High-capacity scattering-augmented optical encryption: supplementary material

Liheng Bian[1,2,3,#,*] Xuyang Chang,[1,2,#] Shaowei Jiang,[5] Liming Yang,[4] Xinrui Zhan,[1,2] Shicong Liu,[1,2] Daoyu Li,[1,2] Rong Yan,[1,2] Zhen Gao,[1,2,3] Guoan Zheng[4,*] and Jun Zhang[1,2,3,*]

[1]*MIIT Key Laboratory of Complex-field Intelligent Sensing, Beijing Institute of Technology, Beijing 100081, China*

[2]*School of Information and Electronics & Advanced Research Institute of Multidisciplinary Science, Beijing Institute of Technology, Beijing 100081, China*

[3]*Yangtze Delta Region Academy of Beijing Institute of Technology (Jiaxing), Jiaxing 314019, China*

[4]*Department of Biomedical Engineering, University of Connecticut, Storrs, Connecticut 06269, USA*

[5]*School of Communication Engineering, Hangzhou Dianzi University, Hangzhou, 310018, China*

[#]*These authors contributed equally to this work*

[*]*bian@bit.edu.cn*

[*]*guoan.zheng@uconn.edu*

[*]*zhjun@bit.edu.cn*

Contents

1 Supplementary Note 1: Scattering layer calibration.

The scattering layer of the reported prototype required to be calibrated before encryption. Benefitting from the ptychography setup, the calibration was implemented using computational methods. We placed the scattering layer to a translation stage and moved to 1521 random positions, as shown in Fig. S1 (a). The distance between each position was about a few microns. The detector captured corresponding intensity-only measurements. Then, these measurements were input to the ePIE algorithm [1] to recover the complex-domain high-resolution scattering layer profile. Algorithm 1 presents the calibration algorithm (ePIE). It started from a random high-resolution initialization, then iteratively propagated among the plaintext plane, the scattering layer plane and the detector plane. The prior constraints were imposed in iterations. After the calibration finished, we selected 4 - 25 positions for encryption, as shown in Fig. S1 (b). These positions and their modulation patterns are significant secret keys for decryption.

