Supplementary Materials for

Multiuser computational imaging encryption and authentication with OFDM-assisted key management

Hongran Zeng[†], Ping Lu[†], Xiaowei Li[†]*, *et al.*

*Corresponding author. xwli@scu.edu.cn (X. L.), liuyg@scu.edu.cn (Y. L.), qionghua@buaa.edu.cn (Q.-H. W.)

This file includes:

S1. SPI encryption and Keys encapsulation

S1.1 SPI encryption algorithm

Algorithm: Procedure of the Fourier SPI encryption

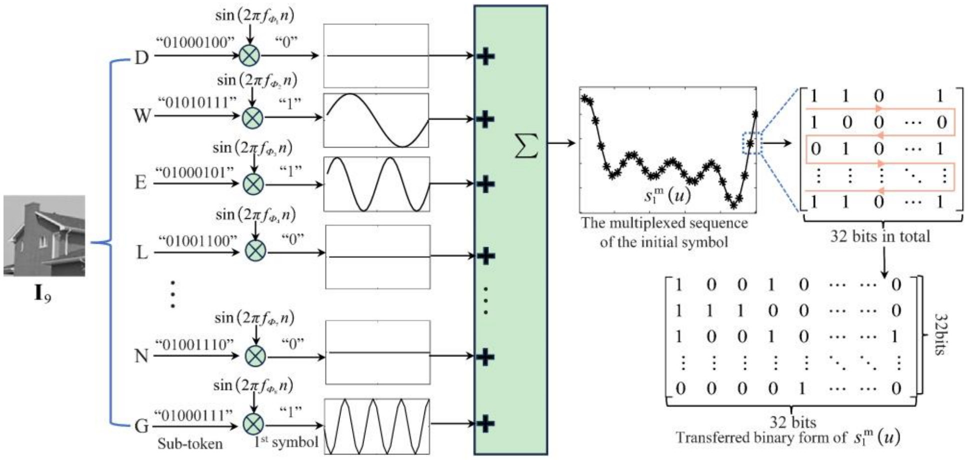
```
Input: $\mathbf{I}_{concate}$ with size $3m \times 3n$;
Output: $\mathbf{I}_{diffuse}^{3m\times 3n}$;
Independently initialize: $\{\mathbf{W}_k^{m\times n}\},\ \{\mathbf{P}_k^{u\times v}\},\ k = 1, 2, ..., K;\ \{\mathbf{D}_q^{3m\times 3n}\},\ q = 1, 2, ..., Q$;
$\mathbf{W}_{concate}^{3m\times 3n} \leftarrow$ concatenate $\{\mathbf{W}_k^{m\times n}\}$ in the same way with $\mathbf{I}_{concate}$;
$\mathbf{I}_{whiten}^{3m\times 3n} = \mathbf{I}_{concate} \oplus \mathbf{W}_{concate}^{3m\times 3n}$;                    ✦ Whitening
$\mathbf{P}_{concate}^{3u\times 3v} \leftarrow$ concatenate $\mathbf{P}_k^{u\times v}$;
$\mathbf{I}_{diffuse}^{3m\times 3n} = \mathbf{I}_{permu}^{3m\times 3n} = \pi(\mathbf{P}_{concate}^{3u\times 3v}, \mathbf{I}_{whiten}^{3m\times 3n})$;                    ✦ Permutation
for $r = 1, 2, ..., 3m$                    ✦ 1st diffusion
    $\mathbf{I}_{diffuse}^{3m\times 3n}(r,:) = \mathbf{I}_{diffuse}^{3m\times 3n}(r,:) \oplus \mathbf{I}_{diffuse}^{3m\times 3n}(r-1,:) \oplus \mathbf{D}_1^{3m\times 3n}(r,:)$;
end for
for $l = 1, 2, ..., 3n$
    $\mathbf{I}_{diffuse}^{3m\times 3n}(:,l) = \mathbf{I}_{diffuse}^{3m\times 3n}(:,l) + \mathbf{I}_{diffuse}^{3m\times 3n}(:,l-1) + \mathbf{D}_1^{3m\times 3n}(:,l)$;
end for
for $r = 1, 2, ..., 3m$                    ✦ 2nd diffusion
    $\mathbf{I}_{diffuse}^{3m\times 3n}(r,:) = (\mathbf{I}_{diffuse}^{3m\times 3n}(r,:) + \mathbf{I}_{diffuse}^{3m\times 3n}(r-1,:) + \mathbf{D}_1^{3m\times 3n}(r,:))MOD2^8$;
end for
for $l = 1, 2, ..., 3n$
    $\mathbf{I}_{diffuse}^{3m\times 3n}(:,l) = (\mathbf{I}_{diffuse}^{3m\times 3n}(:,l) + \mathbf{I}_{diffuse}^{3m\times 3n}(:,l-1) + \mathbf{D}_1^{3m\times 3n}(:,l))MOD2^8$;
end for
$O_\phi = \langle \mathbf{I}_{diffuse}^{3m\times 3n}, \mathbf{J}_\phi \rangle$                    ✦Fourier SPI encryption
```

