randomization, all information on the original ciphertext, which was re-randomized, is completely lost. However, if the private-key scheme is only *compactly* homomorphic then we do not have a guarantee on the distribution of the homomorphically generated ciphertext and the above transformation fails. Hence, we use more complicated constructions, outlined next.

We present two constructions of public-key bit-encryption schemes based on any private-key scheme that is compactly homomorphic w.r.t addition modulo 2. The first construction was suggested to us by Yuval Ishai after we discovered the second construction. Both constructions are fairly straightforward but the first construction has a very simple proof. The second construction has a more complex proof based on an information-theoretic theorem, which may be of independent interest.

For both constructions the basic idea is to run the homomorphic algorithm, which outputs at most $m$ bits, on more than $m$ ciphertexts, and so forcing the algorithm to somehow actually compress the input ciphertexts. In the first construction, the decryption-key is once again a key $k$ of the private-key scheme. The encryption-key consists of a random $\ell$ bit string $r$, where $\ell \gg m$, together with a sequence of encryptions of the bits of $r$ using the key $k$. To encrypt a bit $\sigma$, a random vector $s \in \{0,1\}^\ell$ is selected such that the inner product of $r$ and $s$ equals $\sigma$. The homomorphic operation is then applied to the subset of the $\ell$ ciphertexts in the encryption-key, that correspond to coordinates in which $s$ equals 1. Since the ciphertexts in the public-key are encryptions of the bits of $r$, the encryption process produces a homomorphically generated encryption of the inner product of $r$ and $s$, which equals $\sigma$.

To show that this construction is semantically secure we consider, as a mental experiment, changing the key generation algorithm to encrypt zeros instead of the bits of $r$ in the public encryption-key. We then consider an adversary that is given an encryption of a random bit (under this new scheme) and is asked to guess the bit. Observe that the encryption process now depends solely on $s$ and does not contain any information on $r$. Thus, in essence, the adversary is given the $\ell$ bit string $r$ and $m$ bits of information on $s$, where $m \ll \ell$ and is asked to find $\langle r, s \rangle$. Using the Leftover Hash Lemma, we show that it is impossible to predict $\langle r, s \rangle$ with probability that is noticeably greater than $\frac{1}{2}$. Thus, an adversary for the proposed public-key scheme would imply a distinguisher for the underlying private-key scheme that distinguishes between $\ell$ encryptions of random bits and $\ell$ encryptions of 0.

[3] A bit-encryption scheme is a public-key encryption scheme that only handles single-bit messages. Such schemes suffice to construct full-fledged public-key encryption schemes (see [8]).

The second construction is somewhat similar, however its proof of security is more complex and is based on an information-theoretic theorem, which may be of independent interest. Again, the decryption-key is a key $k$ of the private-key scheme but the public-key consists of two lists of ciphertexts; the first is a list of $\ell$ encryptions of 0 and the second is a list of $\ell$ encryptions of 1. To encrypt a bit $\sigma$ we choose a *random subset* $S \subseteq [\ell]$ that has parity $\sigma$ (i.e. $|S| \equiv \sigma \bmod 2$). We use $S$ to select $\ell$ ciphertexts from the public-key by selecting the $i$-th ciphertext from the first list if $i \notin S$ (and from the second if $i \in S$). By homomorphically adding the selected ciphertexts modulo 2, we obtain a ciphertext that correctly decrypts to $\sigma$.

To prove security, once again we consider a mental experiment in which both lists in the public-key are encryptions of 0. Because the mental experiment is computationally indistinguishable from the actual scheme, proving that the original scheme is secure reduces to showing that when *both* lists consist of encryptions of 0, it is essentially impossible to find the parity of the random subset used in the homomorphic encryption process.

We prove the latter via the following information-theoretic theorem: Let $X_1, \ldots, X_\ell$ and $Y_1, \ldots, Y_\ell$ be independent and identically distributed over a finite set $\Omega$ and let $S$ be a random subset of $[\ell]$. We consider the list $Z$, defined as $Z_i = X_i$ for $i \notin S$ and $Z_i = Y_i$ for $i \in S$. The theorem states that it is essentially impossible to guess the parity of $S$ based on $X$, $Y$ and $m$ bits of information on $Z$. That is, any such guess will be correct with probability that is bounded by (roughly) $\frac{1}{2} + 2^{-\Omega(\ell - m)}$. The proof of the information-theoretic theorem makes use of the Efron-Stein decomposition [4], an extension of Fourier analysis for product distributions.

**Remarks.** First we mention that both of our constructions are secure even if we use a weaker definition of *compact* homomorphic encryption. Specifically, when homomorphically adding ciphertexts, the output can be of length that is a sub-linear function of the number of input ciphertexts (rather than being independent of it).

