Computing Blindfolded: New Developments in Fully Homomorphic Encryption

Vinod Vaikuntanathan
University of Toronto

Abstract— A fully homomorphic encryption scheme enables computation of arbitrary functions on encrypted data. Fully homomorphic encryption has long been regarded as cryptography's prized "holy grail" – extremely useful yet rather elusive. Starting with the groundbreaking work of Gentry in 2009, the last three years have witnessed numerous constructions of fully homomorphic encryption involving novel mathematical techniques, and a number of exciting applications. We will take the reader through a journey of these developments and provide a glimpse of the exciting research directions that lie ahead.

1. Introduction

Encryption has traditionally been viewed as a mechanism that enables *secure communication*, namely the problem of transmitting a message from Alice to Bob over a public channel while keeping it hidden from an eavesdropper. In particular, Public-key Encryption – conceived in the seminal work of Diffie and Hellman [24] and first constructed by Rivest, Shamir and Adleman [68] – provides a way for Alice to encrypt a message into a ciphertext using Bob's public key, and for Bob to decrypt the ciphertext to obtain the message using his secret key. In this view of encryption schemes, *access to encrypted data is all or nothing* – having the secret decryption key enables one to learn the entire message, but without the decryption key, the ciphertext is completely useless.

This state of affairs raises an intriguing question, first posed by Rivest, Adleman and Dertouzos in 1978: *Can we do arbitrary computations on data while it remains encrypted, without ever decrypting it?* This asks for the seemingly fantastical ability to perform computations on encrypted data without being able to "see" the data. Such ability also gives rise to a number of useful applications including the ability to privately outsource arbitrary computations to the "cloud" and the ability to store all data encrypted and perform computations on encrypted data, decrypting only when necessary.

Fully Homomorphic encryption is a special type of encryption system that permits *arbitrarily complex computation* on encrypted data. Long regarded as a "holy grail" of cryptography, fully homomorphic encryption was first shown to be possible in the recent, breakthrough work of Gentry. We will take the reader through a journey of the fascinating mathematical techniques underlying these developments, which in turn raise a number of exciting new questions in cryptography.

Organization of this Survey. Starting with a brief history, we go on to formally define homomorphic encryption and its various useful properties, and then describe the ideas behind Gentry's construction. We then describe the recent works in this area that significantly differ from Gentry's blueprint and result in simpler constructions, better efficiency and better assumptions – all in one. We conclude with a discussion of the applications of fully homomorphic encryption and a (highly incomplete) list of research directions. Our intention in this short survey is to give the reader a taste of the main ideas and developments in this area, while referring to the original works for detailed expositions.

2. The History of Homomorphic Encryption

The notion of encryption schemes that permit non-trivial computation on encrypted data was first proposed by Rivest, Adleman and Dertouzos [67] in 1978, in a remarkably foresightful paper titled "On Data Banks and Privacy Homomorphisms". Rivest et al. [67] proposed the exponentiation function and the RSA function as additive and multiplicative privacy homomorphisms, respectively. Note, however, that neither of these functions by themselves provide even chosen plaintext security.[1]

Email: vinodv@cs.toronto.edu. Supported by an NSERC Discovery Grant and by DARPA under Agreement number FA8750-11-2-0225. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

[1] The El Gamal encryption scheme, derived from the exponentiation function, is multiplicatively homomorphic and CPA-secure. However, methods of turning the RSA function into a CPA-secure encryption scheme, either by the hardcore-bit construction [42] or the RSA-OAEP construction [8], seem to destroy its homomorphic properties.

The first semantically secure homomorphic encryption scheme follows from the work of Goldwasser and Micali [42] that defined the first robust notion of security for encryption. The GM encryption scheme supports addition of encrypted bits mod 2 (that is, the exclusive-OR function). A number of encryption systems that are either additively or multiplicatively homomorphic followed suit. This includes the El Gamal encryption scheme [29], the Paillier encryption scheme [59] and its generalization by Dåmgard and Jurik [23], a host of lattice-based encryption schemes starting from the work of Ajtai and Dwork [4], [65], [66], and many others [20], [55], [57]. All these schemes support either homomorphic addition or multiplication of plaintexts, but not both! Constructing an encryption scheme that is both additively and multiplicatively homomorphic remained a tantalizing open question. Clearly, since addition and multiplication (say, over $\mathbb{Z}_2$) form a complete set of operations, such a scheme enables performing any polynomial-time computation on encrypted data!

Additively homomorphic encryption schemes are already quite useful in a number of applications. We will not attempt to make an exhaustive list here, but we mention three such applications. Cohen and Fischer [20] proposed an additively homomorphic encryption scheme based on higher order residuosity, and showed how to use it to perform secure electronic voting. This proposal and its descendants have made its way to modern day web-based voting systems such as *Helios* [1]. Quite recently, Peikert and Waters [62] constructed lossy and all-but-one trapdoor functions from additively homomorphic encryption schemes (with some extra properties), which they in turn use to construct chosen ciphertext secure (CCA-secure) public-key encryption schemes. The third application is to Private Information Retrieval (PIR) protocols, which we postpone to the end of this section.

It took a long time for us to construct encryption schemes that go beyond simple additive (or multiplicative) homomorphisms. Boneh, Goh and Nissim [11] showed an encryption scheme based on bilinear pairings on elliptic curves that could perform arbitarily many additions as well as a single multiplication on plaintexts (Gentry, Halevi and Vaikuntanathan [38] later showed how to achieve the same ability using lattices). The work of Sander, Young and Yung [72] constructed a scheme that can evaluate NC^1 circuits (the ciphertexts in their scheme grow exponentially with the depth of the circuit, and hence the restriction to NC^1), and a scheme of Aguilar-Melchor, Gaborit and Herranz [53] homomorphically evaluated (multi-variate) polynomials (where the ciphertext grows exponentially in the degree of the polynomial). A proposal for fully homomorphic encryption by Fellows and Koblitz [28] based on the hardness of the Ideal Membership Problem in the multivariate polynomial ring $\mathbb{F}_q[x_1, \ldots, x_n]$ suffered from attacks for many of their parameter choices, although the security of their general scheme still seems open.

An important application of homomorphic encryption comes from the work of Kushilevitz and Ostrovsky [49] who showed how to construct (single-server) Private Information Retrieval (PIR) protocols with sub-linear communication, from *any* additively homomorphic encryption scheme. Private Information Retrieval, defined by Chor, Kushilevitz, Goldreich and Sudan [18], is the problem wherein a user attempts to retrieve the i^{th} item in a database of size N, revealing no information about the index i to the database owner (the basic version of the problem does not concern itself with the complementary problem of protecting the privacy of the database owner). Of course, since this can be trivially achieved by the database owner sending over the entire database to the user, the non-trivial question is: can we do better in terms of the communication complexity? For a database of size N, the Kushilevitz-Ostrovsky protocol requires communicating only N^ϵ bits, for an arbitrarily small constant $\epsilon > 0$.