S1.2 OFDM-like modulation with trigonometric functions

The host first initializes the initial frequency f_I as well as the frequency interval Δf , and then assign a different subcarrier index (i.e., value Φ) to a different user. Because of the orthogonality of the trigonometric functions with different type and frequency, as expressed in Eq. S1, the frequency can in fact be repeatedly used once. Here, to achieve more clearer usage protocol of OFDM-assisted key management, we apply sinusoidal functions only.

$$\begin{cases} \langle \mathcal{Re}\left[e^{j2\pi f_I t}\right] \cdot \mathcal{Re}\left[e^{j2\pi(f_I + \Phi\Delta f)t}\right] \rangle = 0 \\ \langle \mathcal{Re}\left[e^{j2\pi f_I t}\right] \cdot \mathcal{Im}\left[e^{j2\pi(f_I + \Phi\Delta f)t}\right] \rangle = 0 \\ \langle \mathcal{Im}\left[e^{j2\pi f_I t}\right] \cdot \mathcal{Im}\left[e^{j2\pi(f_I + \Phi\Delta f)t}\right] \rangle = 0 \end{cases} \tag{S1}$$

The flow chart of the sinusoidal modulation of letters is illustrated in Fig S1. N_S is set as 32 and $\{\Phi\}$ value for users is set as $\Phi_1 = 13$, $\Phi_2 = 1$, …, $\Phi_7 = 24$, $\Phi_8 = 4$. As shown in Fig S1, each symbol in a sub-token is modulated by the same subcarrier and the symbols in the same position of the eight sub-tokens are multiplied by the respective subcarrier and added together, which is exactly the same principle of OFDM modulation in complex domain. Note that $s_p^{\mathrm{m}}(u), p = 1, 2, ..., 8,$ is the expanding expressing form of the p^{th} row of matrix $\mathbf{S} = \sum_{t=1}^{8} \mathbf{s}_t \cdot \mathbf{y}_{\Phi_t}$. In other words, $s_p^{\mathrm{m}}(u)$ is the expanding expressing form of $\mathbf{S}(p, :)$. The superscript "m" denotes "multiplexing". The added sequence $s_p^{\mathrm{m}}(u)$, $u = 1, 2, ..., 32$, belongs to the decimal sequence. Therefore, to record them as the monochrome meta-image, we convert $s_p^{\mathrm{m}}(u)$ into binary sequence still in terms of the IEEE Standard 754 and only reserve the first 32 significant digits. Each symbol is represented by 32-bit sequence and the total mask contains 8192 bits (i.e., $32 \times 32 \times 8$ bits).



$\mathbf{I}_9$
D
W
E
L
N
G
Sub-token
"01000100"
"01010111"
"01000101"
"01001100"
"01001110"
"01000111"
$\sin(2\pi f_{\Phi_1} n)$
$\sin(2\pi f_{\Phi_2} n)$
$\sin(2\pi f_{\Phi_3} n)$
$\sin(2\pi f_{\Phi_4} n)$
$\sin(2\pi f_{\Phi_7} n)$
$\sin(2\pi f_{\Phi_8} n)$
"0"
"1"
"1"
"0"
"0"
"1"
1st symbol
$\sum$
$s_1^{\mathrm{m}}(u)$
The multiplexed sequence of the initial symbol
32 bits in total
32bits
32 bits
Transferred binary form of $s_1^{\mathrm{m}}(u)$

Fig S1. Schematic of trigonometric OFDM-like modulation of the first character of users. f_{Φ_1} denotes the carrier frequency $f_I + \Phi_1\Delta f$, f_{Φ_2} denotes the carrier frequency $f_I + \Phi_2\Delta f$ and so on.

As the protecting measures for verification information, the subcarrier index should be individually owned by each user (as a Φ), similar to the masks of regional encryption, in order to retain the integrity of the authentication process. Note that the types of trigonometric functions are presumed as prior-known condition of OFDM-assisted key management protocol. Thus, we do not have to state the modulating carrier in advance in OFDM-assisted key management. $\mathbf{\Phi}$ are converted into binary sequence for private channel coding in terms of normal decimal-to-binary regulation and each value ranging $[1,32]$ is represented by 8-bit sequence (e.g., User $1 \rightarrow \mathit{\Phi}_1 \rightarrow$ "13" $\rightarrow$ "00001101").

S1.3 OFDM-like modulation with Hadamard sequences

The modulation process with Hadamard subcarrier is almost the same as that with sin/cosine functions. The only two differences exist in the type of $\mathit{\Phi}$ assigned when generating a different subcarrier for a different user and the regulation of final decimal-to-binary transform of $s_p^{\mathrm{m}}(u)$. When it comes to generating subcarriers for users, we first create a Hadamard matrix with the size of $N_s = 64$. After that, assign different $\{\mathit{\Phi}\}$ values (i.e., the row indices of the matrix) only (i.e., without the subcarrier types and other initialization parameters of trigonometric carriers) to users, such as $\mathit{\Phi}_1 = 14$, $\mathit{\Phi}_2 = 11$, …, $\mathit{\Phi}_7 = 32$, $\mathit{\Phi}_8 = 61$, followed by the modulation process of sub-tokens. For "*Dwelling*" string, the produced $s_p^{\mathrm{m}}(u)$ ranging from -6 to 8 is then normalized as non-negative numbers within the interval $[2,16]$. Finally, in terms of the principle that the digit value of $s_p^{\mathrm{m}}(u)$ equals to the number of ones in converted binary sequence (e.g., $s_1^{\mathrm{m}}(5) = 16 = 1111111111111111$, $s_2^{\mathrm{m}}(2) = 8 = 1111111100000000$), the normalized $s_p^{\mathrm{m}}(u)$ is converted into binary type. Such the transform principle uniformly distributes the numerical magnitudes across every position of the binary sequence, thereby mitigating the obvious impact of high-bit changes on the overall decimal value as observed in traditional decimal-to-binary regulations. In this method, the converted $s_p^{\mathrm{m}}(u)$ contains $64 \times 16 \times 8 = 8192$ bits in total for public channel coding and $\mathbf{\Phi}$ ranging within $[1,64]$ are converted into binary form in terms of the normal decimal-to-binary regulation (i.e., $16 \rightarrow 2^4 \rightarrow 10000$) for private channel coding.