We also mention that while our definition of *compact* homomorphic encryption considers homomorphic operations over *arbitrarily* many ciphertexts, one might instead consider compact homomorphic encryption over just two ciphertexts. However, this definition can be implemented trivially[4] and a more meaningful notion is obtained by requiring that the homomorphic operation support multiple hops. Our constructions can be implemented by a scheme that satisfies the new definition by implementing the homomorphic addition of $\ell$ ciphertexts using a logarithmic number of hops.

### 1.5 Application of our Construction to Fully-Homomorphic Encryption

Our generic transformation from private-key to public-key encryption can be used as a general methodology for constructing (compact) homomorphic public-

[4] As in Footnote 1 while keeping the ciphertexts sufficiently short.

key encryption. One application of this methodology, which actually motivated this work, is to simplify the presentation of the DGHV fully-homomorphic encryption scheme [3].

A fully-homomorphic encryption scheme is an encryption scheme that is homomorphic w.r.t any (efficiently computable) function. The concept of fully-homomorphic encryption was first proposed by Rivest et al. [13] in the 70's, but the first concrete proposal was only made recently in the breakthrough work of Gentry [6].

Building on the work of Gentry [6], van Dijk et al. [3], proposed a simpler fully-homomorphic public-key scheme. From a high-level view, the DGHV fully homomorphic scheme is constructed by first proposing a simple *private-key* homomorphic scheme that is only "somewhat" homomorphic (that is, homomorphic w.r.t some restricted functions), and then showing how to modify this scheme into a somewhat homomorphic *public-key* scheme. Finally, using the bootstrapping technique of [6] the somewhat homomorphic public-key scheme is transformed into a fully-homomorphic public-key scheme.

One way in which our transformation can be used is to replace the aforementioned modification (i.e. from private-key to public-key) that uses specific properties of the DGHV scheme. The advantage is that our transformation is generic. Although the somewhat homomorphic public-key scheme constructed by our transformation is slightly different from the one of [3], the final steps of bootstrapping (see [6]) and reducing the (multiplicative) depth of the decryption circuit can still be applied to both of our constructions.

An alternate way to use our transformation is to first construct a compact fully-homomorphic *private-key* scheme and then, using our generic transformation, to obtain a compact fully-homomorphic *public-key* scheme. We believe that this approach simplifies the presentation of the scheme because the bootstrapping step is done on the simpler private-key scheme. This approach was suggested by Barak [1] for one of the variants of DGHV that is actually *distribution-preserving*. However, using our transformation, the approach can be extended to the compact variants as well.

The two approaches to construct fully homomorphic encryption based on the DGHV scheme are depicted in Figure 1.

## 2 Preliminaries

For a set $S$, we denote by $x \in_R S$ a random element uniformly distributed in $S$. Similarly, we denote by $X \subseteq_R S$ a uniformly distributed random subset of $S$. For a vector $X = X_1, \ldots, X_\ell$ and a set $I \subseteq [\ell]$, we denote by $X_I$ the projection of $X$ to coordinates in $I$; i.e. if $I = \{i_1, \ldots, i_m\}$, where $i_1 < \cdots < i_m$, then $X_I = X_{i_1}, \ldots, X_{i_m}$.

**Non-Standard Notation.** For every $\ell \in \mathbb{N}$, random variables $X = X_1, \ldots, X_\ell$ and $Y = Y_1, \ldots, Y_\ell$ and set $S \subseteq [\ell]$, we denote by $X_{\overline{S}}Y_S$, the random variable $Z = Z_1, \ldots, Z_\ell$ where $Z_i = X_i$ for $i \notin S$ and $Z_i = Y_i$ for $i \in S$.

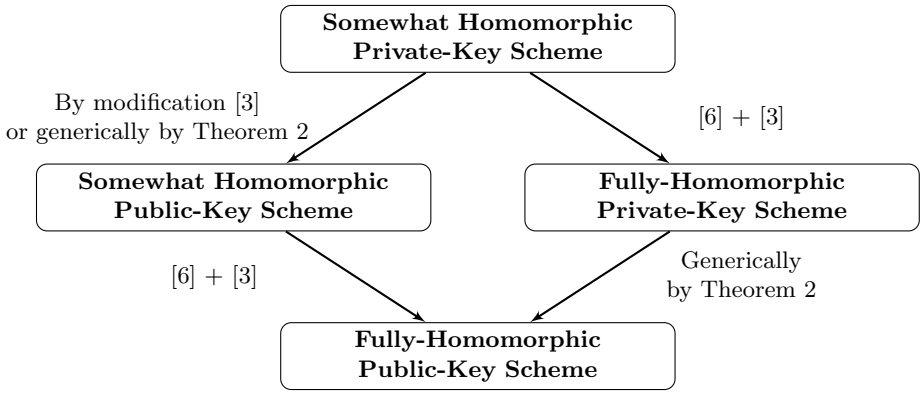


**Fig. 1.** Constructing the compact homomorphic variant of the DGHV fully-homomorphic public-key scheme.

### 2.1 Encryption Schemes

We follow notations and definitions of [8]. In particular we use their definition of semantically secure encryption schemes, both in the private-key and public-key settings. Throughout this paper we restrict our attention to bit-encryption schemes, i.e., schemes that encrypt a single bit. For simplicity, we say public-key (resp. private-key) encryption when we actually mean public-key (resp. private-key) bit-encryption.

