Appendix for ENSEI: Efficient Secure Inference via Frequency-Domain Homomorphic Convolution for Privacy-Preserving Visual Recognition

A. Appendix: Correctness for Eq. (17) to Eq. (19)

For some input $\hat{\mathbf{u}}$, weight vector $\hat{\mathbf{w}}$, secret share $\hat{\mathbf{s}}_B$, HSS modulus p_{A}, NTT modulus p_{N}, we have

$$(\mathsf{INTT}(\hat{\mathbf{u}} \circ \hat{\mathbf{w}} - \hat{\mathbf{s}}_B) \bmod p_{\mathrm{A}})_k \tag{1}$$

$$= \sum_{0}^{n_f-1} ((\hat{u}_i \cdot \hat{w}_i - \hat{s}_{B,i}) \bmod p_{\mathrm{A}})\omega^{ik} \bmod p_{\mathrm{N}} \tag{2}$$

$$= \Big(\sum_{0}^{n_f-1} ((\hat{u}_i \cdot \hat{w}_i - \hat{s}_{B,i}) \bmod p_{\mathrm{A}})\omega^{ik} \bmod p_{\mathrm{N}} \Big) \bmod p_{\mathrm{A}} \tag{3}$$

Assuming $p_{\mathrm{A}} \geq p_{\mathrm{N}}$, we know that for any x, $x \bmod p_{\mathrm{A}} \bmod p_{\mathrm{N}} \equiv x \bmod p_{\mathrm{N}}$.

$$= \Big(\sum_{0}^{n_f-1} ((\hat{u}_i \cdot \hat{w}_i) \bmod p_{\mathrm{A}} - (\hat{s}_{B,i}) \bmod p_{\mathrm{A}})\omega^{ik} \bmod p_{\mathrm{N}} \Big) \bmod p_{\mathrm{A}} \tag{4}$$

$$= \Big(\sum_{0}^{n_f-1} ((\hat{u}_i \cdot \hat{w}_i) \bmod p_{\mathrm{N}} - (\hat{s}_{B,i}) \bmod p_{\mathrm{N}})\omega^{ik} \bmod p_{\mathrm{N}} \Big) \bmod p_{\mathrm{A}} \tag{5}$$

$$= \Big(\sum_{0}^{n_f-1} (\hat{u}_i \cdot \hat{w}_i \bmod p_{\mathrm{N}})\omega^{ik} \bmod p_{\mathrm{N}} \Big) \bmod p_{\mathrm{N}} - \Big(\sum_{0}^{n_f-1} (\hat{s}_{B,i} \bmod p_{\mathrm{N}})\omega^{ik} \bmod p_{\mathrm{N}} \Big) \bmod p_{\mathrm{N}} \tag{6}$$

$$= (\mathsf{INTT}(\hat{\mathbf{u}} \circ \hat{\mathbf{w}}) - \mathsf{INTT}(\hat{\mathbf{s}}_B)) \bmod p_{\mathrm{A}} \tag{7}$$

$$= ((\mathbf{u} * \mathbf{w} - \mathbf{s}_B) \bmod p_{\mathrm{N}}) \bmod p_{\mathrm{A}}, \tag{8}$$

To remove the additive secret sharing, observe that

$$((\mathbf{u} * \mathbf{w} - \mathbf{s}_B) \bmod p_{\mathrm{N}} \bmod p_{\mathrm{A}} + \mathbf{s}_B) \bmod p_{\mathrm{N}} \tag{9}$$

$$= \mathbf{u} * \mathbf{w} \bmod p_{\mathrm{N}} - \mathbf{s}_B \bmod p_{\mathrm{N}} + \mathbf{s}_B \bmod p_{\mathrm{N}} \tag{10}$$

$$= \mathbf{u} * \mathbf{w} \bmod p_{\mathrm{N}}, \tag{11}$$

and this addition can be computed homomorphically using any additive homomorphic encryption scheme.