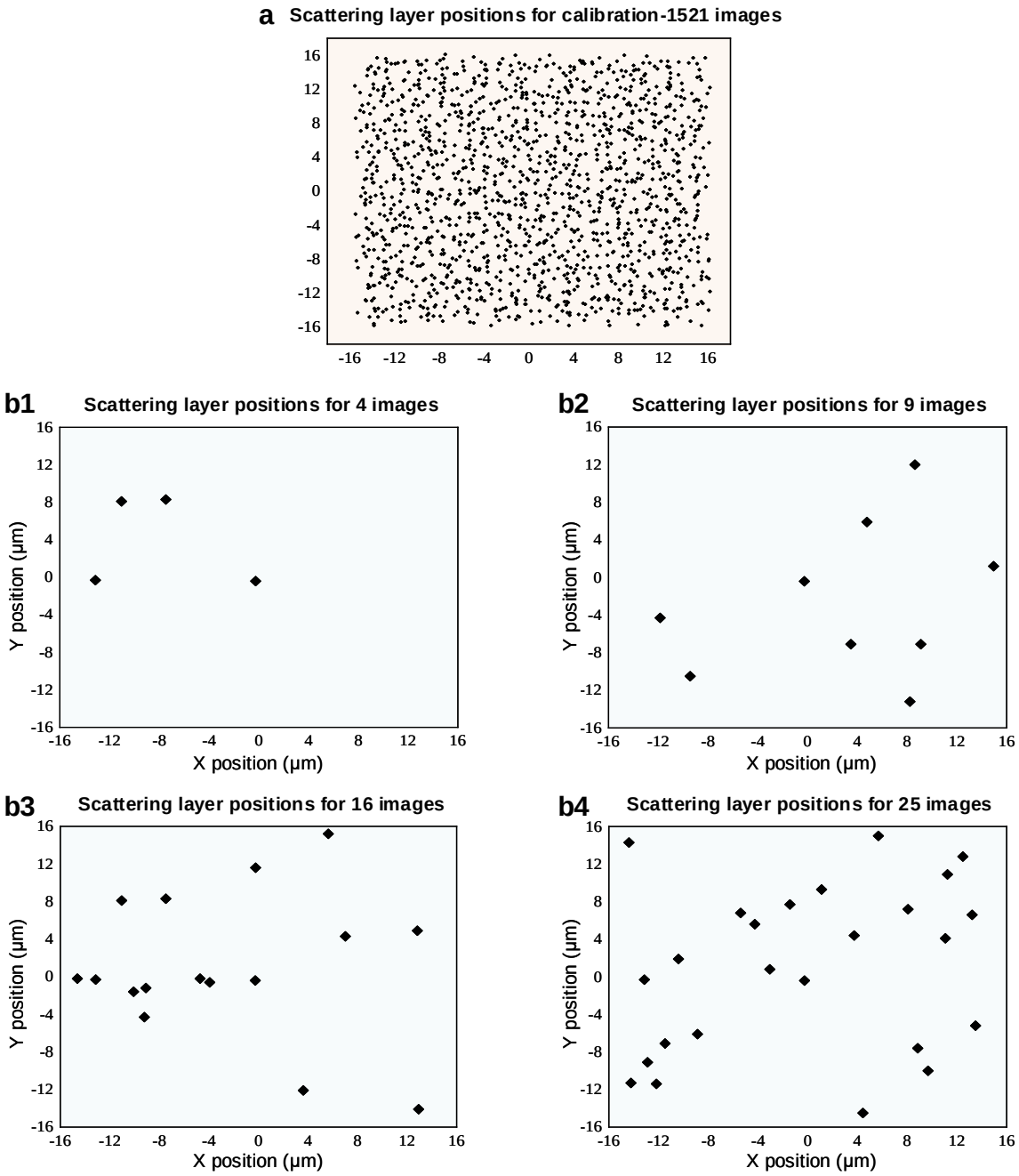

Figure S1: **Positions for calibration and encryption.** (a) The 1521 positions for calibration. (b) The 4 - 25 positions for encryption, which correspond to different compressive ratios.

Algorithm 1 ePIE algorithm for scattering layer calibration

Input: intensity-only measurements I, initialization sample u_0 and scattering layer's pattern D_0, forward model A and free space propagation function $\mathcal{P}(d)$

Output: scattering layer's profile D

$W = u^{(k)} * \mathcal{P}(d_1)$ ▷ Propagation d_1 to scattering layer plane

for $k = 1, 2, \ldots$ **do**

  for $i = 1, 2, \ldots$ **do**

    $\mathcal{W}_i \leftarrow \mathcal{W}\left(x + x_i, y - y_i\right)$ ▷ Sensor shift

    $\phi_i = \mathcal{W}_i \odot D$ ▷ Scattering layer modulation

    $\psi_i = \phi_i * \mathcal{P}(d_2)$ ▷ Propagation d_2 to the detector plane

    $\psi_i' = \sqrt{I_i^{(k)}} \odot \frac{\psi_i}{|\psi_i|}$ ▷ Update wavefront using intensity constraint

    $\phi_i' = \psi_i' * \mathcal{P}(-d_2)$ ▷ Propagation $-d_2$ to the scattering layer plane

    $\mathcal{W} = \mathcal{W} + \frac{\mathrm{conj}(D) \odot \left[\phi_i' - \phi_i\right]}{(1-\alpha_1)|D|^2 + \alpha_1 |D|^2_{\max}}$ ▷ Updata sample wavefront