When discussing private-key schemes, we consider schemes with *multiple-message* security, i.e., semantic security w.r.t to an adversary that gets encryptions of polynomially many messages. Recall that in the private-key setting (in contrast to public-key one), multiple-message security does not follow from single-message security (see [8, Chapter 5]).

### 2.2 Homomorphic Encryption

Since we only consider *compact* homomorphic encryption, from here on, when we say homomorphic we always mean in the *compact* sense as defined next.

**Definition 1.** $(G, E, D, H)$ *is a* homomorphic public-key encryption scheme *with respect to a set of families of polynomial-sized circuits* $\mathcal{C}$ *if* $(G, E, D)$ *are a public-key encryption scheme,* $H$ *is a probabilistic polynomial-time algorithm and there exists a polynomial*[5] $m(\cdot)$ *such that for every circuit family* $\{C_k\}_{k\in\mathbb{N}} \in \mathcal{C}$*, polynomial* $\ell(\cdot)$*, for every* $n \in \mathbb{N}$*, keys* $(e, d) \leftarrow G(1^n)$*, and* $\ell = \ell(n)$ *single bit messages* $b_1, \ldots, b_\ell \in \{0, 1\}$ *the following holds:*

– *Correct decryption of homomorphically generated encryptions:*

$$D_d\left(H\left(e, C_\ell, E_e(b_1), \ldots, E_e(b_\ell)\right)\right) = C_\ell\left(b_1, \ldots, b_\ell\right). \tag{1}$$

[5] For convenience we assume that $m$ is at least linear.

– *The length of homomorphically generated encryptions is independent of $\ell$:*

$$|H\left(e, C_\ell, E_e(b_1), \ldots, E_e(b_\ell)\right)| \leq m(n). \tag{2}$$

Homomorphic *private-key* encryption is defined analogously (with the modification that $H$ does not get the encryption key as part of its input).

### 2.3 $i$-Hop Homomorphic Encryption

The homomorphic evaluation algorithm in Definition 1 is only required to operate on ciphertexts that were output by the encryption algorithm. The definition does not specify what happens if the homomorphic evaluation algorithm is applied to its own output. Gentry et al. [7] defined an $i$-hop homomorphic encryption scheme as a scheme for which it is possible to apply the homomorphic evaluation algorithm consecutively $i$ times.

Let $G, E, D, H$ be a homomorphic encryption scheme w.r.t to a set of circuit families $\mathcal{C}$. For a given encryption key $e$, we denote by $W_0(e)$ the set of all valid ciphertexts of the encryption scheme, i.e., all possible outputs of the encryption algorithm $E_e$ applied to a single bit message. For $j \geq 1$, we define $W_j(e)$ to be the set of all possible outputs of the homomorphic evaluation algorithm $H$ when applied to a sequence of ciphertexts in $W_{j-1}(e)$ and any circuit $C \in \mathcal{C}$. We say that elements in $W_j(e)$ are $j$-th level ciphertexts and define $i$-hop homomorphic encryption (in both the public and private-key settings) by requiring that Equations (1) and (2) of Definition 1 hold not only for standard ciphertexts (i.e., in $W_0(e)$) but also for $j$-th level ciphertexts for $j \leq i$ (i.e., in $W_j(e)$).

## 3 Constructing a Public-Key Scheme from a Homomorphic Private-Key Scheme

In this section we prove our main theorem:

**Theorem 2.** *Any multiple-message semantically secure private-key encryption scheme that is compactly homomorphic with respect to addition modulo 2 can be transformed into a semantically secure public-key encryption scheme. Furthermore, if the private-key scheme is $(i+1)$-hop homomorphic w.r.t to a set of circuit families, then the constructed public-key scheme is $i$-hop homomorphic w.r.t to the same set.*

We present two alternate constructions. Both constructions are fairly straightforward but the proof of the first construction (Construction 3) is more simple. However, the tools used in the proof of the second construction (Construction 7) may be of independent interest.

To prove Theorem 2, we assume the existence of a private-key scheme $(G, E, D, H)$ that is compactly homomorphic with respect to addition modulo 2 and the polynomial $m(\cdot)$ as in Definition 1. We denote by $H_\oplus$ the algorithm $H$ when applied to the circuit family that computes addition modulo 2. The discussion on the homomorphic properties of the schemes (i.e. the furthermore part of Theorem 2) is presented in Section 4.