S1.4 RSA asymmetric coding in private channel

The keys of regional encryption $\{\mathbf{W}_k^{m \times n}, \mathbf{P}_k^{u \times v}\}$ and $\mathbf{\Phi}$ are respectively used to protect plaintexts and authentication information of each user away from hostile steal by other users. They are further designed to be cross-encapsulated by RSA encryption in private channel.

Table S1. Quantified security assessment of fusing Fourier encryption

Image	Histogram			Correlation			Entropy			Sensitivity	
	Var.	Chi.	Flat.	Ver.	Horiz.	Diag.	Global	Local	$[h_{left}^{l\times\alpha}, h_{right}^{l\times\alpha}]$	NPCR	UACI
Cipher	339.61	268.49	0.0031	0.00217	0.00526	0.00283	7.9977	7.9028	*Pass*	99.63%	33.16%
"*Baboon*"	$2.676e^7$	$2.115e^7$	256.00	0.82128	0.77687	0.78227	6.8223	$\to 0$	*Not Pass*	/	/
"*Peppers*"	$2.677e^7$	$2.115e^7$	256.00	0.97433	0.96919	0.94055	6.4012	$\to 0$	*Not Pass*	/	/
"*Chart*"	$2.565e^7$	$2.026e^7$	245.32	0.90249	0.91344	0.82079	2.5601	1.7641	*Not Pass*	/	/

* For local entropy assessment, we randomly divide each image into 30 blocks, and the size of each segmented block is set to 44. Significant level is recommended as 0.001. $h_{left}^{l\times\alpha}$ and $h_{right}^{l\times\alpha}$ are set as 7.9015 and 7.9034 respectively. Var., Chi. and Flat. represent variance and chi-square χ^2 and flatness respectively. Ver., Horiz. and Diag. indicate the correlation in vertical, horizontal and diagonal direction respectively.

Each user first generates a product n of two large prime numbers (p,q) and calculate the number of positive integers in the range $[1, n-1]$ that are coprime to n (the greatest common divisor is 1) according to the Euler's theorem: $\varphi = (p-1)(q-1)$. Consequently, each user defines the public key e in the range $(1, \varphi-1]$ that is coprime to φ and derive the private key d subject to the equation $(d \times e) MOD \varphi = 1$. The product n and key e are then published by each user and d is preserved. The host concatenates the converted binary keys $\{\mathbf{W}_k^{m\times n}, \mathbf{P}_k^{u\times v}\}$ and $\mathbf{\Phi}$ in series for each user as the plaintext of RSA (i.e., User1: $\mathbf{h}_1 = \{\mathbf{W}_1^{m\times n}, \mathbf{P}_1^{u\times v}, \Phi_1\}$) after receiving public keys. Thus, the length of $\mathbf{h}_k$ is calculated as $88\ (\mathbf{W}_k^{m\times n}) + 640\ (\mathbf{P}_k^{u\times v}) + 8\ (\Phi_k) = 736$ bits, smaller than the regulated binary number of RSA coding (1024 bits). Eventually, $\mathbf{h}_k$ is encrypted in decimal form following the principle $\Lambda_k = deci\ ((\mathbf{h}_k)^e\ MOD\, n)$. Different Λ_k involving the private keys of both regional encryption and authentication corresponds to each user and finally all the Λ_k are transformed into binary form recorded on the private channel in sequence of the fabricated metasurface.

S2. Quantified security comparison of $\mathbf{I}_{diffuse}^{3m\times 3n}$

We digitally assess the security performance of the encrypted image $\mathbf{I}_{diffuse}^{3m\times 3n}$ and compare the quantified results with 8-bit-per-pixel images "*Baboon*", "*Peppers*" and monochrome image "*Resolution Chart*" (i.e., in brief "*Chart*"), as shown in Table S1. It can be concluded that the $\mathbf{I}_{diffuse}^{3m\times 3n}$ possesses the general property of a well-encrypted ciphertext, including flat pixel distribution (e.g., Histogram), low correlation among pixels (e.g., Correlation), desired pseudo-random presentation (e.g., Entropy) and high sensitivity (e.g., NPCR).

