Homomorphic Encryption and Lattices, Spring 2011 Instructor: Shai Halevi

Constructions of FHE

June 2, 2011 Scribe: Omer Paneth

1 From SWHE to FHE

In the previous class we constructed a SWHE scheme with $sk = w$,$pk = (d, r)$ and $\mathsf{Dec}_w(c) = [c \cdot w]_d \mod 2$. We want to use "bootstrapping" to convert our SWHE to FHE. Namely we add $c^* = \mathsf{Eec}(w)$ to the public key (assuming circular security), then, given two ciphertexts c_1, c_2, consider the functions:

$$\mathsf{ADD}_{c_1,c_2}(sk) = \mathsf{Dec}_{sk}(c_1) + \mathsf{Dec}_{sk}(c_2)$$
$$\mathsf{MULT}_{c_1,c_2}(sk) = \mathsf{Dec}_{sk}(c_1) \cdot \mathsf{Dec}_{sk}(c_2)$$

If we can evaluate the function homomorphically on c^* then we get two other ciphertexts $c^+, c^\times$ s.t. $\mathsf{Dec}_w(c^+) = c_1 + c_2$ and $\mathsf{Dec}_w(c^\times) = c_1 \cdot c_2$. Our goal is therefore to get a bootstrappable scheme, namely one where the functions $\mathsf{ADD}, \mathsf{MULT}$ are within the homomorphic capacity of the scheme for every two "evaluated ciphertexts" c_1, c_2 and "fresh ciphertext" c^*. So far we have a SWHE scheme that can evaluate polynomials of degree up to $\sqrt{n}$ with up to $n^{2\sqrt{n}}$ terms. But the decryption algorithm for this scheme is given by $\mathsf{Dec}_w(c) = [c \cdot w]_d \mod 2$, where c, w, d have $O(n^{1.5})$ bits. Thus a Boolean circuit to evaluate the decryption operation will be of degree $\tilde{O}(n^{1.5})$ which is too much for the scheme to handle.

We want to reduce the complexity of the decryption without decreasing the homomorphic capacity. We therefore add another "hint" about the secret key to the public key, namely a set of S elements $x_1, \ldots, x_S \in \mathbb{Z}_d$ such that there exists a very sparse subset of the x_i's that sums up to w modulo d. Although in principle adding such additional "hint" may compromise the security of the cryptosystem, in this case one can prove that if the "sparse subset-sum problem" (SSSP) is hard then the cryptosystem remains secure.

Let $\vec{\sigma} = \sigma_1\sigma_2 \cdots \sigma_S$ be the characteristic vector of this subset, namely $\sum_{i=1}^{S} \sigma_i x_i \equiv w \pmod{d}$ and $\mathsf{HW}(\vec{\sigma}) = s << S$. We now view $\vec{\sigma}$ as the secret key. Given a ciphertext c we post-process it to get $y_i = [c \cdot x_i]_d$, $i \in [S]$. Now decryption is given by:

$$\mathsf{Dec}_{\vec{\sigma}}(\vec{y}) = \left[\left[\sum_{i=1}^{S} \sigma_i y_i\right]_d\right]_2 = \left[\left[c\sum_{i=1}^{S} \sigma_i x_i\right]_d\right]_2 = [[cw]_d]_2$$

We would like to express the function $\mathsf{Dec}_{\vec{\sigma}}(\vec{y})$ as a low degree polynomial in the bits σ_i. Since d is odd we have:

$$\left[\left[\sum_{i=1}^{S} \sigma_i y_i\right]_d\right]_2 = \left[\sum_{i=1}^{S} \sigma_i y_i - d \cdot \left\lceil \sum_{i=1}^{S} \frac{\sigma_i y_i}{d} \right\rfloor\right]_2 = \sum_{i=1}^{S} \sigma_i \left[y_i\right]_2 - \left[\left\lceil \sum_{i=1}^{S} \frac{\sigma_i y_i}{d} \right\rfloor\right]_2$$