B. Appendix: Neural Architectures

1. *Convolution*: input image $3 \times 32 \times 32$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times32\times32} \leftarrow \sum \mathbb{R}^{3\times32\times32} * \mathbb{R}^{64\times3\times3}$ + ReLU.
2. *Convolution*: input image $64 \times 32 \times 32$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times32\times32} \leftarrow \mathbb{R}^{64\times32\times32} * \mathbb{R}^{64\times3\times3}$ + ReLU.
3. *Average Pooling*: Outputs $\mathbb{R}^{64\times16\times16}$.
4. *Convolution*: input image $64 \times 16 \times 16$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times16\times16} \leftarrow \mathbb{R}^{64\times16\times16} * \mathbb{R}^{64\times3\times3}$ +ReLU.
5. *Convolution*: same as 6).
6. *Average Pooling*: Outputs $\mathbb{R}^{64\times8\times8}$
7. *Convolution*: input image $64 \times 8 \times 8$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times8\times8} \leftarrow \mathbb{R}^{64\times8\times8} * \mathbb{R}^{64\times3\times3}$ + ReLU.
8. *Convolution*: input image $64 \times 8 \times 8$, weight matrix $64 \times 1 \times 1$, number of output channels 64: $\mathbb{R}^{64\times8\times8} \leftarrow \mathbb{R}^{64\times8\times8} * \mathbb{R}^{64\times1\times1}$ + ReLU.
9. *Fully Connected*: Outputs the classification result $\mathbb{R}^{10\times1} \leftarrow \mathbb{R}^{10\times1024} \cdot \mathbb{R}^{1024\times1}$

Figure 1. The neural architecture from Fig. 13 in [21]

C. Appendix: ENSEI based on FFT

Please refer to Fig. 7.

D. Security of ENSEI

Proposition 1. *If there exists an efficient algorithm $\mathcal{A}$ that learns Bob's model in ENSEI with non-negligible probability, then there also exists an efficient algorithm for that of Gazelle.*

1. *Convolution*: input image $3 \times 32 \times 32$, weight matrix $32 \times 3 \times 3$, number of output channels 32: $\mathbb{R}^{32\times32\times32} \leftarrow \sum \mathbb{R}^{3\times32\times32} * \mathbb{R}^{32\times3\times3}$ + ReLU.
2. *Convolution*: input image $32 \times 32 \times 32$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times32\times32} \leftarrow \mathbb{R}^{32\times32\times32} * \mathbb{R}^{64\times3\times3}$ + Square Activation.
3. *Average Pooling*: Outputs $\mathbb{R}^{64\times16\times16}$.
4. *Convolution*: input image $64 \times 16 \times 16$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times16\times16} \leftarrow \mathbb{R}^{64\times16\times16} * \mathbb{R}^{64\times3\times3}$ + ReLU.
5. *Convolution*: input image $64 \times 16 \times 16$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times16\times16} \leftarrow \mathbb{R}^{64\times16\times16} * \mathbb{R}^{64\times3\times3}$ + Square Activation.
6. *Average Pooling*: Outputs $\mathbb{R}^{64\times8\times8}$
7. *Convolution*: input image $64 \times 8 \times 8$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times8\times8} \leftarrow \mathbb{R}^{64\times8\times8} * \mathbb{R}^{64\times3\times3}$ + ReLU.
8. *Convolution*: input image $64 \times 8 \times 8$, weight matrix $64 \times 1 \times 1$, number of output channels 64: $\mathbb{R}^{64\times8\times8} \leftarrow \mathbb{R}^{64\times8\times8} * \mathbb{R}^{64\times1\times1}$ + ReLU.
9. *Fully Connected*: Outputs the classification result $\mathbb{R}^{10\times1} \leftarrow \mathbb{R}^{10\times1024} \cdot \mathbb{R}^{1024\times1}$

Figure 2. The modified neural architecture from Fig. 13 in [21], where the number of channels in the first convolution layer is reduced by half, and the second and fifth activation function is replaced with square activation.

Proof. We prove Proposition 1 by construction. Since we assumed that $\mathcal{A}$ can obtain the model of Bob given all of the information of Alice, we notice that the same computations are carried out in the time domain in the Gazelle protocol. Therefore, upon some initial image inputs, Alice runs the following procedures.

1. Give Bob the time domain response and execute the Gazelle protocol.
2. For all the inputs to Bob and outputs from Bob, transform the (either plaintext of secret-shared) results into the frequency domain.
3. After the execution of the Gazelle protocol, feed all time- and frequency-domain input and output results to $\mathcal{A}$. Since $\mathcal{A}$ can efficiently solve for Bob's model given all the frequency-domain responses, Alice obtains the same model with the Gazelle protocol.