S3. Metasurface key zoning and sample optimization

S3.1 Metasurface key zoning design

After the hierarchical protection of $\mathbf{\Psi}$, $\mathbf{\Phi}$, the encrypted keys should be recoded on the metasurface. Different regions are recoded for different users. For the metasurface zoning of public channel, the distribution is assigned as Fig S2A. The chaotic conditions of $\{\mathbf{D}_q^{3m\times 3n}\}$ are zoned in the 1^{st} row, whereas the binary form of $\{\mathbf{P}_9^{u\times v}\}$ and chaos of $\{\mathbf{W}_9^{m\times n}\}$ are zoned in $2^{\text{nd}}-9^{\text{th}}$ row. $\mathbf{S}$ is positioned in the remaining space. Fig S2B demonstrates the metasurface zoning of private channel. Note that the prior-known corresponding regions of users are given by the usage protocol in advance, so there is no need for the host to mark extra declaration of the specific region of them on OFDM-assisted key management.

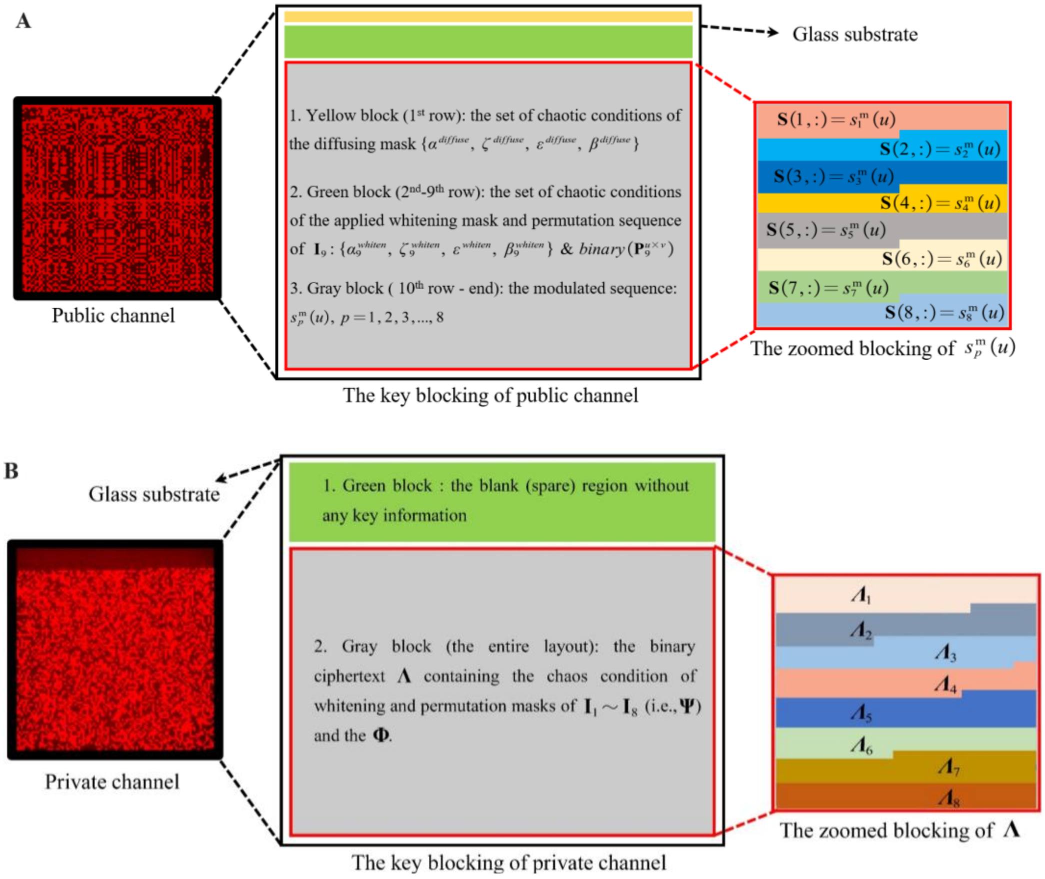

A
Glass substrate
1. Yellow block (1st row): the set of chaotic conditions of the diffusing mask $\{\alpha^{diffuse}, \zeta^{diffuse}, \varepsilon^{diffuse}, \beta^{diffuse}\}$
2. Green block (2nd-9th row): the set of chaotic conditions of the applied whitening mask and permutation sequence of $\mathbf{I}_9: \{\alpha_9^{whiten}, \zeta_9^{whiten}, \varepsilon^{whiten}, \beta_9^{whiten}\}$ & $binary(\mathbf{P}_9^{u\times v})$
3. Gray block (10th row - end): the modulated sequence: $s_p^{\mathrm{m}}(u),\ p=1,2,3,...,8$
$\mathbf{S}(1,:)=s_1^{\mathrm{m}}(u)$
$\mathbf{S}(2,:)=s_2^{\mathrm{m}}(u)$
$\mathbf{S}(3,:)=s_3^{\mathrm{m}}(u)$
$\mathbf{S}(4,:)=s_4^{\mathrm{m}}(u)$
$\mathbf{S}(5,:)=s_5^{\mathrm{m}}(u)$
$\mathbf{S}(6,:)=s_6^{\mathrm{m}}(u)$
$\mathbf{S}(7,:)=s_7^{\mathrm{m}}(u)$
$\mathbf{S}(8,:)=s_8^{\mathrm{m}}(u)$
Public channel
The zoomed blocking of $s_p^{\mathrm{m}}(u)$
The key blocking of public channel
B
Glass substrate
1. Green block : the blank (spare) region without any key information
2. Gray block (the entire layout): the binary ciphertext $\mathbf{\Lambda}$ containing the chaos condition of whitening and permutation masks of $\mathbf{I}_1 \sim \mathbf{I}_8$ (i.e., $\mathbf{\Psi}$) and the $\mathbf{\Phi}$.
Λ_1
Λ_2
Λ_3
Λ_4
Λ_5
Λ_6
Λ_7
Λ_8
Private channel
The zoomed blocking of $\mathbf{\Lambda}$
The key blocking of private channel

