Slowly-Revealed Secure Multi-Party Computation

張珉赫 (장민혁)

December 23, 2024

Introduction

Somewhat homomorphic encryption (SHE) 의 문제점이 뭐였냐 하며는 연산을 거듭함에 따라 암호 체계의 security 를 위해 고의로 추가한 오차가 점차 차오른다는 것이었다. 그래서 연산을 너무 많이 하게 되면 본디 message 가 차오르던 오차에 가려져서 제대로 된 decryption 을 할 수 없게 된다. 이 때문에 fully homomorphic encryption (FHE) 을 위해서는 bootstrapping 이라는 과정이 필요했다.

하지만 오차로 인하여 본디 message 가 가려지는 현상은 to our advantage 역이용할 수 있다: random 값에 의한 masking 이야말로 privacy (내지는 "hiding") 를 달성하는 대표적인 방법이기 때문이다. 아이디어는 비슷한 연산을 거듭함에 따른 "message 증가율" 과 오차의 증가율에 의도적인 차이를 두는 것 – 정확히는 오차의 증가율보다 message 증가율을 크게 하는 것이다. MPC 의 모든 참가자가 circulating ciphertext 에 비슷한 연산을 honestly 혹은 가령 zero-knowledge proof 로써 증명 가능한 방법으로 가한다면, 처음에는 해독이 불가하였던 ciphertext 가 eventually 해독이 가능한 ciphertext 가 되어서 연산 결과가 reveal 된다.

本稿에서는 이러한 SHE 를 바탕으로 하는 limited but secure multi-party computation 을 「정족수 투표」 라는 하나의 구체적이고 몹시 유용한 예를 통해서 살펴본다.

A Model of Quorums

우선, 정족수 투표의 모델을 생각해보자.

정족수란 투표함을 열어보기 위해, 곧 투표가 성사되기 위해 필요한 최소한의 표수이다. 투표에 정족수를 enforce 하는 이유에는 여럿이 있겠다. 가령,

충분한 지지가 필요한 사안에 있어서는 그렇게 할 수 있겠다. 또 한 가지 생각해볼 수 있는 상황은 정족수가 없는 무기명 투표를 하는데 단 두 사람만이 투표를 한 뒤에 개표를 하게 되었다고 하자. 투표 결과를 받아본 뒤 각 party 는 (스스로가 던진 표를 정확하게 기억하고 있다는 전제 하에) 상대방의 표를 정확히 추정할 수 있다. 이것은 무기명 투표라는 목표에 모순된다.

따라서 정족수 투표에 있어서 우리는 정족수 M 이 설정되고 기권이 따로 없는 상황에서 찬성표 (“yeas”) 가 Y 개, 반대표 (“nays”) 가 N 개일 때 $Y + N \geq M$ 일 때만 투표 결과가 visible 하기를 바란다.

A Voting Protocol with a Quorum

그렇다면 정족수 투표 프로토콜을 건설하기 위해 대표적인 SHE scheme 인 BFV Scheme 에서 출발하여보자. Suppose an instance of a BFV Scheme (n, p, q, σ, χ). 어떤 message $m \in R_p = \mathbb{Z}_p[X]/(X^n + 1)$ 을 public key $\text{pk} = (b, a)$ where $b = -as + e$ and $a \leftarrow U(R_q)$ and $e \leftarrow D_\sigma$ (a discrete Gaussian distribution of width σ over R) 로 encrypt 해서 얻은 $\mathbf{c} = (c_0, c_1) = r\,(b, a) + (e_0, e_1) + (\frac{q}{p}m, 0) \in R_q^2$ where $r \leftarrow \chi$ and $e_0, e_1 \leftarrow D_\sigma$ 를 다른 어떤 ciphertext 에 더할 때마다 “에러” 에 해당되는 부분 – 곧 성공적인 decryption 에 의해서는 “round away” 될 부분이 얼마 만큼 증가하는가를 살펴볼 필요가 있다:

$$\mu = c_0 + c_1 s = \frac{q}{p}m + re + e_0 + e_1 s \mod q.$$

먼저, if $e \leftarrow D_\sigma$, there must be some δ such that for $1 - \epsilon$ of the time, say 99% of the time, $e \in \mathbb{Z}_\delta[X]/(X^n + 1)$, i.e. $\|e\|_{\text{sup}} \leq \delta/2$. Therefore, for $1 - \epsilon$ of the time,

$$\begin{aligned}\|re + e_0 + e_1 s\|_{\text{sup}} &\leq \|re\|_{\text{sup}} + \|e_0\|_{\text{sup}} + \|e_1 s\|_{\text{sup}} \\ &\leq n\|\chi\|_{\text{sup}}\|e\|_{\text{sup}} + \|e_0\|_{\text{sup}} + n\|e_1\|_{\text{sup}}\|\chi\|_{\text{sup}} \\ &\leq \big(2n\|\chi\|_{\text{sup}} + 1\big)\,\delta/2,\end{aligned}$$

i.e.,

$$re + e_0 + e_1 s \in \mathbb{Z}_{(2n\|\chi\|_{\text{sup}}+1)\delta}[X]/(X^n + 1).$$