□

1. *Convolution*: input image $3 \times 32 \times 32$, weight matrix $16 \times 3 \times 3$, number of output channels 16: $\mathbb{R}^{16\times32\times32} \leftarrow \sum \mathbb{R}^{3\times32\times32} * \mathbb{R}^{16\times3\times3}$ + Batch Normalization + Binary Activation.
2. *Convolution*: input image $16 \times 32 \times 32$, weight matrix $16 \times 3 \times 3$, number of output channels 16: $\mathbb{R}^{16\times32\times32} \leftarrow \sum \mathbb{R}^{16\times32\times32} * \mathbb{R}^{16\times3\times3}$ + Batch Normalization + Binary Activation.
3. Same as Layer 2.
4. *Average Pooling*: Outputs $\mathbb{R}^{16\times16\times16}$.
5. *Convolution*: input image $16 \times 16 \times 16$, weight matrix $32 \times 3 \times 3$, number of output channels 32: $\mathbb{R}^{32\times16\times16} \leftarrow \sum \mathbb{R}^{16\times16\times16} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
6. *Convolution*: input image $32 \times 16 \times 16$, weight matrix $32 \times 3 \times 3$, number of output channels 32: $\mathbb{R}^{32\times16\times16} \leftarrow \sum \mathbb{R}^{16\times16\times16} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
7. Same as Layer 6.
8. *Average Pooling*: Outputs $\mathbb{R}^{32\times8\times8}$
9. *Convolution*: input image $32 \times 8 \times 8$, weight matrix $48 \times 3 \times 3$, number of output channels 48: $\mathbb{R}^{48\times6\times6} \leftarrow \sum \mathbb{R}^{32\times8\times8} * \mathbb{R}^{48\times3\times3}$ + Batch Normalization + Binary Activation.
10. *Convolution*: input image $48 \times 6 \times 6$, weight matrix $48 \times 3 \times 3$, number of output channels 48: $\mathbb{R}^{48\times4\times4} \leftarrow \sum \mathbb{R}^{48\times6\times6} * \mathbb{R}^{48\times3\times3}$ + Batch Normalization + Binary Activation.
11. *Convolution*: input image $48 \times 4 \times 4$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times2\times2} \leftarrow \sum \mathbb{R}^{48\times4\times4} * \mathbb{R}^{64\times3\times3}$ + Batch Normalization + Binary Activation.
12. *Average Pooling*: Outputs $\mathbb{R}^{64\times1\times1}$
13. *Fully Connected*: Outputs the classification result $\mathbb{R}^{10\times1} \leftarrow \mathbb{R}^{10\times64} \cdot \mathbb{R}^{64\times1}$

Figure 3. BC2 from [25]

1. *Convolution*: input image $3 \times 32 \times 32$, weight matrix $16 \times 3 \times 3$, number of output channels 16: $\mathbb{R}^{16\times32\times32} \leftarrow \sum \mathbb{R}^{3\times32\times32} * \mathbb{R}^{16\times3\times3}$ + Batch Normalization + Binary Activation.
2. *Convolution*: input image $16 \times 32 \times 32$, weight matrix $32 \times 3 \times 3$, number of output channels 32: $\mathbb{R}^{32\times32\times32} \leftarrow \sum \mathbb{R}^{16\times32\times32} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
3. *Convolution*: input image $32 \times 32 \times 32$, weight matrix $32 \times 3 \times 3$, number of output channels 32: $\mathbb{R}^{32\times32\times32} \leftarrow \sum \mathbb{R}^{32\times32\times32} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
4. Same as Layer 3.
5. *Average Pooling*: Outputs $\mathbb{R}^{32\times16\times16}$.
6. *Convolution*: input image $32 \times 16 \times 16$, weight matrix $48 \times 3 \times 3$, number of output channels 48: $\mathbb{R}^{32\times16\times16} \leftarrow \sum \mathbb{R}^{48\times16\times16} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
7. *Convolution*: input image $48 \times 16 \times 16$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times16\times16} \leftarrow \sum \mathbb{R}^{48\times16\times16} * \mathbb{R}^{64\times3\times3}$ + Batch Normalization + Binary Activation.
8. *Convolution*: input image $64 \times 16 \times 16$, weight matrix $80 \times 3 \times 3$, number of output channels 80: $\mathbb{R}^{80\times16\times16} \leftarrow \sum \mathbb{R}^{64\times16\times16} * \mathbb{R}^{80\times3\times3}$ + Batch Normalization + Binary Activation.
9. *Average Pooling*: Outputs $\mathbb{R}^{80\times8\times8}$
10. *Convolution*: input image $80 \times 8 \times 8$, weight matrix $96 \times 3 \times 3$, number of output channels 96: $\mathbb{R}^{96\times6\times6} \leftarrow \sum \mathbb{R}^{80\times8\times8} * \mathbb{R}^{96\times3\times3}$ + Batch Normalization + Binary Activation.
11. *Convolution*: input image $96 \times 6 \times 6$, weight matrix $96 \times 3 \times 3$, number of output channels 96: $\mathbb{R}^{96\times4\times4} \leftarrow \sum \mathbb{R}^{96\times6\times6} * \mathbb{R}^{96\times3\times3}$ + Batch Normalization + Binary Activation.
12. *Convolution*: input image $96 \times 4 \times 4$, weight matrix $128 \times 3 \times 3$, number of output channels 128: $\mathbb{R}^{128\times2\times2} \leftarrow \sum \mathbb{R}^{96\times4\times4} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
13. *Average Pooling*: Outputs $\mathbb{R}^{128\times1\times1}$
14. *Fully Connected*: Outputs the classification result $\mathbb{R}^{10\times1} \leftarrow \mathbb{R}^{10\times128} \cdot \mathbb{R}^{128\times1}$