Fig S2. Metasurface key zoning of two main channels. (**A**) Block partition of different keys in public channel and the bit distribution of modulated sequence $s_p^{\mathrm{m}}(u)$. (**B**) Ciphertext distribution of RSAed keys.

The zoning design allows nanobricks located in different physical blocks to exhibit multiple sorts of keys. In Fig S2A, each sequence $\mathbf{S}(p,:)=s_p^{\mathrm{m}}(u)$, representing the expanding form of the p^{th} row of Matrix $\mathbf{S}$, corresponds to the modulation bits at the same position of eight tokens. Therefore, for each position of $s_p^{\mathrm{m}}(u)$, there is a dispersive and superposed representation of encrypted tokens bits with different subcarriers. To decipher the authentication image $\mathbf{I}_9$ and one's own token, users need to traverse all the information on the public channel. In Fig S2B, each user just simply corresponds to one ciphertext. Once obtaining the ciphertext, the user can directly decrypt it without the need to globally traverse all the ciphertexts in private channel. Note that for the zoning design of both public and private channel, if the corresponding position (such as the 1st row in public channel) is not fully filled, zero-padding will be used to fill the remaining space.

S3.2 Unit cell optimization

Optimize the size of the unit cell operating in the reflective mode. The nanobrick's long and short axis are set as long-axis (l) and short-axis respectively (s) and we conduct the optimization with respect to the incident light for both linear l-polarization and s-polarization. To prevent diffraction with high orders, the *CS* of the unit cell needs to be smaller than the wavelength. The wavelength range is set to be in the interval $[500,750]$ nm. Three parameters, specifically length, width and height of a nanobrick, are swept in the range of $[140,220]$ nm, $[60,140]$ nm and $[30,90]$ nm in both CST microwave studio and FDTD software. In addition, for the nanobricks on a metasurface with arbitrary orientations, the electromagnetic characteristics can undergo fluctuations influenced by the surrounding nanobricks due to coupling effect. Thus, the optimization objective is to achieve higher reflectance at the target wavelength so as to enhance the Plasmon resonance while the coupling effect should be as minor as possible.

2D parameter optimization of nanobricks with respect to the relative polarized reflection efficiency (RPRE) is also analyzed at the target wavelength $\lambda=625$ nm, which is used to analyze and compare the reflective efficiency of polarized light more clearly. RPRE denotes the differential polarized reflectance of l-polarized light and s-polarized light, calculated as $\mathrm{RPRE}=\left|R_l-R_s\right|$, simulated in FDTD software. The optimized size of the nanobrick is set as: $L = 180$ nm, $W = 100$ nm, $H = 50$ nm and $CS = 360$ nm after comprehensively considering RPRE and reflective spectra.

During the EBL fabrication process, manufacturing errors inevitably occur in the physical size of the nanobricks. As a result, the real-word electromagnetic characteristics of the unit structure deviate from the simulated results, leading to a decrease in the tunability of the metasurface to incident light. Thus, to investigate the impact of deviations in the parameter of unit cells on the electromagnetic characteristics of OFDM-assisted key management, the unit structures with different parameters in length, width, height and period were modeled and simulated. The error tolerance performance in the dimensional parameters is displayed in Table S2. The three views and intensity alteration, sample images captured by scanning electron microscope are shown as Fig S3.

Table S2. Error tolerance simulated by both CST and FDTD software

Software	Parameter	Fabricating errors (l)	Fabricating errors (s)	Error tolerance
CST	Length	5 nm	10 nm	± 5 nm
	Width	10 nm	10 nm	± 10 nm
	Height	10 nm	10 nm	± 10 nm
FDTD	Length	10 nm	10 nm	± 10 nm
	Width	10 nm	10 nm	± 10 nm
	Height	10 nm	10 nm	± 10 nm