Private Information Retrieval is intimately connected to Fully Homomorphic Encryption. In fact, one can easily see how to construct *inefficient* fully homomorphic encryption schemes starting from any PIR scheme. The construction proceeds by simply writing out the truth-table of the function (say, with n-bit inputs and a single bit output) as a database. The ciphertext of a message $m \in \{0,1\}^n$ is computed as the query of the PIR user for the m^{th} entry of the truth table, and homomorphic evaluation is simply running the PIR protocol to retrieve the m^{th} entry – namely, $f(m)$. Since the query of the PIR user hides m, the scheme is semantically secure. Furthermore, the length of the ciphertext after homomorphic evaluation is exactly the communication complexity of the PIR protocol which is sub-linear, or even logarithmic [16], [39], [32], [15]. The catch is that for a function with n inputs, homomorphic evaluation requires $2^{O(n)}$ time. More interestingly, Ishai and Paskin [46] showed how to use specific, very efficient *additively* homomorphic encryption schemes to construct an encryption scheme capable of evaluating *branching programs* homomorphically. We believe that these connections between PIR and FHE schemes is far from coincidental, and a deeper exploration of this connection is sure to turn up invaluable treasures.

To summarize the state of affairs pre-2009, we knew encryption schemes that could perform additions and a single multiplication [11], [38] and schemes that could evaluate Branching programs [46], polynomials [53] and NC^1 circuits with an expansion in the size of the ciphertext [72].

The big breakthrough came with the work of Gentry [32] in 2009, which showed the first plausible construction of a fully homomorphic encryption scheme that enables computation of arbitrary functions on encrypted data and producing compact ciphertexts.

3. Homomorphic Encryption: Definitions and Properties

Before describing Gentry's construction and subsequent work on fully homomorphic encryption, we first formalize the notion of homomorphic encryption, describe its various properties and establish notation for later discussions.

3.1. Homomorphic Encryption – Definitions

Throughout this section (and this survey) we use κ to indicate the security parameter. In addition, all schemes in this paper encrypt bit-by-bit and therefore our definitions only refer to this case. The generalization to an arbitrary message space is immediate.

A homomorphic (public-key) encryption scheme $\mathsf{HE} = (\mathsf{HE.Keygen}, \mathsf{HE.Enc}, \mathsf{HE.Dec}, \mathsf{HE.Eval})$ is a quadruple of probabilistic polynomial-time (PPT) algorithms as follows.

- **Key generation.** The algorithm
$$(pk, evk, sk) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$$
takes a unary representation of the security parameter and outputs a public encryption key pk, a public evaluation key evk and a secret decryption key sk.
- **Encryption.** The algorithm $c \leftarrow \mathsf{HE.Enc}_{pk}(\mu)$ takes the public key pk and a single bit message $\mu \in \{0,1\}$ and outputs a ciphertext c.
- **Decryption.** The algorithm $\mu^* \leftarrow \mathsf{HE.Dec}_{sk}(c)$ takes the secret key sk and a ciphertext c and outputs a message $\mu^* \in \{0,1\}$.
- **Homomorphic evaluation.** The algorithm
$$c_f \leftarrow \mathsf{HE.Eval}_{evk}(f, c_1, \ldots, c_\ell)$$
takes the evaluation key evk, a function $f : \{0,1\}^\ell \to \{0,1\}$ and a set of ℓ ciphertexts $c_1, \ldots, c_\ell$, and outputs a ciphertext c_f.
The representation of the function f is an important issue. Since the representation can vary between schemes, we leave this issue outside of this syntactic definition. We remark, however, that in this survey, we will represent f by an arithmetic circuit over $\mathrm{GF}(2)$ (equivalently, a Boolean circuit with AND and XOR gates).

We note that while one can treat the evaluation key as a part of the public key, as has been done in some of the literature on fully homomorphic encryption, we feel that there is an expository value to treating it as a separate entity and to distinguishing between the public elements that are used for encryption (namely, the public key pk) and those that are used only for homomorphic evaluation (namely, the evaluation key evk).

The only security notion we consider in this survey is semantic security, namely security w.r.t. passive adversaries. We use its widely known formulation as IND-CPA security, defined as follows.

Definition 3.1 (CPA security)**.** *A scheme* HE *is IND-CPA secure if for any polynomial time adversary* $\mathcal{A}$ *it holds that*

$$\mathrm{Adv}_{\mathrm{CPA}}[\mathcal{A}] \triangleq \Big| \Pr[\mathcal{A}(pk, evk, \mathsf{HE.Enc}_{pk}(0)) = 1] - \Pr[\mathcal{A}(pk, evk, \mathsf{HE.Enc}_{pk}(1)) = 1] \Big| = \mathrm{negl}(\kappa) \ ,$$

where $(pk, evk, sk) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$.

We move on to define the correctness of homomorphic evaluation. Note that we do not define the "correctness" of the scheme per se, but rather (some form of) correctness will follow from the correctness of homomorphic evaluation.

We start by defining $\mathcal{C}$-homomorphism, which is homomorphism with respect to a specified class $\mathcal{C}$ of functions. This notion is sometimes also referred to as "somewhat homomorphism".

Definition 3.2 ($\mathcal{C}$-homomorphism)**.** *Let* $\mathcal{C} = \{\mathcal{C}_\kappa\}_{\kappa \in \mathbb{N}}$ *be a class of functions (together with their respective representations). A scheme* HE *is* $\mathcal{C}$*-homomorphic (or, homomorphic for the class* $\mathcal{C}$*) if for any sequence of functions* $f_\kappa \in \mathcal{C}_\kappa$ *with* $f_\kappa : \{0,1\}^{\ell(\kappa)} \to \{0,1\}$*, and respective inputs* $\mu_1, \ldots, \mu_\ell \in \{0,1\}$*, it holds that*

$$\Pr[\mathsf{HE.Dec}_{sk}(\mathsf{HE.Eval}_{evk}(f, c_1, \ldots, c_\ell)) \neq f(\mu_1, \ldots, \mu_\ell)] = \mathrm{negl}(\kappa) \ ,$$

where negl *is a negligible function,* $(pk, evk, sk) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$ *and* $c_i \leftarrow \mathsf{HE.Enc}_{pk}(\mu_i)$.

Note that the above syntactic definition by itself permits rather uninteresting homomorphic encryption schemes. In particular, consider a homomorphic evaluation algorithm that acts as the identity function, outputting the ciphertexts $c_1, \ldots, c_\ell$ as well as a description

of the function f. The decryption recovers the messages $\mu_1, \ldots, \mu_\ell$ and outputs $f(\mu_1, \ldots, \mu_\ell)$.

The key property of homomorphic encryption schemes that prevents such uninteresting (and useless!) constructions is the notion of *compactness*, defined below.

Definition 3.3 (compactness)**.** *A homomorphic scheme* HE *is compact if there exists a polynomial* $s = s(\kappa)$ *such that the output length of* $\mathsf{HE.Eval}(\cdots)$ *is at most* s *bits long (regardless of* f *or the number of inputs).*

In other words, compactness requires that the size of the ciphertext after homomorphic evaluation does not depend on either the number of inputs ℓ or the complexity of the function f, but only on the size of the output of f. Note that a $\mathcal{C}$-homomorphic scheme (as defined above) is not necessarily compact.

We give the minimal definition of fully homomorphic encryption, which suffices for most applications.

Definition 3.4 (fully homomorphic encryption)**.** *A scheme* HE *is fully homomorphic if it is both compact and homomorphic for the class of all arithmetic circuits over* $GF(2)$.