### 3.1 First Construction

**Construction 3.** *The public-key encryption scheme $(G', E', D', H')$ is defined as follows:*

**Key Generation -** $G'(1^n)$ :
*Select $k \leftarrow G(1^n)$ and $r \in_R \{0,1\}^\ell$ where $\ell = 4m(n)$.*
*Set $X = (X_1, \ldots, X_\ell)$ where $X_i \leftarrow E_k(r_i)$.*
*Output $(X, r)$ as the public-key and $k$ as the private-key.*

**Encryption -** $E'_{X,r}(\sigma)$ :
*Select at random a vector $s \in \{0,1\}^\ell$ such that $\langle s, r \rangle = \sigma$.*[6] *At convenience, we identity the set $S$ with the natural representation of the vector $s$ as a set, i.e., $S = \{i \colon s_i = 1\}$.*
*Output $H_\oplus(X_S)$ where $X_S$ is the projection of $X$ to coordinates in $S$.*

**Decryption -** $D'_k(c)$ :
*Output $D_k(c)$.*

**Homomorphic Evaluation -** $H'(C, (X, r), c_1, \ldots, c_\ell)$**:**
*Output $H(C, c_1, \ldots, c_\ell)$.*

We start by showing that the decryption algorithm correctly decrypts proper ciphertexts. We then proceed to the main part of the proof, showing that Construction 3 is indeed semantically secure. In Section 4 we discuss the homomorphic properties of the scheme.

**Proposition 4.** *For every $n \in \mathbb{N}$, $\sigma \in \{0,1\}$ and $((X, r), k) \leftarrow G'(1^n)$ it holds that*

$$D'_k\left(E'_{X,r}(\sigma)\right) = \sigma.$$

*Proof.* Based on the first property of homomorphic encryption (Definition 1),

$$D'_k\left(E'_{X,r}(\sigma)\right) = D_k\left(H_\oplus\left(X_S\right)\right) = \bigoplus_{i \in S} D_k(X_i) = \bigoplus_{i \in S} D_k(E_k(r_i))$$

where $S$ is the random subset selected in the encryption algorithm $E'$ and $\bigoplus$ denotes addition modulo 2. Since $D$ decrypts correctly, $D_k(E_k(r_i) = r_i$. Therefore, $D'_k\left(E'_{X,r}\left(\sigma\right)\right) = \bigoplus_{i \in S} r_i = \langle s, r \rangle = \sigma$. □

We proceed to the main part of the proof, showing that Construction 3 is semantically secure.

**Proposition 5.** *If $(G, E, D)$ is a multiple-message semantically secure private-key scheme, then $(G', E', D')$ is a semantically secure public-key scheme.*

[6] If $r = 0^\ell$ then such a vector $s$ does not necessarily exist. However, this case only happens with exponentially vanishing probability and can be handled by choosing $r \neq 0^\ell$ in the key-generation process.

*Proof.* Suppose toward a contradiction that there exists a probabilistic polynomial-time adversary $A$ that can predict the value of a random bit $\sigma$ based on an encryption of $\sigma$. That is, there exists a polynomial $p(\cdot)$ and infinitely many $n \in \mathbb{N}$ for which:

$$\Pr_{\substack{(X,r),k\leftarrow G'(1^n)\\ \sigma\in_R\{0,1\}}}[A\left((X,r),E'_{X,r}(\sigma)\right)=\sigma] > \frac{1}{2}+\frac{1}{p(n)} \tag{3}$$

where the probability is also over the coin tosses of $A$ and $E'$. Based on the definitions of $G'$ and $E'$ this implies that:

$$\Pr_{\substack{s,r\in_R\{0,1\}^\ell,\ k\leftarrow G(1^n)\\ X_i\leftarrow E_k(r_i)}}[A\left(X,r,H_\oplus\left(X_S\right)\right)=\langle s,r\rangle] > \frac{1}{2}+\frac{1}{p(n)} \tag{4}$$

where $S=\{i\colon s_i=1\}$.

Consider $X'$ which is distributed as $\ell$ encryptions of 0 under $k$ (in contrast to $X$ which is distributed as an encryption of the bits of $r$). We claim that for every (computationally unbounded) algorithm $A$ and for every $n \in \mathbb{N}$,

$$\Pr_{\substack{s,r\in_R\{0,1\}^\ell,\ k\leftarrow G(1^n)\\ X'_i\leftarrow E_k(0)}}[A\left(X',r,H_\oplus\left(X'_S\right)\right)=\langle s,r\rangle] \le \frac{1}{2}+3\cdot 2^{-\frac{\ell}{2}+m(n)}\ . \tag{5}$$

Equation (5) implies a simple distinguisher for the private-key scheme. Specifically, we refer to a distinguisher that gets $r \in_R \{0,1\}^\ell$ and $Y$ which is either an encryption of the bits of $r$ or $\ell$ encryptions of zero. The distinguisher chooses a random vector $s \in_R \{0,1\}^\ell$, computes $A(Y,r,H_\oplus(Y_S))$ and outputs 1 if it equals $\langle s,r\rangle$ and 0 otherwise. Based on Equations (4) and (5), this distinguisher distinguishes between the two cases with a noticeable gap. Thus, we only need to show that Eq. (5) holds.