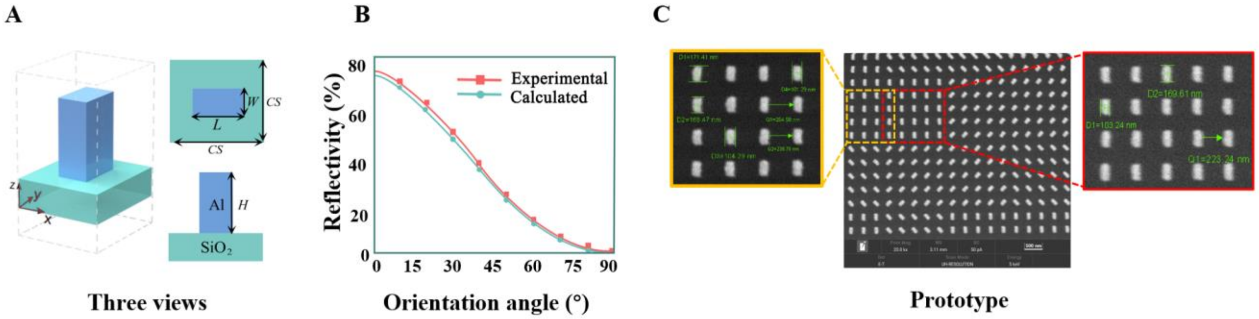


Fig S3. (A) Three views of unit cell. (B) Intensity alteration in terms of angle. (C) Measured length, width and period of the unit cell in the fabricated sample OFDM-assisted key management.

Considering the factors such as the actual bonding strength of aluminum, peeling difficulty, and the resolution of EBL, we suppose that the samples with parameter errors within 10 nm can achieve desired wavefront modulation effects. The errors of L, W and CS of the fabricated sample are approximately maintained no more than 10 nm, meeting the imaging and design requirements.

S3.3 Nanorod orientation design for dual-channel display

Guaranteeing the independent storage but unified management of keys, two polarization responses are multiplexed in terms of Malus's degeneracy. Thus, Jones matrix presentation is derived to quantify the orientation angle of nanobricks according to the public channel and private channel. In the reflectively optical path, a bulk polarizer is set in front of the light source and a bulk analyzer is set between OFDM-assisted key management and an eyepiece. Therefore, the Jones matrix derivation can be expressed as Eq. S2.

$$E = A_{\text{in}} \cdot \begin{bmatrix} \cos^2\alpha_2 & 1/2\sin(2\alpha_2) \\ 1/2\sin(2\alpha_2) & \sin^2\alpha_2 \end{bmatrix} \begin{bmatrix} \cos^2\theta & 1/2\sin(2\theta) \\ 1/2\sin(2\theta) & \sin^2\theta \end{bmatrix} \begin{bmatrix} \cos\alpha_1 \\ \sin\alpha_1 \end{bmatrix} \qquad \text{(S2)}$$

where A_{in} indicates the amplitude of incident light, θ represents the orientation angle of a nanobrick and α_1 as well as α_2 are noted as the polarization direction of the polarizer and the analyzer respectively. Eq. S2 experiences a mathematical matrix derivation and the reflected light intensity can be expressed as follow:

$$I = A_{\text{in}}^2 \cdot \left[\cos(\theta-\alpha_1)\cos(\alpha_2-\theta)\right]^2 \qquad \text{(S3)}$$

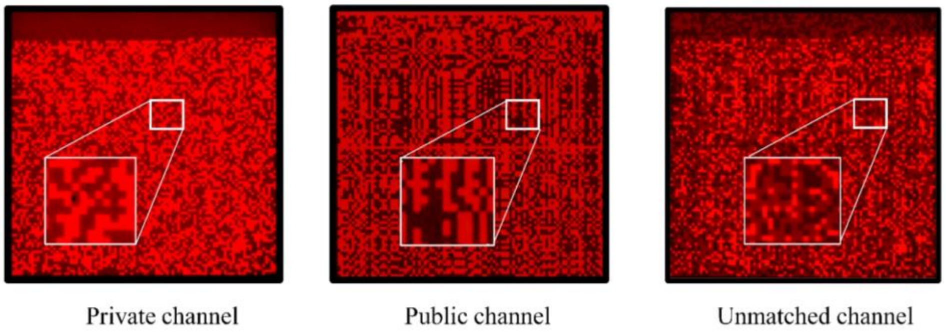
The calculated intensity of the output light after passing through the analyzer for private channel ($\alpha_1 = 45°, \alpha_2 = 90°$) and public channel ($\alpha_1 = 135°, \alpha_2 = -90°$) are displayed in Eq. S4. The four angle candidates of θ are $0°, 45°, 90°$ and $135°$, representing negative of both private channel and public channel ("00"), positive of private channel but negative of public channel ("10"), positive of both private and public channel ("11"), and negative of private channel but positive of public channel ("01").

$$\begin{cases} I_1 = A_{\text{in}}^2 \cdot \left[\cos\left(\theta - \dfrac{\pi}{4}\right)\cos\left(\dfrac{\pi}{2} - \theta\right)\right]^2 \\ I_2 = A_{\text{in}}^2 \cdot \left[\cos\left(\theta - \dfrac{3\pi}{4}\right)\cos\left(-\dfrac{\pi}{2} - \theta\right)\right]^2 \end{cases} \qquad \text{(S4)}$$

S4. Supplemented details of experiments of the proposed scheme

This chapter supplements several experimental details of multi-user SPI secure framework with metasurface. To recover $\mathbf{I}_{diffuse}^{3m\times 3n}$ as accurately as possible, we first transform the 8-bit-per-pixel (8BPP) image $\mathbf{I}_{diffuse}^{3m\times 3n}$ into binary form in terms of normal decimal-to-binary regulation (i.e., $255 \rightarrow 11111111$). Then we segment the transformed $\mathbf{I}_{diffuse}^{3m\times 3n}$ into multiple blocks and illuminate each block five times by Fourier pattern. Finally, we optimize the detected intensity signal by the similar approaches shown in Section S5.2, and apply iterative gradient optimization to postprocess the detected images for recovering $\mathbf{I}_{diffuse}^{3m\times 3n}$ more accurately. Note that the blocks have already been large enough for the users to precisely recognize the recovered block by naked eyes so the optimized image can be completely recovered.