An important relaxation of fully homomorphic encryption is the following.

Definition 3.5 (leveled fully homomorphic encryption)**.** *A leveled fully homomorphic encryption scheme is a homomorphic scheme where* $\mathsf{HE.Keygen}$ *gets an additional input* 1^L *(now* $(pk, evk, sk) \leftarrow \mathsf{HE.Keygen}(1^\kappa, 1^L)$*) and the resulting scheme is homomorphic for all depth-*L *binary arithmetic circuits. The bound* $s(\kappa)$ *on the ciphertext length must remain independent of* L.

In most cases, the only parameter of the scheme that becomes dependent on L is the bit-length of the evaluation key evk.

Two Useful Properties: We describe two useful properties of homomorphic encryption schemes, namely *circuit privacy* and *multi-hop homomorphism*.

In the scenario of outsourcing computation (described in Section 1), using homomorphic encryption protects the privacy of the client but *not the privacy of the server*. *Circuit privacy* is a property of homomorphic encryption that guarantees that the server's input – namely, the function f – remains private from the client. In particular, circuit privacy requires that the output of $\mathsf{HE.Eval}(f, c_1, \ldots, c_\ell)$ does not reveal any information about f to the client, beyond the output $f(\mu_1, \ldots, \mu_\ell)$. Note that the client knows the secret key sk, as well as the randomness used to generate the ciphertexts c_i. This is formalized by a simulation-based definition which requires that the output of $\mathsf{FHE.Eval}$ can be simulated using only $f(\mu_1, \ldots, \mu_\ell)$ (but no other information about f). A number of variants of circuit privacy have been explored – to deal with malicious or semi-honest clients, and to relax the requirement so that the client is permitted to learn some limited information about f, e.g., the size of the circuit computing f, but nothing else. For detailed discussions of circuit privacy, we refer the reader to [46], [37].

In some applications, it is useful to require that the output of $\mathsf{HE.Eval}$ can be used as an input for another homomorphic evaluation. A homomorphic encryption scheme with this property is called a "multi-hop homomorphic" encryption scheme, a notion introduced in the work of Gentry, Halevi and Vaikuntanathan [37]. For a detailed discussion and applications of multi-hop homomorphism, we refer the reader to [37].

The schemes we present in this survey are circuit private (or, can easily be made to be so) as well as multi-hop homomorphic, but we do not describe these properties further in this survey.

4. The First FHE Schemes: Gentry's Blue-print and Instantiations

Gentry's work showed not only the first fully homomorphic encryption scheme, but a general method (a "blue-print") to construct such systems. This blue-print has been instantiated with a number of cryptographic assumptions, yielding progressively simpler and more efficient schemes [25], [73], [75], [14]. In this section, we describe the ideas behind Gentry's original construction, and illustrate the method with the scheme of van Dijk, Gentry, Halevi and Vaikuntanathan [25].

Notwithstanding the elegance and generality of the blue-print, schemes constructed along these lines suffer from a number of deficiencies, including the reliance on a host of non-standard cryptographic assumptions, and severe limitations on efficiency. In Section 5, we describe new developments in fully homomorphic encryption that solve some of these issues.

Gentry's construction has three components: a "somewhat homomorphic" encryption scheme that can evaluate a limited class of functions, a method of "bootstrapping" a sufficiently powerful homomorphic encryption scheme – called a "bootstrappable" encryption scheme – into a fully homomorphic encryption scheme and finally, to connect the dots, a specialized method of turning the somewhat homomorphic scheme into a bootstrappable scheme.

4.1. Step 1: Somewhat Homomorphic Encryption

The first step in Gentry's blueprint is to construct a *somewhat homomorphic encryption (SWHE) scheme*, namely an encryption scheme capable of evaluating "low-degree" polynomials homomorphically.[2] We describe below the scheme of van Dijk et al. [25], the simplest for the purposes of exposition.

The DGHV Construction. The construction is based on the hardness of the approximate greatest common divisors (approx-GCD) problem, formulated and studied by Howgrave-Graham [45].

In the approx-GCD problem, we are given polynomially many numbers of the form $pq_i + r_i$, where

- p is a random odd number chosen from an interval $[0, 2^\eta]$,
- q_i are chosen from a distribution $D = D_p$ which can potentially depend on p, and
- r_i are ("small") noise terms chosen from the interval $[-2^\rho, 2^\rho]$.

Here, η, ρ and the distribution D are parameters of the problem. Our goal is to find the hidden number p. Note that if the noise terms are absent (namely, if $r_i = 0$), we can find p from just two samples by computing their GCD. Essentially, then, the assumption says that when the multiples are "noisy", it is hard to extract the hidden common divisor.

This problem was considered in [45], [25] and more recently by Chen and Nguyen [17] and Cohn and Heninger [21]. The latter two papers show improved algorithms for the approx-GCD problem, nevertheless, the general setting of the problem still remains hard.

We first show how to construct a secret-key somewhat homomorphic encryption scheme. A recent result of Ron Rothblum [70] shows how to convert any secret-key homomorphic encryption scheme (that supports a certain minimal set of operations) into a public-key somewhat homomorphic scheme. Invoking this result, we obtain a public-key somewhat homomorphic scheme.

Our construction uses a number of parameters, borrowed from the approximate GCD problem, which are set as follows:

- ρ We will set ρ to be a sufficiently large polynomial in the security parameter, to thwart the attacks of [17], [21];
- η We will set $\eta \geq \rho \cdot \Theta(\kappa \log^2 \kappa)$, in order to support homomorphism for "deep enough" circuits;
- γ We will set $\gamma = \omega(\eta^2 \log \kappa)$, to thwart various lattice-based attacks on the underlying approx-GCD problem; and the distribution $D = D_{\gamma,p}$ to be the following:

$$\mathcal{D}_{\gamma,p} = \left\{ \mathsf{choose}\ q \xleftarrow{\$} \mathbb{Z} \cap [0,\ 2^\gamma/p) \right\}$$

The encryption scheme works as follows.

- $\mathsf{SH.Keygen}(\kappa)$: The secret key is an odd η-bit integer: $p \xleftarrow{\$} (2\mathbb{Z}+1) \cap [2^{\eta-1}, 2^\eta)$.
  For each $0 \leq i \leq \gamma$: Sample uniformly random numbers q_i from the distribution $D_{\gamma+i,p}$ until $q_i \geq 2^{\gamma+i-1}/p$. Sample $r_0, \ldots, r_\gamma$ to be random integers in the interval $(-2^\rho, 2^\rho)$ and output as the evaluation key

$$evk = \{\ x^{(i)} \leftarrow pq^{(i)} + r^{(i)}\ \}_{0 \leq i \leq \gamma}$$

- $\mathsf{SH.Enc}(sk, \mu \in \{0,1\})$: Sample a uniformly random $q \leftarrow D_{\gamma,p}$, and $r \leftarrow \mathbb{Z} \cap (-2^\rho, 2^\rho)$. Output