    $D = D + \frac{\mathrm{conj}(\mathcal{W}) \odot \left[\phi_i' - \phi_i\right]}{(1-\alpha_2)|\mathcal{W}|^2 + \alpha_2 |\mathcal{W}|^2_{\max}}$ ▷ Updata scattering layer

    $\mathcal{W} \leftarrow \mathcal{W}_i\left(x - x_i, y + y_i\right)$ ▷ Sensor shift back

2 Supplementary Note 2: Additional results of experiments.

The plaintext target was prepared by photoetching technique, as shown in Fig. S2 (left). Two ∼500 μm^2 targets were used as plaintexts. Fig. S2 (right) presents the images of plaintexts that were captured using a microscope with a 10X objective and 3.45-μm camera.

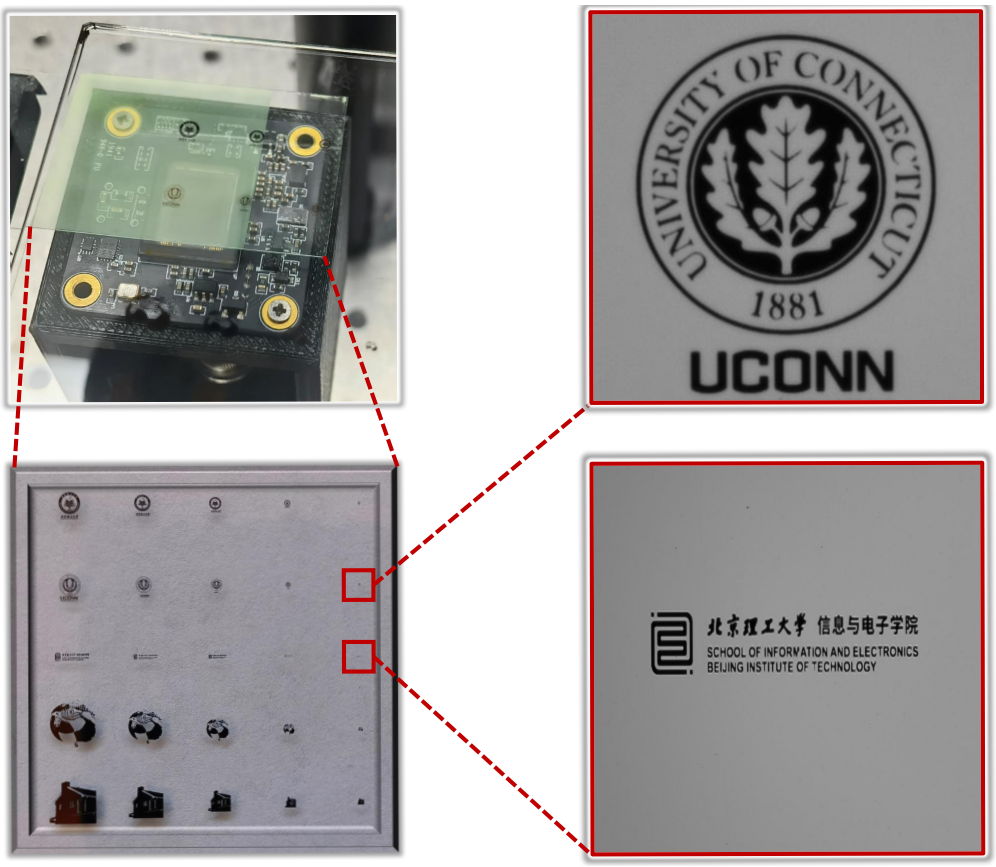

Figure S2: **Photoetched plaintext targets.** The close-ups on the right show the ground truths which are captured using a microscope with 10X objective and 3.45-μm camera.

Figure S3 - S4 show the results of the reported encryption technique. We can see that the ciphertexts are chaotic, without any visual information. The alternating projection (AP) method can only recover degraded results which have limited resolution. In contrast, the reported decryption algorithm can reconstruct high-fidelity plaintexts under various compressive ratios.