For decryption, each user needs to access OFDM-assisted key management system to decode their keys and further retrieve their own plaintext and sub-token. The entire experimental setup of OFDM-assisted key management is based on BA310MET-T microscope, in which the polarizer, analyzer, CMOS, light source and filter are integrated into a single unit during the production. When $\alpha_1 = 45°, \alpha_2 = 90°$, the private channel is expressed, whereas the public channel can be displayed when $\alpha_1 = 135°, \alpha_2 = -90°$. Because of the fixed polarizer in position ④ in Fig 8B in the maintext without any available orientation in microscope, the sample needs to be manually oriented to $45°$ and $135°$ to simulate the orientation of polarizer. The private channel, public channel and unmatched channel are displayed as Fig S4. The CMOS Moticam Pro6 is placed in position ①. A red-light filter with a ground glass is added in position ⑤, to uniformize the incident light distribution with more accurate wavelength. The view of the public channel, private channel and unmatched channel are presented in Fig S4.


Private channel

Public channel

Unmatched channel

Fig S4. Display of three channels under the view of the microscope.

S5. Deep differential attack (DDA)

S5.1 Cryptanalysis background of DDA

The background of cryptanalysis against SPI cryptographical keys is described in Fig S5. Alice intends to send "*Peppers*" (i.e., plaintext $\mathbf{m}$) with $N \times N$ size to Bob secretly through an open scattering channel, where both Charlie and Bob can receive the intensity signal. Alice utilizes a digital encryption method $\mathcal{F}(\cdot)$ (i.e., permutation or scrambling) to encrypt patterns $\mathbf{P}$ as cryptographical keys $\mathbf{P}^* = [\mathbf{p}_1^*; \mathbf{p}_2^*; ...; \mathbf{p}_{N^2}^*]$ in terms of a permutation key $\mathbf{v}$, which is noted as $\mathbf{P}^* = \mathcal{F}(\mathbf{P}|\mathbf{v})$. Further, $\mathbf{P}^*$ are utilized to interact with $\mathbf{m}$, so that encrypted light intensity $\mathbf{c}$ can be broadcasted:

$$\mathbf{c} = \mathbf{P}^*\mathbf{m} + \mathbf{n} = \mathcal{B}(\mathcal{F}(\mathbf{P}|\mathbf{v})) \tag{S5}$$

During the process, $\mathbf{P}$ is in public but there is a private channel for Bob and Alice to share the transcript so that Bob can recover "*Peppers*" by following the equation: $\mathbf{m} = \mathcal{F}(\mathcal{B}^{-1}(\mathbf{c}|\mathbf{P})|\mathbf{v})$. But for Charlie knowing the public $\mathbf{P}$ only, the plaintext definitely cannot be recovered. In the open environment, there also exists an Eve able to access the oracle as a prior assumption to analyze arbitrary plaintext-ciphertext pairs based on chosen plaintext attack (CPA) background. For the deep differential attack against the multiuser SPI cryptography framework with the OFDM-assisted key management, the security and capacity are assessed in terms of external security and internal privacy respectively, where the attack model is conducted by Eve and an internal user respectively. For Eve, the plaintexts and keys are protected by the Fourier SPI encryption with complete OFDM-assisted key management, whereas those are only preserved by regional encryption with OFDM-like coding and RSA encryption for hostile users.

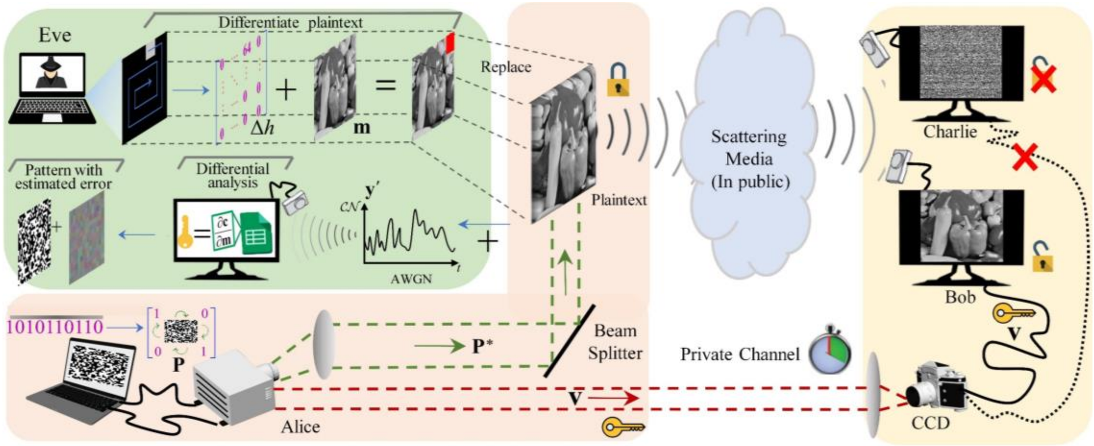


Fig S5. Schematic of DDA under CPA framework.