To prove that Eq. (5) holds, we first view it as follows:

$$\Pr_{\substack{s,r\in_R\{0,1\}^\ell,\ k\leftarrow G(1^n)\\ X'_i\leftarrow E_k(0)}}[A_{X'}\left(r,H_{X'}(s)\right)=\langle s,r\rangle] \le \frac{1}{2}+3\cdot 2^{-\frac{\ell}{2}+m(n)}\ . \tag{6}$$

Observe that since $H_{X'}$ does not depend on $r$, the adversary $A_{X'}$ needs to predict $\langle s,r\rangle$ based on $r$ and $m$ bits of information on $s$. The following proposition shows that $A_{X'}$ can only succeed with a negligible advantage, for *every* $k$ and $X'$ (hence, also for $k$ and $X'$ distributed as above):

**Proposition 6.** *For every function $f:\{0,1\}^\ell \to \{0,1\}^m$ and every (computationally unbounded) algorithm $A$,*

$$\Pr_{r,s\in\{0,1\}^\ell}[A(r,f(s))=\langle s,r\rangle] \le \frac{1}{2}+3\cdot 2^{-\frac{\ell}{2}+m}. \tag{7}$$

*Proof.* By the Leftover Hash Lemma[7] [11], because $h_r(s) = \langle s, r\rangle$ is a universal hash function family, for every $\alpha \in \{0,1\}^m$ the distribution $(r, \langle r, s\rangle)$ conditioned on $f(s) = \alpha$ and the distribution $(r, \sigma)$ for $\sigma \in_R \{0,1\}$ are $2\sqrt{\frac{2}{|f^{-1}(\alpha)|}}$-close in statistical distance. Thus:

$$\begin{aligned}
&\Pr_{r,s\in_R\{0,1\}^\ell}[A(r, f(s)) = \langle r, s\rangle] \\
&\quad= \sum_{\alpha\in\{0,1\}^m} \Pr_{s\in_R\{0,1\}^\ell}[f(s) = \alpha] \cdot \Pr_{r,s\in_R\{0,1\}^\ell}[A(r, \alpha) = \langle r, s\rangle | f(s) = \alpha] \\
&\quad\le \sum_{\alpha\in\{0,1\}^m} \frac{|f^{-1}(\alpha)|}{2^\ell} \cdot \left( \Pr_{\substack{r\in_R\{0,1\}^\ell \\ \sigma\in_R\{0,1\}}}[A(r, \alpha) = \sigma] + 2 \cdot \sqrt{\frac{2}{|f^{-1}(\alpha)|}} \right) \\
&\quad\le \frac{1}{2} \cdot \sum_{\alpha\in\{0,1\}^m} \frac{|f^{-1}(\alpha)|}{2^\ell} + \sum_{\alpha\in\{0,1\}^m} \frac{2\sqrt{2|f^{-1}(\alpha)|}}{2^\ell} \\
&\quad\le \frac{1}{2} + 3 \cdot 2^{-\frac{\ell}{2}+m} \ .
\end{aligned}$$

□

### 3.2 Second Construction

We proceed to present the second construction. Recall that $(G, E, D, H)$ is a homomorphic private-key scheme with respect to addition modulo 2 and the polynomial $m(\cdot)$ as in Definition 1.

**Construction 7.** *The public-key encryption scheme $(G'', E'', D'', H'')$ is defined as follows:*

**Key Generation -** $G''(1^n)$ :

*Select $k \leftarrow G(1^n)$, $X = (X_1, \ldots, X_\ell)$ and $Y = (Y_1, \ldots, Y_\ell)$ where $\ell = 10m(n)$, such that $X_i \leftarrow E_k(0)$ and $Y_i \leftarrow E_k(1)$ (with fresh random coins for each $i$).*

*Output $(X, Y)$ as the public-key and $k$ as the private-key.*

**Encryption -** $E''_{X,Y}(\sigma)$ :

*Select at random a subset $S \subseteq [\ell]$ that has size of parity $\sigma$ (i.e. $|S| \equiv \sigma \bmod 2$).*

*Output $H_\oplus(X_{\overline{S}}Y_S)$ (recall that $X_{\overline{S}}Y_S$ is a list of $\ell$ ciphertexts that are encryptions of 1 for coordinates in $S$ and encryptions of 0 elsewhere).*

---

[7] The Leftover Hash Lemma states that if $h$ is selected at random from a universal hash function family from $\{0,1\}^\ell$ to $\{0,1\}^k$ and $t$ is selected uniformly from a set $T \subseteq \{0,1\}^\ell$ then the distribution $h, h(t)$ and the distribution $h, u$, where $u$ is distributed uniformly in $\{0,1\}^k$ are $2\sqrt{\frac{2^k}{|S|}}$-close (see, e.g. [9, Appendix D]).

**Decryption -** $D''_k(c)$ :
*Output* $D_k(c)$.

**Homomorphic Evaluation -** $H''(C,(X,Y),c_1,\ldots,c_\ell)$**:**
*Output* $H(C,c_1,\ldots,c_\ell)$.

As in Construction 3, we first show that the decryption algorithm works and then move on to the main part, showing that the construction is indeed semantically secure.