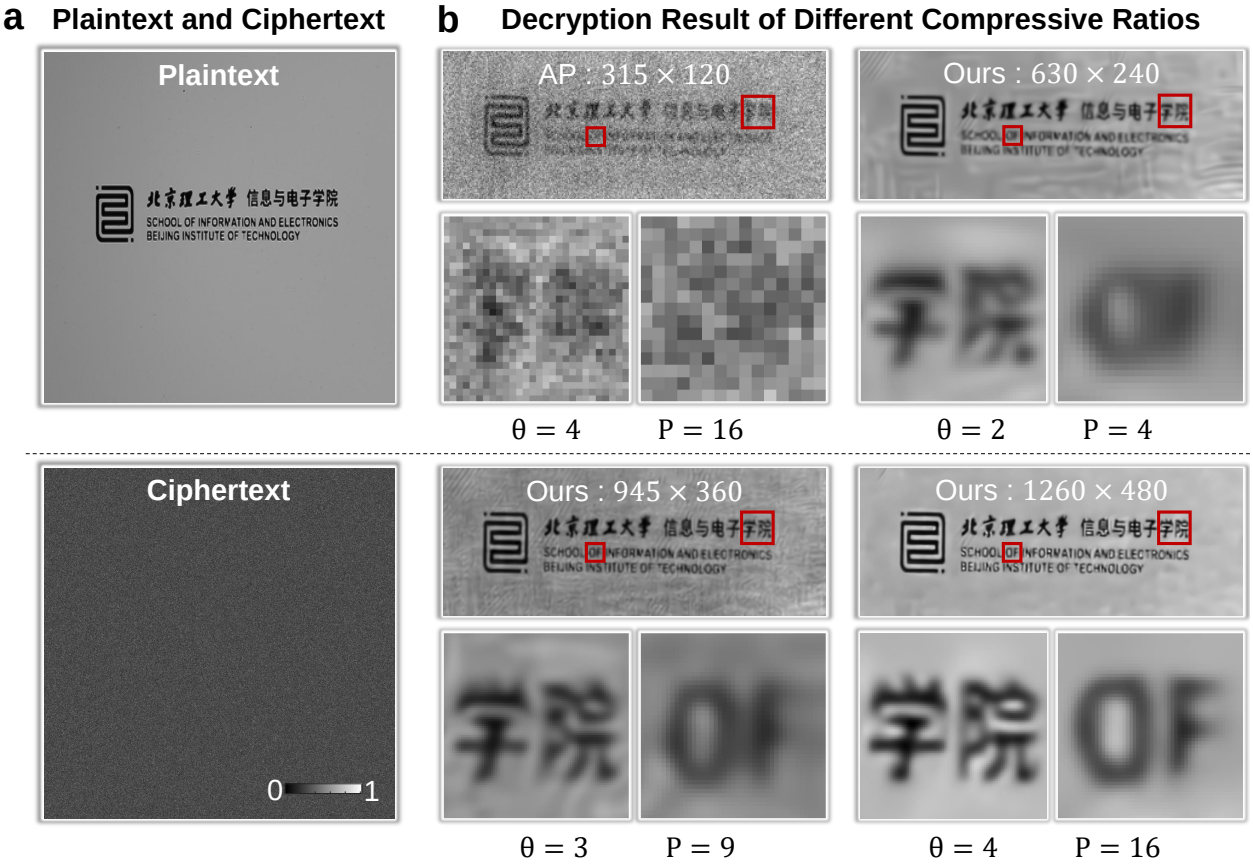

Figure S3: **Additional results of photoetched target under different compressive ratios θ and the number of intensity-only images P.**