$$\mathsf{CT} \leftarrow pq + 2r + \mu$$

- $\mathsf{SH.Eval}(pk, C, \mathsf{CT}_1, \ldots, \mathsf{CT}_t)$: Given the (binary) circuit $\mathcal{C}$ with t inputs (and AND and XOR gates), construct an arithmetic circuit $\bar{\mathcal{C}}$ over the integers where each of the AND and XOR gates are replaced with integer multiplication and addition gates, respectively.
  Given t ciphertexts CT_i, apply the arithmetic circuit to the ciphertexts, performing all the operations over the integers, with the following modifications:
  - For an addition gate with inputs CT_1 and CT_2, output $\mathsf{CT}_{\mathsf{add}} := \mathsf{CT}_1 + \mathsf{CT}_2 \pmod{x^{(0)}}$.
  - For a multiplication gate with inputs CT_1 and CT_2, to compute the product $\mathsf{CT}_{\mathsf{mlt}}^{(\gamma+1)} := \mathsf{CT}_1\mathsf{CT}_2$ over the integers, iteratively compute

$$\mathsf{CT}_{\mathsf{mlt}}^{(i)} := \mathsf{CT}_{\mathsf{mlt}}^{(i+1)} \pmod{x^{(i)}}$$

    from $i = \gamma$ downto 0, and output $\mathsf{CT}_{\mathsf{mlt}}^{(0)}$.

- $\mathsf{SH.Dec}(sk, \mathsf{CT})$: Output

$$\mu' \leftarrow (\mathsf{CT} \bmod p) \bmod 2$$

For a fresh ciphertext $\mathsf{CT} \leftarrow pq + 2r + \mu$, the decryption algorithm computes

$$(\mathsf{CT} \bmod p) \bmod 2 = (2r + \mu) \bmod 2 = \mu$$

as long as the "noise" $2r + \mu$ is "small enough". More precisely, decryption works as long as $2r + \mu < p/2$.

[2] We use the term "low-degree" polynomials rather loosely in this section to mean ℓ-variate polynomials (for some $\ell = \ell(\kappa) = \mathsf{poly}(\kappa)$) where each monomial has degree at most κ^ϵ (for some constant $\epsilon < 1$). For a more formal statement, see Theorem 4.1.

As for the homomorphic operations, note that homomorphic addition computes

$$\mathsf{CT}_1 + \mathsf{CT}_2 = p(q_1 + q_2) + 2(r_1 + r_2) + (\mu_1 + \mu_2)$$

which is an valid ciphertext of $\mu_1 + \mu_2 \pmod 2$, but with twice as much noise as before. Homomorphic multiplication computes

$$\mathsf{CT}_1\mathsf{CT}_2 = p(q_1\mathsf{CT}_2 + q_2\mathsf{CT}_1 - pq_1q_2) + 2(2r_1r_2 + r_1\mu_2 + r_2\mu_1) + \mu_1\mu_2$$

which is a valid ciphertext of $\mu_1\mu_2$, but with quadratically as much noise as before. Taking the result modulo the $x^{(i)}$ ensures that the size of the ciphertext is at most γ bits, without increasing the noise by much. Extending this to homomorphic evaluation of general functions, one can show:

Theorem 4.1. *Assuming the approximate GCD assumption with parameters ρ, η and γ (which define the distribution $D_{\gamma,p}$ as above), the scheme is semantically secure (CPA-secure). Furthermore, the scheme is capable of evaluating any degree D polynomial with $\ell = \mathsf{poly}(\kappa)$ variables as long as $D < C\eta/\rho$ for some universal constant C.*

Note that reducing the security of the scheme to approximate GCD is non-trivial since approximate GCD is a "search assumption" whereas an adversary breaking the semantic security of the scheme provides a "decisional oracle". For more details on the proof, see [25].

4.2. Step 2: The Bootstrapping Theorem

The somewhat homomorphic encryption (SWHE) scheme is only able to evaluate "low-degree" polynomials, falling well short of the eventual goal of fully homomorphic encryption (FHE). To obtain FHE, Gentry provided a remarkable *bootstrapping theorem* which states that given an SWHE scheme that can evaluate its own decryption function (plus an additional operation), one can transform it into a "leveled" FHE scheme, in a completely generic way. Such an SWHE scheme is called a bootstrappable encryption scheme. Furthermore, if we are willing to make an additional assumption – namely that it is safe to encrypt the leveled FHE secret key under its own public key, a requirement that is referred to as "circular security" – then the transformation gives us a "pure" (as opposed to "leveled") FHE scheme. Bootstrapping "refreshes" a ciphertext by running the decryption function on it homomorphically, using an encrypted secret key (given in the evaluation key), resulting in a reduced noise.

We formally define the notion of a bootstrappable encryption scheme and present Gentry's bootstrapping theorem [32], [31] which implies that a bootstrappable scheme can be converted into a fully homomorphic one.

Definition 4.1 (bootstrappable encryption scheme)**.** *Let HE be $\mathcal{C}$-homomorphic, and Let f_{add} and f_{mult} be the the augmented decryption functions of the scheme defined as $f^{c_1,c_2}_{\mathsf{add}}(s) = \mathsf{HE.Dec}_s(c_1)$ XOR $\mathsf{HE.Dec}_s(c_2)$ and $f^{c_1,c_2}_{\mathsf{mult}}(s) = \mathsf{HE.Dec}_s(c_1)$ AND $\mathsf{HE.Dec}_s(c_2)$. Then $\mathcal{E}$ is bootstrappable if*

$$\left\{ f^{c_1,c_2}_{\mathsf{add}}, f^{c_1,c_2}_{\mathsf{mult}} \right\}_{c_1,c_2} \subseteq \mathcal{C} \ .$$

Namely, the scheme can homomorphically evaluate f_{add} and f_{mult}.

We describe two variants of Gentry's bootstrapping theorem. The first implies leveled fully homomorphic encryption but requires no additional assumption; where the second makes an additional *(weak) circular security* assumption and achieves the stronger (non-leveled) variant (namely, Definition 3.4).

The first variant follows.

Theorem 4.2 ([32], [31])**.** *Let HE be a bootstrappable scheme, then there exists a leveled fully homomorphic encryption scheme as per Definition 3.5.*

Specifically, the leveled homomorphic scheme is such that only the length of the evaluation key depends on the level L. All other parameters of the scheme are distributed identically regardless of the value of L.

For the second variant, we need to define circular security.

Definition 4.2 (weak circular security)**.** *A public key encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is weakly circular secure if it is IND-CPA secure even for an adversary with auxiliary information containing encryptions of all secret key bits: $\{\mathsf{Enc}_{pk}(sk[i])\}_i$.*

Namely, no polynomial time adversary can distinguish an encryption of 0 from an encryption of 1 even given the additional information.

We can now state the second theorem.

Theorem 4.3 ([32], [31])**.** *Let HE be a bootstrappable scheme that is also weakly circular secure. Then there is a fully homomorphic encryption scheme as per Definition 3.4.*

4.3. "Squashing" the Decryption Circuit

The final piece in the puzzle is to determine if we can apply the bootstrapping theorem to the known SWHE

schemes, namely determine if they are in fact capable of evaluating their own decryption circuits (plus some). Surprisingly, as if by a strange law of nature, this turned out to *not* be the case for all the (then available) SWHE schemes. For example, in the approximate GCD based scheme, the decryption circuit turns out to have degree $\widetilde{\Omega}(\gamma\eta)$ which is much larger than the homomorphic capacity guaranteed by Theorem 4.1.

