# Lecture 21 — Homomorphic Encryption 2: Construction of "mildly homomorphic" encryption.

Boaz Barak

April 19, 2010

**Reading:** Gentry's thesis, paper by van Dijk, Gentry, Halevi and Vaikuntanathan.

**Notation:** If $x \in \mathbb{R}$, then $\lfloor x \rceil$ denotes the nearest integer to $x$ (say, breaking ties downward), $\lfloor x \rfloor$ denotes the nearest integer smaller than $x$ and $\lceil x \rceil$ denotes the nearest integer larger than $x$.

**Constructing homomorphic encryption** We saw in the last couple of lectures that homomorphic encryptions can be used to do wonderful things, but the same holds for perpetual motion machines, cold fusion, unicorns, etc...

So, the question whether we can actually construct such schemes. Since the question was raised in 1978, there have been no significant candidate for a homomorphic encryption scheme. This was changed last year, when Gentry gave the first such construction. The construction relies on somewhat non-standard, but still rather reasonable assumptions. Also, as mentioned, it is still not practical, requiring at least $k^8$ operation to achieve $2^k$ security. Hopefully, with time we will see improved constructions, using more standard assumptions and more efficient. We will see a close variant of Gentry's scheme now. We remark that all the applications we saw (zero knowledge, multi-party computation, private information retrieval) have alternative constructions that utilize much more standard assumptions.

**Plan** We'll start by showing a "mildly" homomorphic encryption scheme, and then modify it to boost it to a fully homomorphic encryption next lecture. Since the definition of mildly homomorphic is somewhat unnatural, I'll first show the encryption scheme, and only later discuss the definition it satisfies. I note that initially we will construct a *private key* encryption scheme. We will then note that a fully homomorphic private key encryption easily gives rise to such a public key scheme (exercise!).

**Noisy gcd.** Consider the following question: you're given $X_1, \ldots, X_{\mathrm{poly}(n)}$ all $100n$ bit strings, and told that either: **(I)** all of them are random and independent in $[N] = [2^{100n}]$ or **(II)** for all $i$, $X_i = Q_i P$, where $P$ is chosen once at random in $[2^n]$ and $Q_i$ are chosen independently and randomly in $\{1..\lfloor N/P \rfloor\}$. How can you distinguish between the two cases?

Now suppose that we change case **(II** so that $X_i = Q_i P + E_i$ where $E_i$ is chosen independently at random in $[2^{\sqrt{n}}]$.

The *noisy gcd conjecture* is that now **(I)** and **(II)** are computationally indistinguishable. I'll call this LDN for "learning divisor with noise".[1] That is, in the LDN assumption, you are

[1] I phrased LDN as a decision problem, but in the paper van Dyck et al show this is equivalent to the search problem of actually finding the divisor.

given either a box that gives you random numbers in $[N]$ or numbers of the form $QP + E$ where $P$ is some secret random number in $[2^n] = N^{1/10}$ and $E$ is random in $[2^{\sqrt{n}}]$.

**A useful variant.** It turns out that LDN is equivalent to the case that $P$ is odd and $E_i$ is even, in which case we just write $X_i = Q_iP + 2E_i$. This is left as an exercise. One simple claim that is used is the following:

**Shifting interval claim:** If $I$ is an interval, then $U_I$ is within $|a|/|I|$ statistical distance to $U_{I+a}$ where $U_S$ denotes the uniform distribution on the set $S$, and $I + a$ denotes the interval shifted by $a$.

Does LDN imply that **(I)** and **(II)** are indistinguishable even when both $P$ and $E_i$ are even?

**Increasing noise only helps** The following observation will be of use: if LDN is true with noise magnitude $2^{\sqrt{n}}$, it's true with any magnitude in $[2^{\sqrt{n}}, 2^n]$. (In fact, in the latter case the two distributions become *statistically indistinguishable.*)

**Basic cryptosystem** We now construct a CPA-secure private key encryption $(\mathsf{Enc}, \mathsf{Dec})$ based on LDN:

**Key** $P \leftarrow_{\mathrm{R}} [2^n]$. We denote $N = 2^{100n}$.

**Encryption** To encrypt the bit $b \in \{0, 1\}$, choose $Q \leftarrow_{\mathrm{R}} \{1..\lfloor N/Q \rfloor\}$, $E \leftarrow_{\mathrm{R}} [2^{\sqrt{n}}]$, output $X = QP + 2E + b$.