Notice that the expression $\sum_{i=1}^{S} \sigma_i \left[y_i\right]_2$ is linear. In order to show how to express the term $\left[\left\lceil \sum_{i=1}^{S} \frac{\sigma_i y_i}{d} \right\rfloor\right]_2$ as a low degree polynomial, we first consider parameters for the scheme s.t. for ciphertext c, $|[wc]_d|$ is much less than $\frac{d}{2}$. In particular we require:

$$|[wc]_d| = \left|\left[\sum_{i=1}^{S} \sigma_i y_i\right]_d\right| < \frac{d}{2(s+1)} \Rightarrow \left|\left[\sum_{i=1}^{S} \frac{\sigma_i y_i}{d}\right]\right| < \frac{1}{2(s+1)}$$

Second, we consider a low precision representation of $\frac{y_i}{d}$. For $i \in [S]$ let z_i be the number $\frac{y_i}{d}$ with only $l = \lceil log(s+1) \rceil$ bits of representation, i.e. $\left|z_i - \frac{y_i}{d}\right| \leq 2^{-(l+1)} \leq \frac{1}{2(s+1)}$. Since only s of the σ_i's are 1 and the rest are 0, we have:

$$\left| \sum_{i=1}^{S} \sigma_i z_i - \sum_{i=1}^{S} \frac{\sigma_i y_i}{d} \right| \leq \frac{s}{2(s+1)}$$

Since the distance of $\sum_{i=1}^{S} \frac{\sigma_i y_i}{d}$ from the nearest integer is less than $\frac{1}{2(s+1)}$, the distance of $\sum_{i=1}^{S} \sigma_i z_i$ from the same integer is less than $\frac{1}{2(s+1)} + \frac{s}{2(s+1)} = \frac{1}{2}$ and hence $\left\lceil \sum_{i=1}^{S} \sigma_i z_i \right\rfloor = \left\lceil \sum_{i=1}^{S} \frac{\sigma_i y_i}{d} \right\rfloor$. Altogether we have $\mathsf{Dec}_{\vec{\sigma}}(\vec{y}) = \sum_{i=1}^{S} \sigma_i \left[y_i\right]_2 - \left[\left\lceil \sum_{i=1}^{S} \sigma_i z_i \right\rfloor\right]_2$. Since all z_i's have only l bits of precision, we have seen (in the first lecture) that $\mathsf{Dec}_{\vec{\sigma}}(\vec{y})$ can be expressed as polynomial of degree $\leq 2^l \approx s$.

2 FHE Without Squashing

We next sketch the Gentry-Halevi construction [GH11] for transforming SWHE to FHE without relying on the hardness of the sparse-subset-sum problem.

Background: Elementary Symmetric Polynomials For any field K, the k'th elementary symmetric polynomial (ESP) in n variables is (all operations are over the field):

$$e_k(x_1, \ldots, x_n) = \sum_{\substack{I \subseteq [n] \\ |I|=k}} \prod_{i \in I} x_i$$

Fact 1. *For any $0 \leq k \leq n$ and any set of values $v_1, v_2, \ldots, v_n \in K$, $e_k(v_1, \ldots, v_n)$ is the coefficient of z^{n-k} in the univariate polynomial $P_{\vec{v}} = \prod_{i=0}^{n} (z + v_i)$.*

Fact 2. *$e_i(\vec{v}) = 1$ for all $\vec{v}$ s.t. $\mathsf{HW}(\vec{v}) = i$, and $e_i(\vec{v}) = 0$ for all $\vec{v}$ s.t. $\mathsf{HW}(\vec{v}) < i$.*

Corollary 1. *Over any field K, the following $n+1$ vectors in K^{n+1} are linearly independent: $\left\{ \ (e_0(\vec{v}_i), e_1(\vec{v}_i), \ldots, e_n(\vec{v}_i)) \mid \vec{v}_i = 1^i 0^{n-i}, \ 0 \leq i \leq n \right\}$*

Proof. Consider these vectors as the columns of a matrix. Following from Fact 2, this matrix is upper triangular with 1's on its diagonal. □