Thus, the final step is to *squash the decryption circuit* of the SWHE scheme, namely transform the scheme into one with the same homomorphic capacity but a decryption circuit that is simple enough to allow bootstrapping. Gentry [32] showed a way to do this by adding a "hint" about the secret key to the evaluation key. The hint is a large set of elements that has a secret sparse subset that sums to the original secret key. In order to ensure that the hint does not reveal damaging information about the secret key, the security of this transformation relies on a new "sparse subset sum" assumption. The sparsity pushes the decryption complexity at the cost of the additional assumption. This approach can be adapted to the later schemes [25], [73], [14] as well, and it crucially utilizes the fact that the decryption equation of these schemes is (almost) a linear equation in the secret key. For example, in the approximate GCD based scheme, noting that p is odd, one can decrypt using the following formula

$$\begin{aligned} \mu' &= [\mathsf{CT} - \lfloor \mathsf{CT}/p \rceil]_2 \\ &= (\mathsf{CT} \bmod 2) \oplus (\lfloor \mathsf{CT}/p \rceil \bmod 2) \end{aligned}$$

which is almost linear in the "secret key" $1/p$. For more details on squashing, we refer the reader to [32], [31], [25].

4.4. Other Instantiations

A number of other works construct FHE schemes following this framework, essentially by building various instantiations of the SWHE scheme. Gentry's original construction [32] of an SWHE scheme was based on (a variant of) the bounded distance decoding problem on ideal lattices drawn according to a certain distribution.[3] In a subsequent work, Gentry [33] showed a worst-case to average-case reduction for this problem, thus basing the security of his scheme on a worst-case problem over ideal lattices. Smart and Vercauteren [73] construct an SWHE scheme, following Gentry's construction closely, but basing it on the average-case hardness of a "small principal ideal problem" (unlike [32], there is no known worst-case to average-case connection for this problem). Brakerski and Vaikuntanathan [14] showed yet another scheme based on the average-case hardness of the "Ring Learning with Errors" (Ring LWE) problem which, by the results of Lyubashevsky, Peikert and Regev [52], is as hard as various standard worst-case problems on ideal lattices. This scheme, in addition, has very simple and efficient algorithms. See [35], [22], [50] for other variants and optimizations.

All these schemes then go through the squashing and bootstrapping transformations to construct FHE.

[3] Roughly speaking, ideal lattices correspond to a geometric embedding of ideals in a number field. For a formal definition, see [32], [52].

5. A New Era of Fully Homomorphic Encryption

Gentry's blueprint and its instantiations leave open a number of important questions.

First, all the constructions based on the blueprint rely on multiple complexity assumptions, the most problematic of them being the little-studied "sparse subset sum assumption" used in squashing the decryption circuit. Is this assumption necessary? In addition, the constructions use ideals in various rings either explicitly or implicitly [25]. Ideals are a natural mathematical object to construct fully homomorphic encryption since they natively support both addition and multiplication operations, but are they necessary for FHE? A final concern is that all the constructions ultimately rely on the hardness of approximating lattice problems to within a subexponential factor (in the dimension n of the lattice). Can we base security on the hardness of approximation in the polynomial range?

Secondly, schemes that follow Gentry's blueprint turn out to have inherent efficiency limitations (see [13] for an argument to this effect). When speaking of efficiency, we are interested in the length of the ciphertext (per bit encrypted) and the keys, and the time it takes to encrypt and decrypt. More importantly, it turns out that the bottleneck in practical deployments of FHE is the per-gate evaluation time, defined as the ratio of the time it takes to evaluate a circuit C homomorphically to the time it takes to evaluate C on plaintext inputs. The schemes that follow Gentry's blue-print [32], [25], [35], [73], [15] have a per-gate evaluation time of $\Omega(\kappa^4)$ (where κ is the security parameter), even by fairly generous estimates.

A series of new works address these concerns. In particular, Brakerski and Vaikuntanathan [15] show that (leveled) FHE can be based on the hardness of the much more standard "learning with error" (LWE) problem introduced by Regev [66] which, by the results of Regev [66] and Peikert [61] is as hard as solving various short vector problems on arbitrary (not ideal) lattices

in the worst case. In effect, they show how to obtain a direct construction of a bootstrappable encryption scheme without having to squash the decryption circuit and thus, without relying on the non-standard sparse subset sum assumption. In a concurrent work, Gentry and Halevi [34] show how to get rid of squashing as well, using a completely different technique. Their construction still relies on ideal lattices.

In an even newer development, Brakerski, Gentry and Vaikuntanathan [13] build on (a refinement of) the main technique in [15] to construct an FHE scheme with asymptotically linear efficiency (namely, a per-gate computation time of $\widetilde{O}(\kappa)$) under the assumption that short vector problems on arbitrary lattices are hard to approximate to within a slightly super-polynomial factor (more precisely, $n^{O(\log n)}$ where n is the dimension of the lattice) in the worst-case.

We now describe the ideas behind these new developments. Since the works of Brakerski and Vaikuntanathan [15], and Gentry and Halevi [35] appear in these proceedings, we will provide here a brief description of the result of [13] which builds on [15] and, at the time of writing, constitutes the state-of-the-art in fully homomorphic encryption, both in terms of the mildness of assumptions as well as efficiency.

5.1. The BGV Result

In this section, we present a high-level, but mostly self-contained description of the FHE scheme of Brakerski, Gentry and Vaikuntanathan [13], which in turn builds on techniques from [15].

The starting point for our description of the scheme is a somewhat homomorphic encryption scheme based on "Ring LWE" [52] (although the scheme can be instantiated with a number of other SWHE schemes in the literature). The scheme works over the rings $R = Z[x]/(f(x))$ where $f(x)$ is an irreducible polynomial of degree n and $R_q = R/qR$ where q is a prime modulus. An additional parameter is an "error distribution" χ over R that outputs polynomials with "small coefficients". As before, we describe a secret-key system with message space $R_2 := R/2R$, for simplicity.

- $\mathsf{SH.Keygen}(1^\kappa)$: Sample $sk := \mathbf{s} \leftarrow R_q$ to be a polynomial with small coefficients chosen from the error distribution χ.
- $\mathsf{SH.Enc}(sk, \mu \in R_2)$: Sample $\mathbf{a} \leftarrow R_q$ at random, and a polynomial $\mathbf{e}$ with small coefficients from an error distribution χ. Output $\mathbf{c} := (\mathbf{a}, \mathbf{as} + 2\mathbf{e} + \mu)$.
- $\mathsf{SH.Dec}(sk, \mathbf{c} = (\mathbf{a}, \mathbf{b}))$: Compute $\widetilde{\mu} := \mathbf{b} - \mathbf{as}$ over R_q and output $\mu := \widetilde{\mu} \pmod 2$.

As before, the success of decryption is contingent on the noise in the ciphertext being "small enough". Homomorphic operations in this scheme increase the noise – multiplication more than addition – and the noise after homomorphically evaluating a multivariate polynomial with degree D and A monomials turns out to be $O(A \cdot n^{O(D)})$. In other words, the noise increases *exponentially in the degree of the polynomial*.