**Decryption** To decrypt $X$, output $(X \pmod P) \pmod 2$.

**Correctness** Since $E \ll P$, $QP + 2E + b \pmod P = 2E + b$, and then taking $\pmod 2$ we get $b$.

**Security** We need to show $\mathsf{Enc}(0) \approx \mathsf{Enc}(1)$, which will follow by showing in both cases they are indistinguishable from $U_{[N]}$. Indeed, under our assumptions all the ciphertexts $X_1, \ldots, X_{\mathrm{poly}(n)}$ that the adversary obtains in a CPA attack are of the form $X_i = Q_iP + 2E_i$ or $X_i = Q_iP + 2E_i + 1$, but since $Q_iP + 2E_i \approx U_{[N]}$, then also the same holds for $Q_iP + 2E_i + 1$ via the shifting interval claim.

**Homomorphic** In what sense is this system homomorphic? We claim that it satisfies the following: given $X, X'$ that are encryptions of $b, b'$ respectively, we can manufacture (without access to the secret key) ciphertexts $X_\oplus$ and $X_\times$ such that $X_\oplus$ will decrypt to $b \oplus b'$ and $X_\times$ will decrypt to $b \cdot b'$.

This is very simple— just multiply or add the ciphertexts!

Write $X = QP + 2E + b$ and $X' = Q'P + 2E' + b'$ then

$$X + X' = (Q + Q')P + 2(E + E') + (b + b')$$

and so, since $E + E' \ll P$, it's clear that $(X + X' \pmod P) \pmod 2 = b + b' \pmod 2$.

now

$$X \cdot X' = QQ'P^2 + 2E'QP + b'QP + 2EQ'P + 4EE' + 2Eb' + bQ'P + 2bE' + bb'$$

lets group together all the terms that are multiples of $P$, and then the remaining terms that are definitely even, to get

$$X \cdot X' = (QQ'P + 2E'Q + b'Q + 2EQ + bQ')P + 2(2EE' + Eb' + bE') + bb'$$

now we have $2EE' + Eb' + bE' \leq 3 \cdot 2^{2\sqrt{n}} \leq 2^{3\sqrt{n}} \ll P$ and so we get $(X \cdot X' \pmod{P}) \pmod{2} = bb' \pmod{2}$.

**Are we there yet?** This encryption scheme guarantees that we can transform ciphertexts corresponding to $b$ and $b'$ into ciphertexts corresponding to $b \oplus b'$ or $bb'$ respectively, and by combining them one can easily get a ciphertext corresponding to $\overline{b \wedge b'}$, so why isn't this a fully homomorphic encryption scheme?

The answer is that while, for example, the ciphertext $X_\times$ will decrypt to $bb'$, it will not be statistically close to a standard encryption of $bb'$. In fact, it will not even have the same length! Indeed, $X_\times$ will be a number of size roughly $N^2$.

This also shows that there is a limit to how much we can continue applying these $\oplus$ and $\times$ operation. This limit comes into play in both the size of the ciphertexts and the magnitude of the noise, and in both cases the $\times$ operation is much more expensive than the $\oplus$, and we can only compose it with itself a logarithmic number of times:

- *Size of ciphertext* If $X, X'$ were of $m$ bits size, then $X_\oplus$ will have size about $m+1$, while $X_\times$ will have size $2m$.
- *Magnitude of noise* if $X, X'$ had magnitude of noise at most $E$, then $X_\oplus$ will have magnitude of noise at most $2E$, while $X_\times$ will have magnitude of noise at most $3E^2 < E^3$.

Suppose we compose these operations in a polynomial size circuit with $\oplus$ and $\times$ gates, where we allow arbitrary fan-in for the $\oplus$ gates and fan-in two for the $\times$ gates. If the circuit has depth at most $\log n/10$ then we'll be OK: let $E_i$ be the magnitude of noise at level $i$, then $E_1 \leq 2^{\sqrt{n}}$ and $E_i \leq E_{i-1}^3$ for all $i$. Similarly, let $s_i$ be the number of bits of the ciphertexts at level $i$, then $s_1 = 100n$ and $s_i \leq 2s_{i-1}$ for all $i$.

So we can evaluate circuits up to that size. But as mentioned the guarantee is much weaker than what we wanted— all we know is that the decryption will succeed, and this is in some sense trivial (see exercise).

**Recap of basic scheme** Unfortunately we will now have to complicate our scheme somewhat, in preparation for the fully homomorphic scheme.

**Reducing ciphertext size** The fact that the ciphertexts grow (especially with multiplication) makes the scheme much less efficient, and also as mentioned above makes it trivial in some sense. We now want to keep all the ciphertexts at reasonable size.