**Proposition 8.** *For every* $n \in \mathbb{N}$, $\sigma \in \{0,1\}$ *and* $((X,Y),k) \leftarrow G''(1^n)$*:*

$$D''_k\left(E''_{X,Y}(\sigma)\right) = \sigma.$$

*Proof.* Based on the first property of homomorphic encryption (Definition 1),

$$D''_k\left(E''_{X,Y}(\sigma)\right) = D_k\left(H_\oplus\left(X_{\overline{S}}Y_S\right)\right) = \bigoplus_{i=1}^{\ell} D_k(Z_i)$$

where $S$ is the random subset selected in the encryption process, $Z_i = Y_i$ for $i \in S$ and $Z_i = X_i$ otherwise. Since $D$ decrypts correctly, $D_k(X_i) = 0$ and $D_k(Y_i) = 1$. Therefore, $D''_k\left(E''_{X,Y}(\sigma)\right) = \bigoplus_{i\in S} 1 = |S| \bmod 2 = \sigma$. □

We proceed to the main part of the proof, showing that Construction 7 is semantically secure.

**Proposition 9.** *If* $(G,E,D)$ *is a multiple-message semantically secure private-key scheme then* $(G'',E'',D'')$ *is a semantically secure public-key scheme.*

*Proof.* Assume toward a contradiction that $(G'',E'',D'')$ is not semantically secure. This means that there exists a probabilistic polynomial-time adversary $A''$ and a polynomial $p(\cdot)$ such that for infinitely many $n \in \mathbb{N}$:

$$\Pr_{\substack{(X,Y),k\leftarrow G''(1^n)\\ \sigma\in_R\{0,1\}}}\left[A''\left(X,Y,E''_{X,Y}(\sigma)\right)=\sigma\right] > \frac{1}{2}+\frac{1}{p(n)}\ . \tag{8}$$

To derive a contradiction, we consider $n$ from this infinite set and construct a probabilistic polynomial-time adversary $A$ for the underlying private-key scheme. The adversary $A$ receives $2\ell$ ciphertexts $(\alpha_1,\ldots,\alpha_\ell,\beta_1,\ldots,\beta_\ell)$ and will be shown to distinguish between the following two cases:

- $\alpha_1,\ldots,\alpha_\ell$ are encryptions of 0 and $\beta_1,\ldots,\beta_\ell$ are encryptions of 1.
- $\alpha_1,\ldots,\alpha_\ell,\beta_1,\ldots,\beta_\ell$ are encryptions of 0.

Algorithm $A$ operates as follows:

1. Set $X = (\alpha_1,\ldots,\alpha_\ell)$ and $Y = (\beta_1,\ldots,\beta_\ell)$.
2. Select $S \subseteq_R [\ell]$.
3. Output 1 if $A''(X,Y,H_\oplus(X_{\overline{S}}Y_S)) = |S| \bmod 2$ and 0 otherwise.

Accordingly,

$$\Pr_{\substack{k\leftarrow G(1^n)\\ \alpha_j,\beta_j}}\left[A\left(\alpha_1,\ldots,\alpha_\ell,\beta_1,\ldots,\beta_\ell\right)=1\right]=\Pr_{\substack{k\leftarrow G(1^n)\\ X,Y,S}}\left[A''\left(X,Y,H_\oplus\left(X_{\overline{S}}Y_S\right)\right)=|S| \bmod 2\right].$$

We proceed by analyzing $A$'s behavior in the two different cases. In the first case, $\alpha_i = E_k(0)$ and $\beta_i = E_k(1)$. Consequently, $H_\oplus(X_{\overline{S}}Y_S)$ is distributed identically to an encryption of a random bit under $E''$ and so, by Eq. (8), it holds that

$$\begin{aligned}\Pr_{\substack{k\leftarrow G(1^n)\\ X,Y,S}}\left[A''\left(X,Y,H_\oplus\left(X_{\overline{S}}Y_S\right)\right)=|S| \bmod 2\right] &= \Pr_{\substack{(X,Y),k\leftarrow G''(1^n)\\ \sigma\in_R\{0,1\}}}\left[A''\left(X,Y,E''_{X,Y}\left(\sigma\right)\right)=\sigma\right]\\ &> \frac{1}{2}+\frac{1}{p(n)}\ .\end{aligned}$$

In the second case, $\alpha_i = \beta_i = E_k(0)$. We argue that in this case for every $n \in \mathbb{N}$ and even for an unbounded adversary $A'$,

$$\Pr_{\substack{k\leftarrow G(1^n)\\ X,Y,S}}\left[A''\left(X,Y,H_\oplus\left(X_{\overline{S}},Y_S\right)\right)=|S| \bmod 2\right]<\frac{1}{2}+2^{-0.2\ell+m(n)+1}. \quad (9)$$

Equation (9) follows from an information-theoretic theorem (Theorem 10) that will be stated next. See the full version of this paper [14] for the proof of Theorem 10.

