Appendices:

A. Error Estimation

For some given $\mathrm{p}, \mathrm{q}, \mathrm{d} \in \mathrm{N}$, let $\mathrm{R_p} = \mathrm{Z_p}[\mathrm{X}]/(\mathrm{X^d}+1)$, $\mathrm{R_q} = \mathrm{Z_q}[\mathrm{X}]/(\mathrm{X^d}+1)$, we'll use $\mathrm{R_p}$ to denote the plaintext space of our somewhat homomorphic encryption scheme, and $(\mathrm{R_q})^2$ to denote the ciphertext space. Given two ciphertexts $c = (c_0, c_1), c' = (c_0', c_1') \in (R_q)^2$, we may suppose they bring the messages $m, m' \in R_p$, and errors $e, e' \in R_q$. For example, we can write

$$c_0 = -c_1 \cdot sk + m \cdot \rho + e$$
$$c_0' = -c_1' \cdot sk + m' \cdot \rho + e'$$

where $\rho = \mathrm{q} \cdot \mathrm{p}^{-1}$. Notice that ρ can be regarded as the "tolerance" of error, that is, a ciphertext c can be decrypted correctly when ever $\|e\|_\infty < 2^{-1}$.

We begin with four basic operations to estimate the magnitude of errors: *Encryption, Addition*, and *Multiplication without Key Switching, and Key Switching*.

1. Public Key Encryption

Since

$$BFV.Enc(m) = (r \cdot pk_0 + e_0 + m \cdot \rho, r \cdot pk_1 + e_1)$$

and

$$pk_1 \cdot sk + pk_0 = e_{pub},$$

we have

$$(r \cdot pk_1 + e_1) \cdot sk + r \cdot pk_0 + e_0 + m \cdot \rho = m \cdot \rho + r \cdot e_{pub} + e_0 + e_1 \cdot sk.$$

The new error is

$$r \cdot e_{pub} + e_0 + e_1 \cdot sk$$

2. Addition

Since

$$BFV.Add(c, c') = (c_0 + c'_0, c_1 + c'_1)$$

and

$$c_0 + c'_0 = -(c_1 + c'_1) \cdot sk + (m + m') \cdot \rho + e + e'.$$

The new message is $m + m'$, and the new error is $e + e'$.

3. Multiplication

Notice that

$$BFV.MulWithoutKeySwitch(c, c') = (c_0^{mul'}, c_1^{mul'}, c_2^{mul'}) = (\lceil \rho^{-1} \cdot c_0 c'_0 \rfloor, \lceil \rho^{-1} \cdot (c_0 c'_1 + c_1 c'_0) \rfloor, \lceil \rho^{-1} \cdot c_1 c'_1 \rfloor),$$

and

$$\rho^{-1} \cdot c_0 c'_0 + \rho^{-1} \cdot (c_0 c'_1 + c_1 c'_0) \cdot sk + \rho^{-1} \cdot c_1 c'_1 \cdot (sk)^2 = \rho^{-1} \cdot (c_0 + c_1 \cdot sk)(c'_0 + c'_1 \cdot sk) = \rho^{-1} \cdot (m \cdot \rho + e)(m' \cdot \rho + e') = m \cdot m' \cdot \rho + (me' + m'e) + \rho^{-1} \cdot (e + e').$$

Since $\|\lceil x \rfloor - x\|_\infty \leq 2^{-1}$ for all $x \in R_q$, we have

$$c_0^{mul'} + c_1^{mul'} \cdot sk + c_2^{mul'} \cdot (sk)^2 = m \cdot m' \cdot \rho + (me' + m'e) + \rho^{-1} \cdot (e + e') + \varepsilon_0 + \varepsilon_1 \cdot sk + \varepsilon_2 \cdot (sk)^2,$$

where $\|\varepsilon_i\|_\infty \leq 2^{-1}$. (We'll ignore the errors generated by ε_i and $\rho^{-1} \cdot (e + e')$ in the following estimation.)