Corollary 2. *Let K be any field, and let $S(x_1, \ldots, x_n)$ be any symmetric function in n variables. Then there exist linear coefficients $\alpha_0, \alpha_1, \ldots, \alpha_n \in K$ s.t. for any $\vec{v} \in \{0,1\}^n \subseteq K^n$ it holds that $S(\vec{v}) = \sum_{k=0}^{n} \alpha_k e_k(\vec{v})$.*

Proof. Since S and the e_i's are symmetric, it is enough to prove only for vectors of the form $\vec{v}_i = 1^i 0^{n-i}$ for $0 \leq i \leq n$. By Corollary 1 the vectors $\{(e_0(\vec{v}_i), e_1(\vec{v}_i), \ldots, e_n(\vec{v}_i))\}_{0 \leq i \leq n}$ are independent. Therefore for every vector in K^{n+1}, in particular for the vector $(S(\vec{v}_0), S(\vec{v}_1), \ldots, S(\vec{v}_n))$, there exist coefficients $\alpha_0, \alpha_1, \ldots, \alpha_n \in K$ s.t. for all $0 \leq i \leq n$, $S(\vec{v}_i) = \sum_{k=0}^{n} \alpha_k e_k(\vec{v}_i)$. □

Lemma 1 (Ben-Or)**.** *Let K be a field s.t. $|K| \geq n+1$ and let $S(x_1, \ldots, x_n)$ be any symmetric function in n variables over K. Then there is an $\mathsf{AC3}$ ($\Sigma\Pi\Sigma$) arithmetic circuit C over K s.t. for every $\vec{v} \in \{0,1\}^n \subseteq K^n$, $C(\vec{v}) = S(\vec{v})$.*

Proof. Fix an arbitrary subset $A = \{a_1, \ldots, a_{n+1}\}$ of K of size $n+1$. Note that the polynomial $P_{\vec{v}}(z) = \prod_{i=1}^{n}(z + v_i)$ is of degree n and therefore it can be interpolated from its values on any $n+1$ points in K. In particular, any coefficient of $P_{\vec{v}}(z)$ can be expressed as a linear combination of the $n+1$ values $P_{\vec{v}}(a_j) = \prod_{i=1}^{n}(a_j + v_i)$. Moreover, the interpolation coefficients depend only on A, not on $\vec{v}$. Recall that every $e_k(\vec{v})$ is a coefficient of $P_{\vec{v}}(z)$ and hence can be computed as $e_k(\vec{v}) = \sum_{j=1}^{n+1} \lambda_j P_{\vec{v}}(a_j) = \sum_{j=1}^{n+1} \lambda_j \prod_{i=1}^{n}(a_j + v_i)$. by Corollary 2 we have:

$$S(\vec{v}) = \sum_{k=0}^{n} \alpha_k e_k(\vec{v}) = \sum_{k=0,j=1}^{n,n+1} \alpha_k \lambda_{j,k} \prod_{i=1}^{n}(a_j + v_i)$$

□

Note that the ΣΠΣ circuit that we get has a very special structure: No mater what the function S is, the bottom Σ layer always consists of evaluating the $n(n+1)$ linear functions $(x_i + a_j)$ on the values $x_i = \vec{v}_i$. Note also that given the $n+1$ values $S(\vec{v}_0), \ldots, S(\vec{v}_n)$ we can easily compute the coefficients $\alpha_0, \ldots, \alpha_n$, and since the $\lambda_{j,k}$'s are the interpolation coefficients (which are easy to compute from the A), then we can compute an explicit description of the entire ΣΠΣ circuit.

In what follows, we call such ΣΠΣ circuits A-restricted circuits. Specifically, we require that the bottom Σ layer contain only gates of the form $(x_i + a_j)$ or just (x_i).