Figure 4. BC3 from [25]

1. *Convolution*: input image $3 \times 32 \times 32$, weight matrix $32 \times 3 \times 3$, number of output channels 32: $\mathbb{R}^{32\times32\times32} \leftarrow \sum \mathbb{R}^{3\times32\times32} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
2. *Convolution*: input image $32 \times 32 \times 32$, weight matrix $32 \times 3 \times 3$, number of output channels 32: $\mathbb{R}^{32\times32\times32} \leftarrow \sum \mathbb{R}^{32\times32\times32} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
3. *Convolution*: input image $32 \times 32 \times 32$, weight matrix $48 \times 3 \times 3$, number of output channels 48: $\mathbb{R}^{48\times32\times32} \leftarrow \sum \mathbb{R}^{32\times32\times32} * \mathbb{R}^{48\times3\times3}$ + Batch Normalization + Binary Activation.
4. *Convolution*: input image $48 \times 32 \times 32$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times32\times32} \leftarrow \sum \mathbb{R}^{48\times32\times32} * \mathbb{R}^{64\times3\times3}$ + Batch Normalization + Binary Activation.
5. *Convolution*: input image $64 \times 32 \times 32$, weight matrix $64 \times 3 \times 3$, number of output channels 64: $\mathbb{R}^{64\times32\times32} \leftarrow \sum \mathbb{R}^{64\times32\times32} * \mathbb{R}^{64\times3\times3}$ + Batch Normalization + Binary Activation.
6. *Average Pooling*: Outputs $\mathbb{R}^{64\times16\times16}$.
7. *Convolution*: input image $64 \times 16 \times 16$, weight matrix $80 \times 3 \times 3$, number of output channels 80: $\mathbb{R}^{80\times16\times16} \leftarrow \sum \mathbb{R}^{64\times16\times16} * \mathbb{R}^{80\times3\times3}$ + Batch Normalization + Binary Activation.
8. *Convolution*: input image $80 \times 16 \times 16$, weight matrix $80 \times 3 \times 3$, number of output channels 80: $\mathbb{R}^{80\times16\times16} \leftarrow \sum \mathbb{R}^{80\times16\times16} * \mathbb{R}^{80\times3\times3}$ + Batch Normalization + Binary Activation.
9. Same as Layer 8.
10. Same as Layer 8.
11. *Average Pooling*: Outputs $\mathbb{R}^{80\times8\times8}$
12. *Convolution*: input image $80 \times 8 \times 8$, weight matrix $128 \times 3 \times 3$, number of output channels 128: $\mathbb{R}^{128\times6\times6} \leftarrow \sum \mathbb{R}^{80\times8\times8} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
13. *Convolution*: input image $128 \times 6 \times 6$, weight matrix $128 \times 3 \times 3$, number of output channels 128: $\mathbb{R}^{128\times4\times4} \leftarrow \sum \mathbb{R}^{128\times6\times6} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
14. *Convolution*: input image $128 \times 4 \times 4$, weight matrix $128 \times 3 \times 3$, number of output channels 128: $\mathbb{R}^{128\times2\times2} \leftarrow \sum \mathbb{R}^{128\times4\times4} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
15. *Average Pooling*: Outputs $\mathbb{R}^{128\times1\times1}$
16. *Fully Connected*: Outputs the classification result $\mathbb{R}^{10\times1} \leftarrow \mathbb{R}^{10\times128} \cdot \mathbb{R}^{128\times1}$