4. Key Switching

Let

$d'[iT : iT + T - 1]$ denote the bits between the $iT - th$ bit and the $(iT + T - 1) -$ th bit of d', then

$$BFV.\mathrm{KeySwitch}(d') = \sum_{i=0}^{\lfloor log\ q\ /\ T \rfloor} d'^{[iT:iT+T-1]} \cdot (b_{eva_{iT}}, a_{eva_{iT}}).$$

For some given $sk' \in R_q$, the goal of re-linearization is to create $d = (d_0, d_1)$ such that

$$d' \cdot sk' \approx d_0 + d_1 \cdot sk.$$

To do this, the one who holds the secret key hides the key in the RLWE samples (a_{eva_i}, b_{eva_i}), say

$$b_{eva_i} + a_{eva_i} \cdot sk = 2^i sk' + e_{eva_i}$$

and

$$d_0 + d_1 \cdot sk = \sum_{i=0}^{\lfloor log\ q\ /\ T \rfloor} d'^{[iT:iT+T-1]} \cdot (2^{iT} sk' + e_{eva_{iT}}) = d'sk' + \sum_{i=0}^{\lfloor log\ q\ /\ T \rfloor} d'[iT : iT + T - 1] \cdot e_{eva_{iT}}.$$

Then, the term $\sum_{i=0}^{\lfloor log\ q\ /\ T \rfloor} d'[iT : iT + T - 1] \cdot e_{eva_{iT}}$ is the additional error.

We conclude that the ciphertext generated by $\mathrm{BFV.Add}$ brings the error $e + e'$, the ciphertexts generated by $\mathrm{BFV.Mul}$ brings the error $\mathrm{m}e' + m'e + e_{switch}$, and $\mathrm{BFV.Rotate}$ brings the error e'_{switch}.

To claim the correctness of the auction result, we need to claim that each entry of the resulting ciphertext's error is less than $\rho/2$. Instead of calculating the direct upper bound of error, we calculate the variance (of the error) for estimating the probability of correct decryption.

To generate the RLWE samples, the bidder or auctioneer needs to sample errors from a bounded discrete Gaussian distribution (we use χ to represent this distribution):

1. The auctioneers sample Gaussian errors e_{pub} *and* e_ϕ for generating the public key and evaluation keys. Thus, we may write
$$pk_0 = -pk_1 \cdot sk + e_{pub}$$
$$b_\phi = -a_\phi \cdot sk + m_\phi \cdot \rho + e_\phi$$
2. Each bidder needs to sample Gaussian errors $e_{bidder_i,j}$ to encrypt their message, say
$$c_0^{bidder_i} = r_i \cdot pk_0 + e_{bidder_i,0}$$
$$c_1^{bidder_i} = r_i \cdot pk_1 + e_{bidder_i,1}.$$

When estimating the variance, we suppose all other elements are fixed, and each (fresh) error, say e_α, is sampled from the Gaussian distribution $\mathcal{E}_\alpha$.

For a ciphertext c with $c_0 = -c_1 \cdot sk + m \cdot \rho + e$, the message can be correctly decrypted (i.e., $BFV.Dec(c, sk) = \mathrm{m}$) if $|(e)_i| < 2^{-1} \cdot \rho$ for each i. Suppose the distribution of the error in the resulting ciphertext is $\mathcal{E}_{\mathrm{result}}$. Then the probability of correct decryption is at least $1 - \sum_{\mathrm{i}} \mathrm{Prob}\{|(\mathcal{E}_{\mathrm{result}})_{\mathrm{i}}| > 2^{-1} \cdot \rho\}$.