Now suppose we add M such ciphertexts $\mathbf{c}_i = (c_{i0}, c_{i1}) = r_i\,(b, a) + (e_{i0}, e_{i1}) + (\frac{q}{p}m_i, 0) \in R_q^2\,(i = 1, \ldots, M)$ together. The resulting ciphertext

$$\mathbf{c} = \sum_{i=1}^{M} \mathbf{c}_i = \left(\sum_{i=1}^{M} c_{i0}, \sum_{i=1}^{M} c_{i1}\right) = \sum_{i=1}^{M} r_i\,(b, a) + \left(\sum_{i=1}^{M} e_{i0}, \sum_{i=1}^{M} e_{i1}\right) + \left(\frac{q}{p}\sum_{i=1}^{M} m_i, 0\right);$$

$$\mu = \sum_{i=1}^{M} c_{i0} + \sum_{i=1}^{M} c_{i1}s = \frac{q}{p}\sum_{i=1}^{M} m_i + \sum_{i=1}^{M} r_i e + \sum_{i=1}^{M} e_{i0} + \sum_{i=1}^{M} e_{i1}s \mod q;$$

and thus for $1-\epsilon$ of the time,

$$\begin{aligned}\|\sum_{i=1}^{M} r_i e + \sum_{i=1}^{M} e_{i0} + \sum_{i=1}^{M} e_{i1} s\|_{\text{sup}} &\leq Mn\|\chi\|_{\text{sup}}\delta/2 + M\delta/2 + Mn\left(\delta/2\right)\|\chi\|_{\text{sup}} \\ &\leq M\left(2n\|\chi\|_{\text{sup}}+1\right)\delta/2,\end{aligned}$$

or

$$\sum_{i=1}^{M} r_i e + \sum_{i=1}^{M} e_{i0} + \sum_{i=1}^{M} e_{i1} s \in \mathbb{Z}_{M(2n\|\chi\|_{\text{sup}}+1)\delta}[X]/(X^n+1).$$

Thus, we may safely conclude that the absolute value of each coefficient of the error polynomial is bounded by a linear function of M with slope $\left(2n\|\chi\|_{\text{sup}}+1\right)\delta/2$. We may as well as have started with a δ such that for $1-\epsilon$ of the time, $\|e\|_{\text{sup}} \leq \frac{1}{2n\|\chi\|_{\text{sup}}+1}\delta$. 이때

$$\|\sum_{i=1}^{M} r_i e + \sum_{i=1}^{M} e_{i0} + \sum_{i=1}^{M} e_{i1} s\|_{\text{sup}} \leq M\delta.$$

The protocol begins with a call for a vote. Parties who wish to learn about the outcome of the vote each instantiate a BFV Scheme and prepare an initial ciphertext $\mathbf{c}_0$ by encrypting some initial message $m_0 \in \mathbb{Z}_p$ as follows:

$$\mathbf{c}_0 = (c_{00}, c_{01}) = r_0\,(b, a) + (e_{00}, e_{01}) + (m_0, 0).$$

It is important that m_0 be kept secret; some other conditions on m_0 will be discussed later. Note that the protocol can be initiated and run independently by each participant (although practically, some aspects would be better coordinated); therefore our description will proceed focusing on one initiator' s ciphertext.

Notice how $\mathbf{c}_0$ is not a ciphertext of m_0 per se, as a proper ciphertext of m_0 should incorporate as its final term $\left(\frac{q}{p}m_0, 0\right)$ and not $(m_0, 0)$. Readers may further notice that this sort of encryption can also be found in the CKKS scheme. In fact, this CKKS-style encryption is the only form of encryption used throughout the protocol, so we may as well as call this "encryption" .

The ciphertext is then passed around. All information about the BFV scheme is sent with it, including the secret key. Note that all votes cast are hidden by the initial message m_0: all parties oblivious to m_0 cannot possibly remove m_0 from the ciphertext to find out the current tally. Thus it is important that the ciphertext does not return to the initiator (until the very end) once other voters start accumulating votes on it. Each party encrypts their vote: $m_i = \gamma + \zeta$ for yea or $m_i = \gamma$ for nay and evaluates the addition of their encrypted vote with the circulating ciphertext.

Once a total of M votes have been cast, the circulating ciphertext may be returned to the initiator. We want a situation where, following the model of quorum voting described above, the ciphertext the initiator receives is decipherable if $Y + N \geq M$.