**First attempt** As a first attempt - lets reduce all numbers modulo $N$. Does this work??

**Second attempt** Lets assume the encryption algorithm also outputs some number $N'$ close to $N$ such that $N' = QP + E$ (for example, trying many such random numbers and outputting the largest one — this does not give any information about $P$ since they are indistinguishable from random numbers in $[N]$). I claim that in this case reducing modulo $N'$ will work for addition.

**Actual construction** We now turn to the actual construction, however to do so we need to strengthen the LDN assumption. We assume now that the adversary has a polynomial number of interactions with a box to which he can give any number $N \geq 2^{100n}$ of his choice and gets in case **(I)** a random number in $[N]$, and in case **(II)** a number of the form $QP + E$ where $P$, as before was chosen once for the all interactions at random from $[2^n]$, $Q$ is chosen at random from 1 to $\lfloor N/P \rfloor$ and $E$ is chosen at random in $[-2^{n^{0.2}}, +2^{n^{0.2}}]$.[2]

Our encryption scheme will now be the following:

**Key** $P \leftarrow_{\mathrm{R}} [2^n]$.

**Public parameters** (This is used by the EVAL procedures, one can also think of these as being appended to each encryption.) Choose $N_0, \ldots, N_{1000n}$ s.t. for all $i$, $N_i = Q_i P_i + E_i$ with $|E_i| \leq 2^{\sqrt{n}}$ and $N_0 \in [2^{100n-1}, 2^{100n}]$, $N_i \in [1.5N_{i-1}, 2N_i)$. For every $U$, a number of the form $QP + E$ in the interval $[1.5U, 2U]$ can be chosen by taking a random $Q \in \{1..\lfloor 2U/P \rfloor\}$, and $E \in [-2^{\sqrt{n}}, +2^{\sqrt{n}}]$ and outputting $QP+E$ if $QP+E$ is in this interval (which will happen with probability about 3/4), or otherwise trying again.

**Encryption** To encrypt the bit $b \in \{0,1\}$, choose $Q \leftarrow_{\mathrm{R}} \lfloor N_0/Q \rfloor$, $E \leftarrow_{\mathrm{R}} [2^{\sqrt{n}}]$, output $X = QP + 2E + b$.

**Decryption** To decrypt $X$, output $(X + 2\lfloor P/4 \rfloor \pmod{P}) \pmod{2}$. (The addition of $2\lfloor P/4 \rfloor$ it to handle negative noise, see below.)

**Add** Given $X, X'$ we define $ADD(X, X')$ as $X + X' \pmod{N_0}$.

**Mult** Given $X, X'$ we compute $MULT(X, X')$ as follows: let $Y_{1000n+1} = X \cdot X'$, and for all $i$ let $Y_i = Y_{i+1} \pmod{N_i}$. We output $Y_0$.

**Analysis** Clearly the output of $\mathsf{Enc}$, $ADD$ and $MULT$ is always a number in $[N_0]$, so now we want to argue that the system is secure, and decryption will succeed on both outputs generated by encryption and by Add and Mult.

**Security** All operations are done with access to the "box" that outputs values close to multiples of $P$. If that box is replaced with the box that on input $N$ outputs a random number in $[N]$, then the encryptions of 0 and of 1 are *statistically* indistinguishable.

**Decryption succeeds on plaintext** Since $2\lfloor p/4 \rfloor$ is an even number of size about $P/2$ we have the guarantee that $2\lfloor P/4 \rfloor + 2E + b \in (0, P)$ for all $b \in \{0,1\}$ and $E$ such that $|E| < P/5$ and $2\lfloor P/4 \rfloor + 2E + b \pmod{2} = b$ for all $b \in \{0,1\}$. Hence

$$(QP + 2\lfloor P/4 \rfloor + 2E + b \pmod{P}) \pmod{2} = 2\lfloor P/4 \rfloor + 2E + b \pmod{2} = b$$

**Decryption succeeds on ADD, MULT** This follows from the following claim:

**Claim 1.** *Let $N = QP + E$ and let $X = Q'P' + E'$ be such that $X \leq cN$. Then $X \pmod{N} = Q''P + E''$ such that $|E''| \leq |E'| + c|E|$.*

Proof by picture.

**Can we handle arbitrary circuits???** Yes - if the decryption algorithm can be written as a circuit with depth less than $\log n/10$.

**Re-randomization**

[2] We make the assumption slightly stronger by thinking of smaller noise, this is just for simplification of notation later on, and we could have stayed with $2^{\sqrt{n}}$. Considering negative noise is also not a major issue, but will help us slightly in the future.