To simplify the estimation procedure, we'll always assume the errors in two different ciphertexts are independent (In fact,

they are not. We believe that this simplification would not significantly affect the result). Another simplification is we ignore the errors generated by ε_i. Notice that $\|\varepsilon_0 + \varepsilon_1 \cdot sk + \varepsilon_2 \cdot (sk)^2\|_\infty \le 2^{-1}(1 + d + d^2)$ (if $\|sk\|_\infty \le 1$). In our experiment setting, $\mathrm{d} = 8192$, this error is bounded by 2^{26}, which can barely affect the result.

By some acknowledge in probability, if $\mathcal{E}_1, \mathcal{E}_2$ are discrete Gaussian distributions with variances $(\sigma_1)^2, (\sigma_2)^2$, then $\mathcal{E}_3 = a_1\mathcal{E}_1 + a_2\mathcal{E}_2$ is also a Gaussian distribution, which has variance $(a_1\sigma_1)^2 + (a_2\sigma_2)^2$. With this principle and the assumption above, we can estimate the variance of our errors.

In a practical setting, $\mathrm{d} = 8192$, the errors are sampled from χ^d, where χ is Gaussian with variance less than 10. We'll use $p = 17$. Notice that we only use the messages in the subring $\mathrm{M} = \{z \in R_p \mid z = \sum_i z_i \cdot X^{1024 \cdot i}\}$, where $\mathrm{M} \cong (\mathrm{Z}_{17})^8$. For each $\mathrm{m} \in \mathrm{M}$, we treat each entry of m as an element in $[-8, 8]$, so we always have $\|m\|_\infty \le 8$. By the calculation above, all errors are Gaussian, no matter how many operations they passed through.

Hence, we can calculate the variance of the error of the following four basic operations:

1. $\mathrm{BFV.Enc}(m)$

Input a message m, output a ciphertext brings the message

$$r \cdot e_{pub} + e_0 + e_1 \cdot sk\,,$$

which has the distribution of

$$\mathrm{r} \cdot \mathcal{E}_{\mathrm{pub}} + \mathcal{E}_0 + \mathcal{E}_1 \cdot \mathrm{sk}\,.$$

Recall that the variances of $\left(\mathcal{E}_{\mathrm{pub}}\right)_i, (\mathcal{E}_0)_\mathrm{i}, and\ (\mathcal{E}_1)_\mathrm{i}$ are 10, so

$$\begin{aligned}\left(r \cdot \mathcal{E}_{pub} + \mathcal{E}_0 + \mathcal{E}_1 \cdot sk\right)_i &= (\mathcal{E}_0)_i + \sum_{0 \le k \le i} (r)_k \left(\mathcal{E}_{pub}\right)_{i-k} \\ &+ (sk)_k (\mathcal{E}_1)_{i-k} \\ &- \sum_{i<k<8192} (r)_k \left(\mathcal{E}_{pub}\right)_{8192+i-k} \\ &+ (sk)_k (\mathcal{E}_1)_{8192+i-k}\,.\end{aligned}$$

The distribution of the i-th entry of the new error can be bounded by

$$\begin{aligned}Var((\mathcal{E}_0)_i) &+ \max_k\{Var\left(\left(\mathcal{E}_{pub}\right)_k\right)\} \cdot \sum_{0 \le k \le i} ((r)_k)^2 \\ &+ \max_k\{Var((\mathcal{E}_1)_k)\} \cdot \sum_{0 \le k \le i} ((sk)_k)^2\,,\end{aligned}$$

where $(r)_k$ and $(sk)_k$ are sampled from $\{0,1\}$, so we may conclude that the new distribution (of each entry) has variance less than $10 \cdot 16385 \le 2^{18}$.

2. $\mathrm{BFV.Add}(c, c')$

Suppose the distributions of input errors are Gaussian, say $\mathcal{E}$ and $\mathcal{E}'$. We further suppose $\mathrm{Var}((\mathcal{E})_i) \le \sigma^2$ and $\mathrm{Var}((\mathcal{E}')_i) \le \sigma'^2$. By the discussion above, the output error has distribution $\mathcal{E} + \mathcal{E}'$, and $(\mathcal{E} + \mathcal{E}')_\mathrm{i} = (\mathcal{E})_\mathrm{i} + (\mathcal{E}')_\mathrm{i}$ also follows a Gaussian, which has variance less than $\sigma^2 + \sigma'^2 \le 2 \cdot max\{\sigma^2, \sigma'^2\}$.