Corollary 3. *for any prime p, any function $g : \mathbb{Z}_p \to \mathbb{Z}_p$, and any vector $z_0, \ldots, z_{n-1} \in \mathbb{Z}_p$ of coefficients, denote $m = \max_{i=0}^{n-1} z_i$. If $p \leq mn + 1$ then there is a restricted ΣΠΣ circuit C of size* $\text{poly}(mn)$ *s.t. for any vector $\vec{v} \in \{0,1\}^n \subseteq K^n$ it holds that $C(\vec{v}) = g(\sum_{i=0}^{n-1} z_i v_i)$.*

Proof. Consider the function $S(x_1, \ldots, x_{mn}) = g(x_1 + \cdots + x_{mn})$. This is a symmetric function, so by Lemma 1 it has a restricted ΣΠΣ circuit C' that agrees with g on every input in $\{0,1\}^{mn}$ (since $p \geq mn + 1$). We get the required circuit C simply by replicating each input v_i for z_i times (and padding with 0's as needed). Namely:

$$C(v_0 \cdots v_{n-1}) = C'(v_0^{z_0} 0^{m-z_0}\ v_1^{z_1} 0^{m-z_1}\ \cdots\ v_{n-1}^{z_{n-1}} 0^{m-z_{n-1}})$$

Since C' is restricted, then so is C. Moreover, if we know $g(0), g(1), \ldots, g(mn)$ then we can explicitly compute C' and C. □

Recall our SWHE scheme: $pk = (d, r)$,$sk = w$ and $\mathsf{Eec}_{pk}(m) = [2\vec{e}(r) + m]_d$,$\mathsf{Dec}_{sk}(c) = [[wc]_d]_2$. In what follows we want to work with plaintext space $\mathbb{Z}_p$ rather than $\mathbb{Z}_2$. Now we have: $\mathsf{Eec}_{pk}(m) = [p\vec{e}(r) + m]_d$,$\mathsf{Dec}_{sk}(c) = \left[w^{-1}\left[wc\right]_d\right]_p$. For simplicity, we assume that $w \equiv 1 \mod p$ and also $d \equiv 1 \mod p$ (more on how to achieve this in the homework).

Let $\sigma_{n-1} \cdots \sigma_1 \sigma_0$ be the binary representation of w. We post-process the ciphertext c to get $y_i = \left[2^i c\right]_d$. Then decryption becomes:

$$\mathsf{Dec}_{sk}(c) \equiv \left[\sum_{i=0}^{n-1} \sigma_i y_i\right]_d \equiv \sum_{i=0}^{n-1} \sigma_i y_i - d \cdot \left[\sum_{i=0}^{n-1} \sigma_i \frac{y_i}{d}\right] \equiv \sum_{i=0}^{n-1} \sigma_i \left[y_i\right]_p - \left[\sum_{i=0}^{n-1} \sigma_i \frac{y_i}{d}\right] \pmod p$$

As before, if wc is within $\frac{d}{2n}$ from a multiple of d then it is enough to keep only $l = \lceil log(n) \rceil$ bits of precision for $\frac{y_i}{d}$. Let $z_i \leq 2^l$ be the closest integer to $2^l \cdot \frac{y_i}{d}$ and let $x_i = [y_i]_p$. Then decryption becomes: $\sum_{i=0}^{n-1} \sigma_i x_i - \left\lfloor \sum_{i=0}^{n-1} 2^{-l} \sigma_i z_i \right\rfloor \pmod p$.

Consider the function $g : \mathbb{Z}_p \to \mathbb{Z}_p, g(x) = \left[\left\lceil 2^{-l} x \right\rfloor\right]_p$. By Corollary 3 there is a restricted ΣΠΣ circuit computing $s(\sigma_0 \cdots \sigma_{n-1}) = g(\sum_{i=0}^{n-1} \sigma_i z_i)$ which is just the non-linear part of the decryption formula. The complexity of the circuit is $\text{poly}(n, 2^l) = \text{poly}(n)$.