Using Theorem 10, we conclude that $A$ distinguishes between the two cases with non-negligible probability, in contradiction to the multiple-message security of $(G, E, D)$, □

**Information-Theoretic Theorem.** Let $\Omega$ be a finite non-empty set and $\ell \in \mathbb{N}$. Let $\mu_1, \ldots, \mu_\ell$ be distributions over $\Omega$ and $\mu = \mu_1 \times \cdots \times \mu_\ell$ be a product distribution over $\Omega^\ell$. Let $X$ and $Y$ be independent random variables identically distributed according to $\mu$ over $\Omega^\ell$.

**Theorem 10.** *For any $\ell, m \in \mathbb{N}$ and any functions $h\colon \Omega^\ell \to \{0,1\}^m$ and $g\colon \Omega^\ell \times \Omega^\ell \times \{0,1\}^m \to \{0,1\}$, it holds that*

$$\Pr_{X,Y,S\subseteq_R[\ell]}\left[g\left(X,Y,h(X_{\overline{S}}Y_S)\right)=|S| \bmod 2\right]<\frac{1}{2}+2^{-0.2\ell+m+1}.$$

Equation (9) seems to follow immediately from Theorem 10 by setting $A''$ as $g$, $H_\oplus$ as $h$ and having $X$ and $Y$ distributed as $\ell$ independent encryptions of 0 each. However, there is a small subtlety - Theorem 10 addresses $g$ and $h$ that are *deterministic* functions, in contrast to $A''$ and $H$ that are *probabilistic* algorithms. Additionally, since $X$ and $Y$ are distributed w.r.t to the same randomly chosen key, they are not product distributions as required by Theorem 10.

Both issues are resolved by an averaging argument. If Eq. (9) does not hold for some $n \in \mathbb{N}$, then there exist random coins for $A''$, $H$ and a fixed private-key $k$ for

which it does not hold. Once we fix these coins, $A''$ and $H$ become deterministic functions. Additionally, we set $X$ and $Y$ to each be distributed as $\ell$ encryptions of 0 under the fixed key $k$, which is in particular a product distribution. Thus, the hypothesis that Eq. (9) does not hold contradicts Theorem 10. The proof of Theorem 10 uses the Efron-Stein decomposition [4], an extension of Fourier analysis for general product distributions and appears in the full version of this paper [14] (see an outline next).

**Outline of the Proof of Theorem 10.** Theorem 10 considers a game in which a computationally unbounded adversary sees $X$, $Y$ and $m$ bits of information on $X_{\overline{S}}Y_S$ and needs to decide whether $S$ is of even or odd cardinality. That is, the adversary specifies a function $h : \Omega^\ell \to \{0,1\}^m$ and based on $X, Y, h(X_{\overline{S}}Y_S)$ needs to find $|S|$ mod 2. Theorem 10 states that winning this game with probability noticeably better than $\frac{1}{2}$ is impossible as long as $m$ is sufficiently smaller than $\ell$. Note that winning the game becomes easy if $m$ is sufficiently larger[8] than $\ell$ (as long as the probability of a collision in each coordinate, i.e. $\Pr[X_i = Y_i]$, is sufficiently small).

To prove Theorem 10, we would like to that show that for a *typical* $\gamma \in \{0,1\}^m$, the number of odd $S$ that map to $\gamma$ (i.e., $h(X_{\overline{S}}Y_S) = \gamma$) and the number of even $S$ are roughly the same. This would imply that any adversary, which sees only $X$, $Y$ and $\gamma$, cannot guess whether $\gamma$ was produced from an odd or even $S$, which is exactly what we are looking to prove.

The proof is composed of two lemmas. The main lemma states that for *every* $\gamma \in \{0,1\}^m$, w.h.p, the number of odd $S$ that map to $\gamma$ is fairly close to the number of even $S$ (in absolute terms). We prove this lemma by showing that the probability of a collision of two random sets $S$ and $T$ of the same parity (i.e. the probability that $h(X_{\overline{S}}Y_S) = h(X_{\overline{T}}Y_T)$ where $|T| = |S|$) is roughly the same as the collision probability of two sets of different parity. We use the Efron-Stein decomposition to express the collision probability and bound it. The second lemma is more straightforward and states that for a *typical* $\gamma$ the total number of $S$ that map to it is very large. Combining these two lemmas we prove Theorem 10. See the full version of the paper for details.

## 4 Homomorphic Properties of the Public-Key Scheme

In this section, we discuss the homomorphic properties of the public-key schemes presented in Section 3. We show that if the underlying private-key scheme supports $i + 1$ repeated homomorphic operations then both Construction 3 and Construction 7 support $i$ such operations. Intuitively, this follows by the fact that in both constructions, the encryption algorithm applies a single homomorphic operation (see Fact 12), thus exactly one hop is lost.