3. $\mathrm{BFV.MulWithoutKeySwitch}(\mathrm{c}, \mathrm{c}')$

Suppose the distributions of input errors are $\mathcal{E}$ and $\mathcal{E}'$, then the output error would be $m'(\mathcal{E})^\mathrm{n} + \mathrm{m}(\mathcal{E}')^n$, is also a Gaussian. Suppose $\mathrm{Var}((\mathcal{E})_i) \le \sigma^2$ and $\mathrm{Var}((\mathcal{E}')_i) \le \sigma'^2$, then $\mathrm{m}'(\mathcal{E})^\mathrm{n} + \mathrm{m}(\mathcal{E}')^n$ has variance less than $\sigma'^2 \sum_i (m_i)^2 + \sigma^2 \sum_i (m_i')^2$. Since $\mathrm{m}, \mathrm{m}' \in \{z \in R_p \mid z = \sum_i z_i \cdot X^{1024 \cdot i}\}$, so $\sum_i (m_i)^2, \sum_i (m_i')^2 \le 8 \cdot 8^2$, and the new error should have variance less than

$$8^3 \cdot (\sigma^2 + \sigma'^2) \le 2^{10} \cdot max\{\sigma^2, \sigma'^2\}$$

4. $\mathrm{BFV.KeySwitch}(\mathrm{d}')$

Since the additional error is $\sum_{i=0}^{\lfloor log_2 q/T \rfloor} d'[iT : iT + T - 1] \cdot \mathcal{E}_{eva_{iT}}$, which has variance $\sum_{i=0}^{\lfloor log_2 q/T \rfloor} \sum_{0 \le k < 8192} ((d'[iT : iT + T - 1])_k)^2 \cdot 10$. Also, since $(d'[iT : iT + T - 1])_k$ is a T-bit integer, $((d'[iT : iT + T - 1])_k)^2 \le 2^{2T}$, so the variance is less than $10 \cdot 2^{2\mathrm{T}} \cdot (\lfloor log_2 q/T \rfloor + 1)$. Let $\mathrm{T} = 20$, $\lfloor log_2 q/T \rfloor = 11$, we have $10 \cdot 2^{2\mathrm{T}} \cdot \lfloor log_2 q/T \rfloor \le 2^{47}$.

We'll use these results to estimate the variance of the error of the auction result. If the standard error is less than $13^{-1}(\rho/2)$, then the probability of error decryption is less than 2^{-128}.

In the 5-bidder SHE-based secure auction protocol, we also use $d = 8192$, $log_2 q = 218$, $p = 17$, and $M = \{z \in R_p \mid z = \sum_i z_i \cdot X^{1024 \cdot i}\}$. We need to compromise the standard error of auction result less than $13^{-1}(\rho/2)$, so the failure rate of our protocol is less than 2^{-128}. We often put the error and tolerance in log scale, for example, $13^{-1}(\rho/2) \approx 2^{210}$, so we say the tolerance of standard error is about 209-bit, and the tolerance of variance is $418 - \mathrm{bit}$. Variance of the output error of Addition is at most 2 times than the input, so we say this operation consumes 1-bit tolerance (of variance). Similarly, Multiplication Without Key Switching consumes 10 -bit tolerance. Encryption and Key Switching only affect the noise by adding a Gaussian with variance less than 2^{18} and 2^{47} respectively.

Without the detail of computation, each swap consumes 85-bit, and the depth of the protocol is exactly 4 swaps, and 47-bit for Key Switching and Encryption. Since $47 + 85 \cdot 4 = 387 < 418$, our protocol could be decrypted correctly with probability higher than $1 - 2^{-128}$.