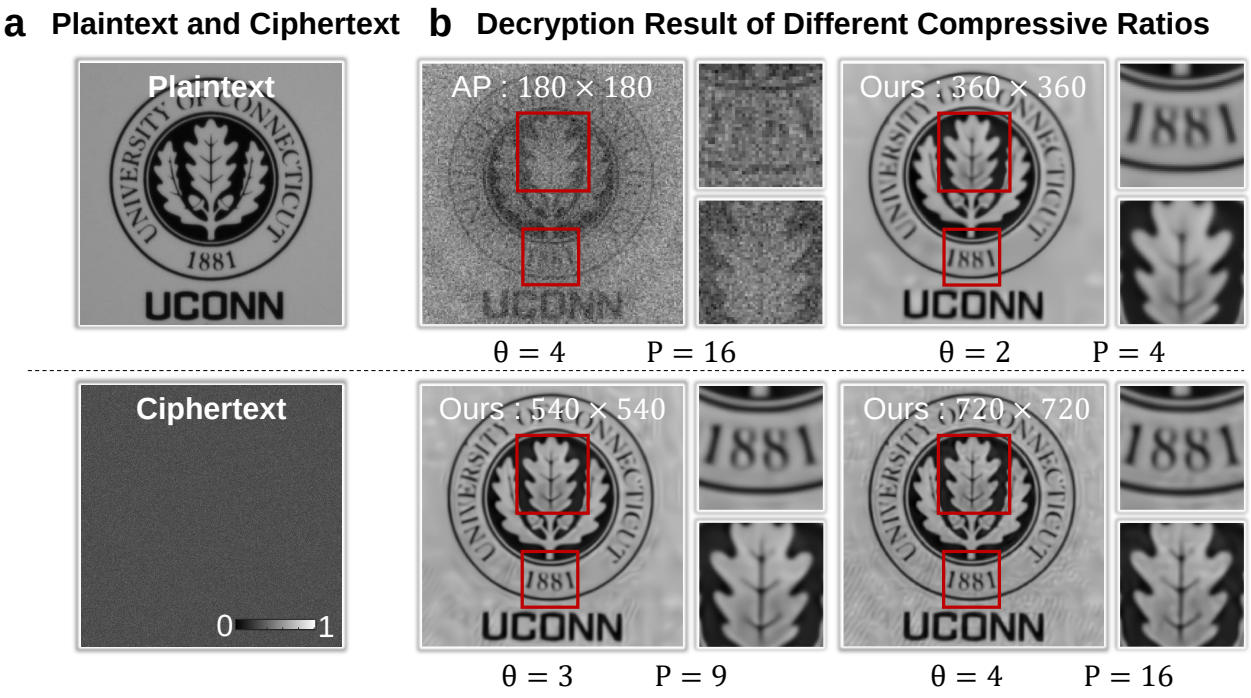

Figure S4: **Additional results of photoetched target under different compressive ratios θ and the number of intensity-only images P.**

3 Supplementary Note 3: Randomness test.

An ideal random stream generator requires randomness and unpredictability. Specifically, the elements of the generated sequence are independent of each other. The subsequent element in the sequence cannot be predicted, even if the previous elements are known. In our test, we extracted the phase of modulated scattering layer, and compared its randomness with other computer-generated patterns. These comparison patterns were generated by different random number generation (RNG) algorithms. To unify the data types, these patterns were first mapped to the range of 0 to 255, and then transformed into 8-bit binary code.

Randomness is a probabilistic property. The properties of a random sequence can be characterized and described in terms of probability. According to this theory, we employed a statistical test program from the National Institute of Standards and Technology (NIST) [2] to test randomness. The NIST Test Suite is a statistical package that contains 15 items to test the randomness of (arbitrarily long) binary sequences produced by either hardware or software based cryptographic random or pseudorandom number generators. These tests focus on exploring various types of non-randomness in the test sequences. The details about the 15 test items are as follows

1) Frequency (Monobit) Test. The frequency test aims to calculate the proportion of zeroes and ones for the entire sequence. Generally, the number of zeroes and ones should be about the same in a truly random sequence.