**Proposition 11.** *Suppose $G, E, D, H$ are an $(i+1)$-hop homomorphic private-key scheme w.r.t to a set of circuit families $\mathcal{C}$ that includes addition modulo 2.*

[8] If $m \geq \ell \log(|\Omega|)$ just take $h$ to be the identity function.

*Then Constructions 3 and Construction 7 are $i$-hop homomorphic w.r.t the set $\mathcal{C}$.*

We prove that Proposition 11 holds for Construction 3 while noting that the proof for Construction 7 is completely analogous. Thus, we refer to $G', E', D', H'$ as in Construction 3.

Let $(X, r), k$ be a pair of encryption/decryption keys for Construction 3 (w.r.t to the security parameter $n$). We denote the $j$-th level ciphertexts of the private-key scheme by $W_j(k)$ and the $j$-th level ciphertexts of the public-key scheme by $W'_j(X, r)$.

**Fact 12.** *For every $j \in \mathbb{N}$, $W'_j(X, r) \subseteq W_{j+1}(k)$.*

*Proof.* By induction on $j$.

Let $\{C_k\}_k \in \mathcal{C}$, $0 \leq j \leq i$, $\ell = \ell(n)$ and $w_1, \ldots, w_\ell$ be $j$-th level ciphertexts of the public-key scheme (i.e., in $W'_j(X, Y)$). We proceed by showing that it $j$-th level ciphertexts decrypt properly. By Fact 12, it holds that $w_1, \ldots, w_\ell \in W_{j+1}(k)$ and thus,

$$\begin{aligned} H'(C_\ell, (X, r), w_1, \ldots, w_\ell) &= H(C_\ell, w_1, \ldots, w_\ell) \\ &= C_\ell(D_k(w_1), \ldots, D_k(w_\ell)) \\ &= C_\ell(D'_k(w_1), \ldots, D'_k(w_\ell)). \end{aligned}$$

where the first and third equalities follow from the definition of $H'$ and $D'$ respectively and the second equality follows from the fact that $(G, E, D, H)$ are $i+1$-hop homomorphic and that $w_1, \ldots, w_\ell$ are ciphertexts of level $j+1 \leq i+1$ of the private-key scheme.

A similar argument shows that the scheme is also compact. Indeed, since $w_1, \ldots, w_\ell \in W'_j(X, Y) \subseteq W_{j+1}(k)$ it holds that,

$$|H'(C_\ell, (X, r), w_1, \ldots, w_\ell)| = |H(C_\ell, w_1, \ldots, w_\ell)| \leq m(n)$$

for every $0 \leq j \leq i$.

## Acknowledgments

I would like to express my thanks and appreciation to my M.Sc. advisor, Oded Goldreich, for his encouragement and guidance in completing this work. In particular, I would like to thank him for many helpful discussions and constructive comments that helped present this work in a more coherent way.

I would like to thank Benny Applebaum for useful conversations. I am also extremely grateful to Yuval Ishai for suggesting Construction 3 and for allowing me to include it in this paper.

## References


1. Barak, B.: Cryptography course - Lecture notes, COS 433. Princeton University, Computer Science Department (Spring 2010), available at http://www.cs.princeton.edu/courses/archive/spring10/cos433
2. Di Crescenzo, G., Malkin, T., Ostrovsky, R.: Single database private information retrieval implies oblivious transfer. In: EUROCRYPT: Advances in Cryptology: Proceedings of EUROCRYPT (2000)
3. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 24–43. Springer (2010), http://dx.doi.org/10.1007/978-3-642-13190-5
4. Efron, B., Stein, C.: The jackknife estimate of variance. The Annals of Statistics 9(3), 586–596 (1981), http://www.jstor.org/stable/2240822
5. Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: CRYPTO. pp. 10–18 (1984), http://dx.doi.org/10.1007/3-540-39568-7_2
6. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009. pp. 169–178. ACM (2009), http://doi.acm.org/10.1145/1536414.1536440
7. Gentry, C., Halevi, S., Vaikuntanathan, V.: $i$-hop homomorphic encryption and rerandomizable yao circuits. In: Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference. pp. 155–172. Springer (2010), http://dx.doi.org/10.1007/978-3-642-14623-7
8. Goldreich, O.: Foundations of Cryptography: Volume 2: Basic Applications. Cambridge University Press (2004)
9. Goldreich, O.: Computational complexity: a conceptual perspective. SIGACT News 39(3), 35–39 (2008)
10. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences 28(2), 270–299 (1984)
11. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing 28, 12–24 (1999)
12. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: single database, computationally-private information retrieval. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science, FOCS 1997. p. 364. IEEE Computer Society (1997)
13. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation. pp. 169–180. Academic Press (1978)
14. Rothblum, R.: Homomorphic encryption: from private-key to public-key. Electronic Colloquium on Computational Complexity (ECCC) 17, 146 (2010), http://eccc.hpi-web.de/report/2010/146