An ordinary BFV ciphertext is decipherable if the error term is bounded by $\frac{1}{2}\frac{q}{p}$. Essentially, the ciphertext is decipherable if the plaintext portion of $\mu = c_0 + c_1 s$ is separable from its error portion at the q/p-th place. We may extend this condition of decipherability to any boundary: a ciphertext need only be separable, wherever the boundary may be. Whether this is the case for the current circulating ciphertext can be checked by the voters; if the μ computed from the ciphertext' s upper bits are stable (do not change across additional votes) this indicates the error has "lost out" and cannot mangle the overall result of the vote.

정족수를 만족시킨 상태의 circulating ciphertext 를 decrypt 한다고 하자. μ 를 살펴보면, message (payload) 는

$$m_0 + Y(\gamma + \zeta) + N\gamma = m_0 + M\gamma + Y\zeta$$

이고 error 는

$$\|\sum_{i=1}^{M} r_i e + \sum_{i=1}^{M} e_{i0} + \sum_{i=1}^{M} e_{i1} s\|_{\text{sup}} \leq M\delta$$

를 만족시키기 때문에 we may formulate the decipherability condition as follows:

$$m_0 + M\gamma + Y\zeta \geq 2^B.$$

Here, B is the positive integer such that the B-th bit of the binary representation of μ is the most significant bit the error occupies $1 - \epsilon$ of the time, i.e.,

$$B = \lfloor \log_2 M\delta \rfloor.$$

Solving for Y,

$$Y \geq \frac{2^B - m_0 - M\gamma}{\zeta} \geq \frac{M(\delta - \gamma) - m_0 - 1}{\zeta}.$$

The ciphertext is decipherable only if the number of yeas Y satisfies the above inequality. Therefore, if we conveniently set $P := \frac{M(\delta-\gamma)-m_0-1}{\zeta}$ to the number of yeas this vote requires to pass, we have a voting system which only decrypts with sufficient votes (specified roughly by M) and when there are enough yeas (specified roughly by P).

For example, given $\delta = 1023$, $M = 300$, $P = 151$, we can obtain parameters for the protocol, i.e., suitable γ and ζ values for each voter to cast, and a suitable range for m_0 for the initiator to encrypt.

So, upon receiving a quorum-satisfying ciphertext, the initiator can evaluate the addition of $-m_0$ to obtain a decipherable ciphertext. They can then round away the error part to obtain an exact-to-the-$(B+1)$-th-bit count of the number of yeas Y.

Notes on the Protocol

Note that the error is merely bounded by $M\delta$, so it is possible that the ciphertext' s error is insufficient to hide the votes until the quorum has been met. Therefore, it may be desirable to sample the errors from a discrete Gaussian distribution with a positive mean, while reducing the width σ.

In this protocol, both the public and secret keys were circulated. Therefore, we may as well as use private-key RLWE encryption and decryption, and share only one key.

It is crucial that m_0 be kept secret from the voters and the circulating ciphertext be kept secret from the initiator during the accumulation of votes. Or else, either side can decrypt the ciphertext. Let us call this the isolation condition. The isolation condition can be rather easily be enforced by (literally) circulating the ciphertext according to some agreed-upon order, as with some other voting protocols.

Finally, for the integrity of each vote, it is important that each party honestly casts either $\gamma + \zeta$ or γ. This can be enforced perhaps with a ledger of non-interactive zero-knowledge proofs of the fact from each voter. This ledger need not be hidden from the initiator – in this respect, the initiator may be regarded as a trusted vote-tallying center.

Security against Semi-honest Participants

The protocol, with one semi-honest initiator and at least M semi-honest voters, is secure, given the adversary does not control both an initiator and a voter (i.e., does not violate the isolation condition). If the adversary controls a lone initiator, the a simulator can be constructed by simply following the protocol with the outcome of the vote: add Y $(\gamma + \zeta)$s and N γs, etc. In the case that the adversary controls any number of voters, the circulating ciphertext cannot be deciphered, since no voter has knowledge of m_0.

Further Research

The isolation condition is crucial for the protocol to have any meaning. It could be said that the protocol is merely a HE-version of the voting protocol discussed in class. However, the fact that the initial ciphertext $\mathrm{Enc}_{\mathrm{pk}}(m_0)$ is a ciphertext of a HE scheme gives us a chance to construct a scheme that does not rely on this condition.

What remains is to find a way to share an m_0 so that no participant knows its value until the parties agree that the upper bits are stable (free of errors), that it' s time to open the ballot box, and thus share their shares of m_0. This would be easy using an MPC protocol, but the challenge was (mostly) to construct an MPC protocol entirely from an HE scheme. This is the work that remains.