The key contribution in the work of [13] is a new *noise-management technique* that keeps the noise in check by reducing it after homomorphic operations, without bootstrapping. Essentially, the noise will grow to $O(A \cdot n^{O(d)})$, where d *is the depth of the circuit computing the polynomial*. Since the degree D of a circuit is usually exponentially larger than its depth d, we achieve an exponential improvement in the homomorphic capacity.

The key technical tool they use for noise management is the "modulus switching" technique developed by Brakerski and Vaikuntanathan [15], the essence of which is captured in the following lemma.

In words, the lemma says that an evaluator, who does not know the secret key $\mathbf{s}$ but instead only knows a bound on its length, can transform a ciphertext $\mathbf{c}$ modulo q into a different ciphertext $\mathbf{c}'$ modulo p while preserving correctness – namely, $(\mathbf{c}'\mathbf{s} \pmod p) \pmod 2 = (\mathbf{cs} \pmod q) \pmod 2$. The transformation from $\mathbf{c}$ to $\mathbf{c}'$ involves simply scaling by (p/q) and rounding appropriately! Most interestingly, if $\mathbf{s}$ is short and p is sufficiently smaller than q, the "noise" in the ciphertext actually decreases – namely,

$$|\mathbf{c}'\mathbf{s} \pmod p| < |\mathbf{cs} \pmod q|$$

Lemma 5.1. *Let p and q be two odd moduli, and let $\mathbf{c} = (\mathbf{a}, \mathbf{b})$ be a ciphertext modulo q. Define $\mathbf{c}' = (\mathbf{a}', \mathbf{b}')$ to be the integer vector closest to $(p/q) \cdot \mathbf{c} = ((p/q) \cdot \mathbf{a}, (p/q) \cdot \mathbf{b})$ such that $\mathbf{c}' = \mathbf{c} \bmod 2$. Then, for any $\mathbf{s}$ with $|\mathbf{b} - \mathbf{as} \bmod q| < q/2 - (q/p) \cdot \ell_1(\mathbf{s})$, we have*

$$(\mathbf{b}' - \mathbf{a}'\mathbf{s} \bmod p) \bmod 2 = (\mathbf{b} - \mathbf{as} \bmod q) \bmod 2$$

$$\text{and } |\mathbf{b}' - \mathbf{a}'\mathbf{s} \bmod p| < (p/q) \cdot |\mathbf{b} - \mathbf{as} \bmod q| + \ell_1(\mathbf{s})$$

where $\ell_1(\mathbf{s})$ is the ℓ_1-norm of (the co-efficient vector corresponding to) $\mathbf{s}$.

Amazingly, this trick permits the evaluator to reduce the magnitude of the noise without knowing the secret key, and without bootstrapping. In other words, modulus switching gives us a very powerful and lightweight way to *manage the noise* in FHE schemes!

The BGV Noise Management Technique. At first, it may look like modulus switching is not a very effective noise management tool. If p is smaller than q, then of course modulus switching may reduce the magnitude

of the noise, but it reduces the modulus size by essentially the same amount. In short, the ratio of the noise to the "noise ceiling" (the modulus size) does not decrease at all. Isn't this ratio what dictates the remaining homomorphic capacity of the scheme, and how can potentially worsening (certainly not improving) this ratio do anything useful?

In fact, it's not just the ratio of the noise to the "noise ceiling" that's important. The *absolute magnitude of the noise* is also important, especially in multiplications. Suppose that $q \approx x^k$, and that you have two mod-q SWHE ciphertexts with noise of magnitude x. If you multiply them, the noise becomes x^2. After 4 levels of multiplication, the noise is x^{16}. If you do another multiplication at this point, you reduce the ratio of the noise ceiling (i.e. q) to the noise level by a huge factor of x^{16} – i.e., you reduce this gap very fast. Thus, the actual magnitude of the noise impacts how fast this gap is reduced. After only $\log k$ levels of multiplication, the noise level reaches the ceiling.

Now, consider the following alternative approach. Choose a *ladder of gradually decreasing moduli* $\{q_i \approx q/x^i\}$ for $i < k$. After you multiply the two mod-q ciphertexts, switch the ciphertext to the smaller modulus $q_1 = q/x$. As the lemma above shows, the noise level of the new ciphertext (now with respect to the modulus q_1) goes from x^2 back down to x. (Let's suppose for now that $\ell_1(\mathbf{s})$ is small in comparison to x so that we can ignore it.) Now, when we multiply two ciphertexts (wrt modulus q_1) that have noise level x, the noise again becomes x^2, but then we switch to modulus q_2 to reduce the noise back to x. In short, each level of multiplication only reduces the ratio (noise ceiling)/(noise level) by a factor of x (not something like x^{16}). With this new approach, we can perform about k (not just $\log k$) levels of multiplication before we reach the noise ceiling. We have just increased (without bootstrapping) the number of multiplicative levels that we can evaluate by an exponential factor!

This exponential improvement is enough to achieve leveled FHE without squashing or bootstrapping. For *any* polynomial k, we can evaluate circuits of depth k. The performance of the scheme degrades with k – e.g., we need to set $q = q_0$ to have bit length proportional to k – but it degrades only polynomially with k.

Performance-wise, this scheme trounces previous (bootstrapping-based) FHE schemes (at least asymptotically; the concrete performance remains to be seen). Instantiated with ring-LWE, it can evaluate L-level arithmetic circuits with per-gate computation $\tilde{O}(\kappa \cdot L^3)$ – i.e., computation *quasi-linear* in the security parameter. Since the ratio of the largest modulus (namely, $q \approx x^L$) to the noise (namely, x) is exponential in L, the scheme relies on the hardness of approximating short vectors to within an exponential in L factor. The performance can be improved further using batching tricks.

In essence, the new noise management technique allows us to evaluate exponentially deeper circuits at the same cost as before. Combining this technique with bootstrapping gives us further performance gains and allows us to base security on better assumptions – namely, the hardness of approximating shortest vector to a quasi-polynomial factor (in the dimension n). See [13] for more details.

6. Future Directions

The study of fully homomorphic encryption has led to a number of new and exciting concepts and questions, as well as a powerful tool-kit to address them. We conclude this survey by describing a number of research directions related to FHE and more generally, the problem of computing on encrypted data.

6.1. Can Fully Homomorphic Encryption be Practical?

While Gentry's original construction was viewed as being impractical, recent constructions and implementation efforts have drastically improved the efficiency of fully homomorphic encryption. The initial implementation efforts focused on Gentry's original scheme and its variants [73], [35], [74], [22] which seemed to pose rather inherent efficiency bottlenecks. Later implementations leverage the recent algorithmic advances [14], [15], [13] that result in asymptotically better FHE systems, as well as new algebraic techniques to improve the concrete efficiency of these schemes [50], [36], [74].