2) Frequency Test within a Block. This test is similar to the frequency test. It calculates the proportion of ones within a M-bit blocks, and the expected number is approximately M/2.

3) Runs Test. The purpose of this test is to determine whether the oscillation between zeros and ones is too fast or too slow. A run is an uninterrupted sequence of identical bits, and this item records the total number of runs in the test sequence.

4) Test for the Longest-Run-of-Ones in a Block. This test calculates the longest run of ones within M-bit blocks.

5) Binary Matrix Rank Test. This test checks the linear dependence among fixed length sub-strings of the original sequence using the rank of disjoint sub-matrices of the entire sequence.

6) Discrete Fourier Transform (Spectral) Test. The spectral test detects the periodic features (such as repetitive patterns between adjacent elements) in the tested sequence that would indicate a deviation from the assumption of randomness.

7) Overlapping Template Matching Test. The focus of this test is the number of occurrences of prespecified target strings.

8) Maurer's 'Universal Statistical' Test. Generally, a significantly compressible sequence is considered to be non-random. This test calculates the number of bits between matching patterns, and tests whether the sequence can be compressed without information loss.

9) Linear Complexity Test. The linear complexity test records the length of a linear feedback shift register to determine whether the sequence is complex enough.

10) Approximate Entropy Test. The approximate entropy test is the same as the serial test. However, it tests the adjacent m and m+1 lengths.

11) Cumulative Sums (Cusums) Test. This test focuses on the maximal excursion (from zero) of the random walk which is defined by the cumulative sum of adjusted (-1, +1) digits in the sequence.

12) Non-overlapping Template Matching Test. This test records the occurrence numbers of pre-specified target strings. The purpose is to detect generators that produce too many occurrences of a given non-periodic (aperiodic) pattern.

13) Serial Test. The focus of this test is the frequency of all possible overlapping m-bit patterns across the entire sequence.

14) Random Excursions Test. This test detects the number of cycles having exactly K visits in a cumulative sum random walk. The cumulative sum random walk is derived from partial sums after the (0,1) the sequence is transferred to the appropriate (-1, +1) sequence.

15) Random Excursions Variant Test. This test records the total number of times that a particular state is visited in a cumulative sum random walk.

4 Supplementary Note 4: Ciphertext entropy.

The ciphertext entropy evaluates the correlation of each pixel. The ciphertext was modeled as a 1D discrete sequence $\mathcal{C}_{cipher} \in R^{N\times 1}$ (N is the total pixel number). We normalized the ciphertext sequence to integers with the range of 0$\sim$255. In this way, each pixel in the ciphertext sequence should be independent and has an approximate probability between 0$\sim$255. In our experiment, we regarded the value range as a sample space ($M = 256$), which contains 256 elements S_i ($i = 1, 2, \cdots, M$). Then we calculated the total number of each element within the ciphertext sequence. Thus, the probability of sample elements can be indicated as

$$\mathrm{P}\left(S_i\right) = \frac{T}{N}, \tag{1}$$

where T is the frequency of the element S_i within the ciphertext sequence. Then, according to the information entropy theory, the ciphertext entropy $H(\mathcal{S})$ is the sum of a series of sample elements

$$H(\mathcal{S}) = -\sum_{i=1}^{M} \mathrm{P}\left(\mathcal{S}_i\right) \log \mathrm{P}\left(\mathcal{S}_i\right). \tag{2}$$

According to Eq. (2), if each sample element has the same probabilities, the maximum ciphertext entropy should be 8.

References

1. Maiden, A. M. & Rodenburg, J. M. An improved ptychographical phase retrieval algorithm for diffractive imaging. *Ultramicroscopy* **109**, 1256–1262 (2009).

2. Rukhin, A., Soto, J., Nechvatal, J., Smid, M. & Barker, E. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. Rep., Booz-allen and hamilton inc mclean va (2001).