Figure 5. BC4 from [25]

1. *Convolution*: input image $3 \times 32 \times 32$, weight matrix $32 \times 3 \times 3$, number of output channels 32:
$\mathbb{R}^{32\times32\times32} \leftarrow \sum \mathbb{R}^{3\times32\times32} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
2. *Convolution*: input image $32 \times 32 \times 32$, weight matrix $32 \times 3 \times 3$, number of output channels 32:
$\mathbb{R}^{32\times32\times32} \leftarrow \sum \mathbb{R}^{32\times32\times32} * \mathbb{R}^{32\times3\times3}$ + Batch Normalization + Binary Activation.
3. Same as Layer 2.
4. Same as Layer 2.
5. *Convolution*: input image $32 \times 32 \times 32$, weight matrix $48 \times 3 \times 3$, number of output channels 48:
$\mathbb{R}^{48\times32\times32} \leftarrow \sum \mathbb{R}^{32\times32\times32} * \mathbb{R}^{48\times3\times3}$ + Batch Normalization + Binary Activation.
6. *Convolution*: input image $48 \times 32 \times 32$, weight matrix $48 \times 3 \times 3$, number of output channels 48:
$\mathbb{R}^{48\times32\times32} \leftarrow \sum \mathbb{R}^{48\times32\times32} * \mathbb{R}^{48\times3\times3}$ + Batch Normalization + Binary Activation.
7. *Average Pooling*: Outputs $\mathbb{R}^{48\times16\times16}$.
8. *Convolution*: input image $48 \times 16 \times 16$, weight matrix $80 \times 3 \times 3$, number of output channels 80:
$\mathbb{R}^{80\times16\times16} \leftarrow \sum \mathbb{R}^{48\times16\times16} * \mathbb{R}^{80\times3\times3}$ + Batch Normalization + Binary Activation.
9. *Convolution*: input image $80 \times 16 \times 16$, weight matrix $80 \times 3 \times 3$, number of output channels 80:
$\mathbb{R}^{80\times16\times16} \leftarrow \sum \mathbb{R}^{80\times16\times16} * \mathbb{R}^{80\times3\times3}$ + Batch Normalization + Binary Activation.
10. Same as Layer 7.
11. Same as Layer 7.
12. Same as Layer 7.
13. Same as Layer 7.
14. *Average Pooling*: Outputs $\mathbb{R}^{80\times8\times8}$
15. *Convolution*: input image $80 \times 8 \times 8$, weight matrix $128 \times 3 \times 3$, number of output channels 128:
$\mathbb{R}^{128\times6\times6} \leftarrow \sum \mathbb{R}^{80\times8\times8} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
16. *Convolution*: input image $128 \times 8 \times 8$, weight matrix $128 \times 3 \times 3$, number of output channels 128:
$\mathbb{R}^{128\times8\times8} \leftarrow \sum \mathbb{R}^{128\times8\times8} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
17. Same as 16.
18. *Convolution*: input image $128 \times 8 \times 8$, weight matrix $128 \times 3 \times 3$, number of output channels 128:
$\mathbb{R}^{128\times6\times6} \leftarrow \sum \mathbb{R}^{128\times8\times8} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
19. *Convolution*: input image $128 \times 6 \times 6$, weight matrix $128 \times 3 \times 3$, number of output channels 128:
$\mathbb{R}^{128\times4\times4} \leftarrow \sum \mathbb{R}^{128\times6\times6} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
20. *Convolution*: input image $128 \times 4 \times 4$, weight matrix $128 \times 3 \times 3$, number of output channels 128:
$\mathbb{R}^{128\times2\times2} \leftarrow \sum \mathbb{R}^{128\times4\times4} * \mathbb{R}^{128\times3\times3}$ + Batch Normalization + Binary Activation.
21. *Average Pooling*: Outputs $\mathbb{R}^{128\times1\times1}$
22. *Fully Connected*: Outputs the classification result $\mathbb{R}^{10\times1} \leftarrow \mathbb{R}^{10\times128} . \mathbb{R}^{128\times1}$