6.2. Non-Malleability and Homomorphic Encryption

Homomorphism and Non-malleability are antipodal properties of an encryption scheme. Homomorphic encryption schemes permit anyone to transform an encryption of a message m into an encryption of $f(m)$ for non-trivial functions f. Non-malleable encryption, on the other hand, prevents precisely this sort of thing – it requires that no adversary be able to transform an encryption of m into an encryption of any "related" message. In reality, what we need is a combination of both properties that *selectively permit homomorphic computations*. Namely, the evaluator should be able to homomorphically compute any function from some prespecified class $\mathcal{F}_{\mathsf{hom}}$, yet she should not be able to transform an encryption of m into an encryption of $f(m)$ for any $f \notin \mathcal{F}_{\mathsf{hom}}$. Thus, the question is: *Can we control what is being (homomorphically) computed?*

Formalizing this notion turns out to be tricky. Boneh, Segev and Waters [12] propose the notion of *targeted malleability* – a candidate formalization of such a requirement – as well as constructions of such encryption schemes. Their encryption scheme is based on a strong "knowledge of exponent-type" assumption, and allows iterative evaluation of at most t functions, where t is a pre-specified constant. Improving their construction as well as the underlying complexity assumptions is an important open problem.

Furthermore, it is interesting to extend the definition of non-malleability to allow for chosen ciphertext attacks. Consider, for example, implementing an encrypted targeted advertisement system that generates advertisements depending on the contents of a user's e-mail. Since the e-mail is stored encrypted (with the user's public key), the e-mail server performs a homomorphic evaluation and computes an encrypted advertisement to be sent back to the user. The user decrypts it, and performs an action depending on what she sees. Namely, if the advertisement is relevant, she might choose to click on it, but otherwise, she will ignore it. Now, if the e-mail server is privy to this information, namely whether the user clicked on the ad or not, they can use this as a restricted "decryption oracle" to break the security of the user's encryption scheme and perhaps even recover her secret key. Such attacks are ubiquitous whenever we compute on encrypted data, almost to the point that CCA security seems inevitable. Yet, it is easy to see that chosen ciphertext secure (CCA2-secure) homomorphic encryption schemes cannot exist. Thus, we need an appropriate security definition and constructions that achieve the definition.

6.3. FHE and Functional Encryption

Homomorphic encryption schemes permit anyone to evaluate functions on encrypted data, but the evaluators never see any information about the result. Is is possible to construct an encryption scheme where a user can compute $f(m)$ *in the clear* from an encryption of a message m, but she should learn no other information about m (including the intermediate results in the computation of f)? Thus, the question is: *Can we control what the evaluator can see?* Such an encryption scheme is called a functional encryption scheme, first defined by Sahai and Waters [71] and explored in a number of works ([48], [51], [12], [2] and many others). Although these constructions work for several interesting families of functions (such as monotone formulas and inner products), constructing a *fully functional encryption scheme* is wide open.

More generally, what we need is a new, broad vision for encryption systems that provide us with *fine-grained control* over what one can see and what one can compute on data.

6.4. Other Problems and Applications

Another important open question relates to the cryptographic assumptions underlying FHE schemes. All known FHE schemes are based on the hardness of lattice problems. Can we construct FHE from other, perhaps number-theoretic assumptions? How about the hardness of factoring or discrete logarithms?

Aside from the multitude of scenarios where it is beneficial to keep all data encrypted and to perform computations on encrypted data, fully homomorphic encryption has been used to solve a number of other problems in cryptography. Two such examples are the problems of verifiably outsourcing computation [41], [30], [19], [6] and constructing short non-interactive zero-knowledge proofs [32]. Some of the applications of FHE do not require its full power – for PIR, it is sufficient to have a somewhat homomorphic encryption scheme capable of evaluating simple database indexing functions.

Acknowledgments. This survey has benefited immensely from discussions with Boaz Barak, Dan Boneh, Zvika Brakerski, Craig Gentry, Shafi Goldwasser, Shai Halevi, Kristin Lauter, Michael Naehrig, Nigel Smart and Marten van Dijk. In writing this, I have also liberally made use of material from a number of my papers [25], [14], [15], [13], for which I thank my co-authors.

REFERENCES

[1] B. Adida, "Helios: Web-based open-audit voting," in *USENIX Security Symposium*, 2008, pp. 335–348.

[2] S. Agrawal, D. M. Freeman, and V. Vaikuntanathan, "Functional encryption for inner product predicates from learning with errors," Cryptology ePrint Archive, Report 2011/410, 2011, http://eprint.iacr.org/.

[3] J. H. Ahn, D. Boneh, J. Camenisch, S. Hohenberger, A. Shelat, and B. Waters, "Computing on authenticated data," Cryptology ePrint Archive, Report 2011/096, 2011, http://eprint.iacr.org/.

[4] M. Ajtai and C. Dwork, "A public-key cryptosystem with worst-case/average-case equivalence," in *STOC*, 1997, pp. 284–293.

[5] B. Applebaum, "Key-dependent message security: Generic amplification and completeness," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed., vol. 6632. Springer, 2011, pp. 527–546.

[6] B. Applebaum, Y. Ishai, and E. Kushilevitz, "From secrecy to soundness: Efficient verification via secure computation," in *ICALP (1)*, 2010, pp. 152–163.

[7] M. Bellare and S. Goldwasser, "New paradigms for digital signatures and message authentication based on non-interative zero knowledge proofs," in *CRYPTO*, 1989, pp. 194–211.

[8] M. Bellare and P. Rogaway, "Optimal asymmetric encryption," in *EUROCRYPT*, 1994, pp. 92–111.

[9] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications (extended abstract)," in *STOC*, 1988, pp. 103–112.

[10] D. Boneh and D. M. Freeman, "Homomorphic signatures for polynomial functions," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed., vol. 6632. Springer, 2011, pp. 149–168.

[11] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Theory of Cryptography - TCC'05*, ser. Lecture Notes in Computer Science, vol. 3378. Springer, 2005, pp. 325–341.

[12] D. Boneh, G. Segev, and B. Waters, "Targeted malleability: Homomorphic encryption for restricted computations," Cryptology ePrint Archive, Report 2011/311, 2011, http://eprint.iacr.org/2011/311.

[13] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping," Cryptology ePrint Archive, Report 2011/277, 2011, http://eprint.iacr.org/2011/277.pdf.

[14] Z. Brakerski and V. Vaikuntanathan, " Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *CRYPTO*, vol. 6841, 2011, p. 501.

[15] ——, "Efficient fully homomorphic encryption from (standard) LWE," in *FOCS*, 2011, appears in this proceedings. Also available at Cryptology ePrint Archive, Report 2011/344.

[16] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," in *EUROCRYPT*, 1999, pp. 402–414.

[17] Y. Chen and P. Q. Nguyen, "Faster algorithms for approximate common divisors: Breaking fully-homomorphic-encryption challenges over the integers," Cryptology ePrint Archive, Report 2011/436, 2011, http://eprint.iacr.org/2011/436.

[18] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, 1998.

[19] K.-M. Chung, Y. T. Kalai, and S. P. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *CRYPTO*, 2010, pp. 483–501.

[20] J. D. Cohen and M. J. Fischer, "A robust and verifiable cryptographically secure election scheme (extended abstract)," in *FOCS*. IEEE, 1985, pp. 372–382.

[21] H. Cohn and N. Heninger, "Approximate common divisors via lattices," Cryptology ePrint Archive, Report 2011/437, 2011, http://eprint.iacr.org/2011/437.

[22] J.-S. Coron, A. Mandal, D. Naccache, and M. Tibouchi, "Fully homomorphic encryption over the integers with shorter public keys," in *CRYPTO*, 2011, pp. 487–504.