"Chimeric" - FHE with ElGamal Let $p = 2q+1$ be a safe prime s.t. DDH holds in the group $\mathsf{QR}(p)$ of quadratic residues modulo p. We will use $\mathsf{QR}(p)$ as our plaintext space for ElGamal, and use $\mathbb{Z}_p$ as the plaintext space for the SWHE scheme. Let g be a generator for $\mathsf{QR}(p)$ (e.g. $g = 4$). We first generate keys for ElGamal: $sk = e \in z_q$ and $pk = h = g^{-e} \pmod p \in \mathsf{QR}(p)$. Recall that the encryption of a message $m \in \mathsf{QR}(p)$ is a pair $(g^r, m \cdot h^r) \in \mathsf{QR}(p)^2$.

Next we generate the keys for the SWHE scheme: $pk = (d, r)$, $sk = w$. In addition, we encrypt the bits of e and of w under the SWHE scheme. We choose a subset $A \subseteq \mathsf{QR}(p)$ of size $2|w|^2 + 1$ s.t. for all $a \in A$ it holds that also $a + 1 \in \mathsf{QR}(p)$. For every bit σ_i in the binary representation of w, we encrypt under ElGamal all the elements $\{a + \sigma_i | a \in A\}$. Now we compute the explicit representation of the A-restricted $\Sigma\Pi\Sigma$ circuit C of the form $C(\vec{x}) = \sum_{k=0,j=1}^{n,n+1} \alpha_k \lambda_{j,k} \prod_{i=1}^{n} (a_j + x_i)$ that on inputs in $\{0,1\}^n$ agrees with the function $S(x_1, \ldots, x_n) = \left[\left\lceil 2^{-l} \sum_{i=1}^{n} 2^{-l} x_i z_i \right\rfloor\right]_p$. The SWHE parameters are set so that it can evaluate polynomials of degree up to $2|p|$ with "sufficiently many" terms (e.g., $(2p)^{2|p|}$ terms).

Given a ciphertext c of the SWHE scheme, we show how to compute homomorphic decryption. Post-process c to get $y_i = \left[2^i c\right]_d$ and compute $x_i = [y_i]_p$ and $z_i = \left\lceil 2^l \cdot \frac{y_i}{d} \right\rfloor$. For every bit σ_i in the binary representation of w, let $S_{i,j}$ be the ElGamal encryption of the element $(a_j + \sigma_i) \in \mathsf{QR}(p)$. Use the multiplicative homomorphism to ElGamal to compute for every k, j and ElGamal ciphertext $S_{j,k} = \alpha_k \lambda_{j,k} \prod_{i=1}^{n} S_{i,j}$. Using the bits of e that are encrypted under the SWHE, compute the SWHE ciphertext that encrypt the same values $C_{j,k} = \mathsf{Eec}_w\left(\alpha_k \lambda_{j,k} \prod_{i=1}^{n} (a_j + \sigma_i)\right)$ and add them all to get $C(\sigma_0 \ldots \sigma_{n-1}) = \left[\left\lceil 2^{-l} \sum_{i=0}^{n-1} \sigma_i z_i \right\rfloor\right]_p$ encrypted under the SWHE scheme. Then use the encryption of w under the SWHE to compute also $\sum_{i=0}^{n-1} \sigma_i x_i$, and add everything to compute the homomorphic decryption.

Since we are using the SWHE scheme to homomorphically evaluate the ElGamal decryption circuit, it is left to show how to compute ElGamal decryption using polynomial of degree $|p|$. Let $(y, z) = (g^r, m \cdot g^{-er})$ be a ciphertext, and let $e_{t-1} \cdots e_1 e_0$ be the binary representation of the ElGamal secret key. We post-process the ciphertext by computing $u_i = \left[y^{2^i} - 1\right]_p$ for $0 \le i \le t-1$. Note that $y^{e_i 2^i} = e_i(y^{2^i} - 1) + 1$ and therefore:

$$m = y^e \cdot z = y^{\sum_{i=0}^{t-1} e_i 2^i} \cdot z = z \cdot \prod_{i=0}^{t-1} e_i(u_i + 1)$$

which is a polynomial of degree $|q|$ in the e_i's.

References

[GH11] Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. Cryptology ePrint Archive, Report 2011/279, 2011. `http://eprint.iacr.org/`.