Figure 6. BC5 from [25]

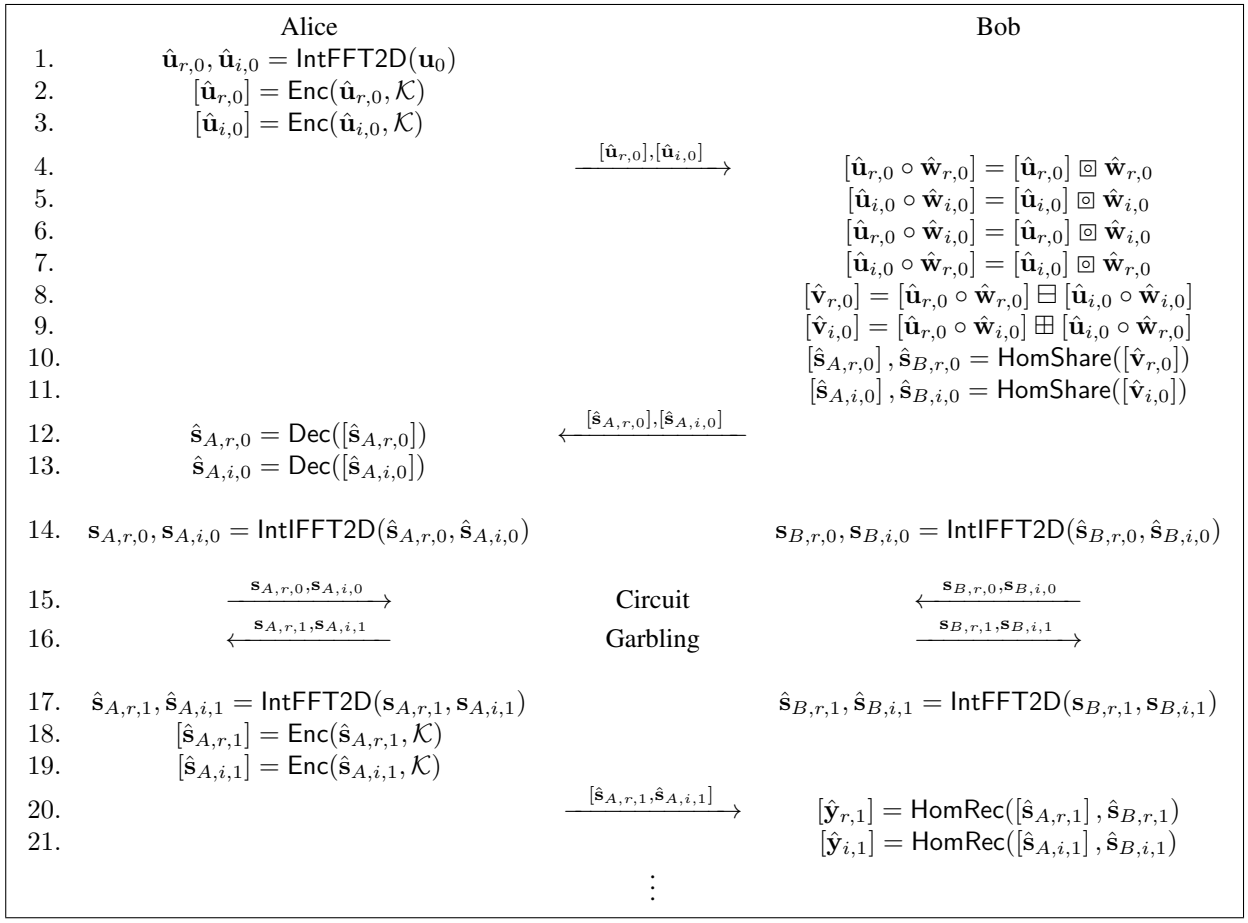

Figure 7. One start and one intermediate rounds of the Gazelle protocol with frequency-domain convolution via $\mathsf{IntFFT2D}$.