[23] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of paillier's probabilistic public-key system," in *Public Key Cryptography*, 2001, pp. 119–136.

[24] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644–654, 1976.

[25] M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *EUROCRYPT*, 2010, pp. 24–43, full Version in http://eprint.iacr.org/2009/616.pdf.

[26] D. Dolev, C. Dwork, and M. Naor, "Nonmalleable cryptography," *SIAM J. Comput.*, vol. 30, no. 2, pp. 391–437, 2000.

[27] U. Feige, D. Lapidot, and A. Shamir, "Multiple noninteractive zero knowledge proofs under general assumptions," *SIAM J. Comput.*, vol. 29, no. 1, pp. 1–28, 1999.

[28] M. Fellows and N. Koblitz, "Combinatorial cryptosystems galore!" Finite Fields: Theory, Applications and Algorithms, pp. 51–61, 1993.

[29] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *CRYPTO*, 1984, pp. 10–18.

[30] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," in *CRYPTO*, 2010, pp. 465–482.

[31] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009, http://crypto.stanford.edu/craig.

[32] ——, "Fully homomorphic encryption using ideal lattices," in *STOC*, 2009, pp. 169–178.

[33] ——, "Toward basing fully homomorphic encryption on worst-case hardness," in *CRYPTO*, 2010, pp. 116–137.

[34] C. Gentry and S. Halevi, "Fully homomorphic encryption without squashing using depth-3 arithmetic circuits," Cryptology ePrint Archive, Report 2011/279, 2011, to appear in this Proceedings.

[35] ——, "Implementing gentry's fully-homomorphic encryption scheme," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, K. G. Paterson, Ed., vol. 6632. Springer, 2011, pp. 129–148.

[36] C. Gentry, S. Halevi, and N. Smart, "Personal communication," 2011.

[37] C. Gentry, S. Halevi, and V. Vaikuntanathan, "-hop homomorphic encryption and rerandomizable yao circuits," in *CRYPTO*, ser. Lecture Notes in Computer Science, T. Rabin, Ed., vol. 6223. Springer, 2010, pp. 155–172.

[38] ——, "A simple BGN-type cryptosystem from LWE," in *EUROCRYPT*, 2010, pp. 506–522.

[39] C. Gentry and Z. Ramzan, "Single-database private information retrieval with constant communication rate," in *ICALP*, ser. Lecture Notes in Computer Science, L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, Eds., vol. 3580. Springer, 2005, pp. 803–815.

[40] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," *J. ACM*, vol. 43, no. 3, pp. 431–473, 1996.

[41] S. Goldwasser, Y. T. Kalai, and G. N. Rothblum, "Delegating computation: interactive proofs for muggles," in

STOC, 2008, pp. 113–122.

[42] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984.

[43] J. Groth, R. Ostrovsky, and A. Sahai, "Non-interactive zaps and new techniques for nizk," in *CRYPTO*, 2006, pp. 97–111.

[44] ——, "Perfect non-interactive zero knowledge for np," in *EUROCRYPT*, 2006, pp. 339–358.

[45] N. Howgrave-Graham, "Approximate integer common divisors," in *CaLC*, 2001, pp. 51–66.

[46] Y. Ishai and A. Paskin, "Evaluating branching programs on encrypted data," in *TCC*, ser. Lecture Notes in Computer Science, S. P. Vadhan, Ed., vol. 4392. Springer, 2007, pp. 575–594.

[47] A. Juma and Y. Vahlis, "Protecting cryptographic keys against continual leakage," in *CRYPTO*, ser. Lecture Notes in Computer Science, T. Rabin, Ed., vol. 6223. Springer, 2010, pp. 41–58.

[48] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *EUROCRYPT*, 2008, pp. 146–162.

[49] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval," in *FOCS*, 1997, pp. 364–373.

[50] K. Lauter, M. Naehrig, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" pre-print available at http://eprint.iacr.org/2011/405.

[51] A. B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *EUROCRYPT*, ser. Lecture Notes in Computer Science, H. Gilbert, Ed., vol. 6110. Springer, 2010, pp. 62–91.

[52] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *EUROCRYPT*, 2010, pp. 1–23, draft of full version was provided by the authors.

[53] C. A. Melchor, P. Gaborit, and J. Herranz, "Additively homomorphic encryption with *d*-operand multiplications," in *CRYPTO*, 2010, pp. 138–154.

[54] S. Micali, "Computationally sound proofs," *SIAM J. Comput.*, vol. 30, no. 4, pp. 1253–1298, 2000.

[55] D. Naccache and J. Stern, "A new public key cryptosystem based on higher residues," in *ACM Conference on Computer and Communications Security*, 1998, pp. 59–66.

[56] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," in *STOC*, 1990, pp. 427–437.

[57] T. Okamoto and S. Uchiyama, "A new public-key cryptosystem as secure as factoring," in *EUROCRYPT*, 1998, pp. 308–318.

[58] R. Ostrovsky and W. E. Skeith III, "A survey of single-database private information retrieval: Techniques and applications," in *Public Key Cryptography*, ser. Lecture Notes in Computer Science, T. Okamoto and X. Wang, Eds., vol. 4450. Springer, 2007, pp. 393–411.

[59] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999, pp. 223–238.

[60] K. G. Paterson, Ed., *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6632. Springer, 2011.

[61] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: extended abstract," in *STOC*, 2009, pp. 333–342.

[62] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," in *STOC*, 2008, pp. 187–196.

[63] T. Rabin, Ed., *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, ser. Lecture Notes in Computer Science, vol. 6223. Springer, 2010.

[64] C. Rackoff and D. R. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack," in *CRYPTO*, 1991, pp. 433–444.

[65] O. Regev, "New lattice-based cryptographic constructions," *J. ACM*, vol. 51, no. 6, pp. 899–942, 2004.

[66] ——, "On lattices, learning with errors, random linear codes, and cryptography," in *STOC*, 2005, pp. 84–93.

[67] R. Rivest, L. Adleman, and M. Dertouzos, "On data banks and privacy homomorphisms," in *Foundations of Secure Computation*. Academic Press, 1978, pp. 169–177.

[68] R. L. Rivest, A. Shamir, and L. M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems (reprint)," *Commun. ACM*, vol. 26, no. 1, pp. 96–99, 1983.

[69] G. Rothblum, "Delegating computation reliably: Paradigms and constructions," Ph.D. dissertation, MIT, 2009, http://dspace.mit.edu/handle/1721.1/54637.

[70] R. Rothblum, "Homomorphic encryption: From private-key to public-key," in *TCC*, 2011, pp. 219–234.

[71] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *EUROCRYPT*, 2005, pp. 457–473.

[72] T. Sander, A. Young, and M. Yung, "Non-interactive cryptocomputing for NC^1," in *FOCS*, 1999, pp. 554–567.

[73] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Public Key Cryptography*, ser. Lecture Notes in Computer Science, P. Q. Nguyen and D. Pointcheval, Eds., vol. 6056. Springer, 2010, pp. 420–443.

[74] N. Smart and F. Vercauteren, "Fully homomorphic simd operations," Cryptology ePrint Archive, Report 2011/133, 2011, http://eprint.iacr.org/2011/133.

[75] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *ASIACRYPT*, 2010, pp. 377–394.