S5.2 Principles of DDA

First and foremost, the differential attack is used to analyze patterns. For different encryption methods, the differential analysis is different in terms of the specific steps. Here, we only present the fundamental cracking principle against SPI cryptographical keys. When a difference occurs on a specific pixel of $\mathbf{m}$, the value of the pattern at the corresponding position can be reflected by the degree of change in $\mathbf{c}$. Thus, the parameters of encrypted pattern $\hat{\mathbf{P}}^*$ (i.e., Jacobian matrix) can be calculated by taking partial derivatives with respect to the $\mathbf{m}$ in terms of Eq. S6. When $\hat{\mathbf{P}}^*$ is obtained, Eve can directly recover the plaintext by simply following the SPI reconstruction principle: $\hat{\mathbf{m}} = \mathcal{B}^{-1}(\mathbf{c}|\hat{\mathbf{P}}^*)$.

$$\hat{\mathbf{P}}^* = \nabla_{\mathbf{m}}\mathbf{c} = \frac{\partial \mathbf{c}}{\partial \mathbf{m}} \tag{S6}$$

When encryption algorithms are so complex that $\hat{\mathbf{P}}^*$ greatly deviate from the original, deep learning is required to mitigate the distortion, attempting to establish a connection of key management and correct keys. However, for an image with no fixed pixel distribution and little correlation between pixels (i.e., $\eta \approx 0$), extracting features from it in a targeted manner is challenging. Thus, we artificially extend the width of each symbol (i.e., impulse) at a certain replication rate (i.e., here, a threefold replication with 100% duty cycle is used), the correlation between each binary pixel can be reestablished, generating a random On-off keying (OOK) signal, as depicted in Fig S6, which is quite common in aerospace applications.

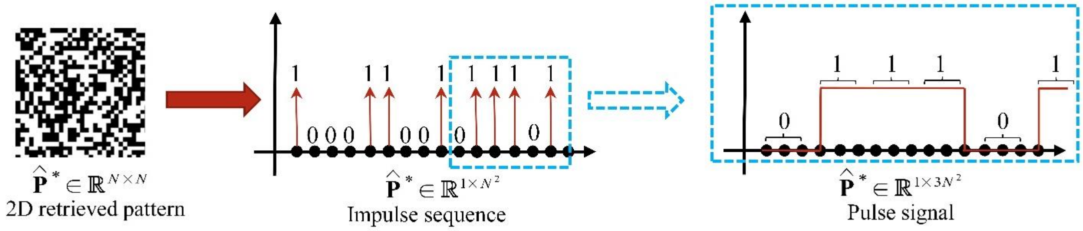

1
1 1
1 1 1 1
1
0 0 0
0 0
0
0
1 1 1
1
0
0
$\hat{\mathbf{P}}^* \in \mathbb{R}^{N\times N}$
2D retrieved pattern
$\hat{\mathbf{P}}^* \in \mathbb{R}^{1\times N^2}$
Impulse sequence
$\hat{\mathbf{P}}^* \in \mathbb{R}^{1\times 3N^2}$
Pulse signal

Fig S6. Transformation of a 2D pattern into a 1D pulse signal.

In this way, the denoising issue of pseudo-random binary patterns is transferred into an optimization problem for normal pulse signals, which can be addressed by kinds of digital signal processing methods. Moreover, since DDA has a clear mathematical foundation, as described in Eq. S6, we can derive the error distribution generated at each step. Therefore, computer simulations can be conducted to generate the key dataset following different error distribution in advance.

The initial errors are approximated as Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$ with mean μ and variance σ^2. At the same time, to quantitatively elucidate the facilitating effect of pulse width extension on patterns recovery under different encryption depth, we employ signal-to-noise ratio (SNR) of Gaussian noise for results visualization. For instance, for the pure SPI encryption without any other process on $\mathbf{P}$, the generated errors are derived as:

$$\Delta\mathbf{n} = \frac{\mathbf{n}_1 - \mathbf{n}_2}{\Delta h} \sim \mathcal{N}\left(\frac{\mu_1 - \mu_2}{\Delta h}, \frac{\sigma_1^2 + \sigma_2^2}{\Delta h}\right) \tag{S7}$$

After simulating the noised patterns for network training, we can further use a sequential network to recover distorted patterns. Fig S7 illustrates the optimized results of $\tilde{\mathbf{P}}^*$ tested under different SNR after 50 iterations in one-dimension and two-dimension, respectively. The displayed signal profile is represented in terms of the average recovering accuracy after 1000 Monte Carlo experiments. It can be observed that the optimization effect on $\tilde{\mathbf{P}}^*$ becomes poorer with the decreasing of SNR. When $\text{SNR} = 10\,\text{dB}$, the 0/1 inversion occurs. However, the general envelope can still be recovered, indicating that the DDA is still applicable in this situation for several encryption methods not extremely sensitive to the alteration of keys (i.e., revealed by NPCR and UACI). When analyzing $\hat{\mathbf{P}}^*$ under $\text{SNR} = 20\,\text{dB}$ or higher, our method is capable of recovering relatively accurate patterns, thus leading to more favorable decrypting results.

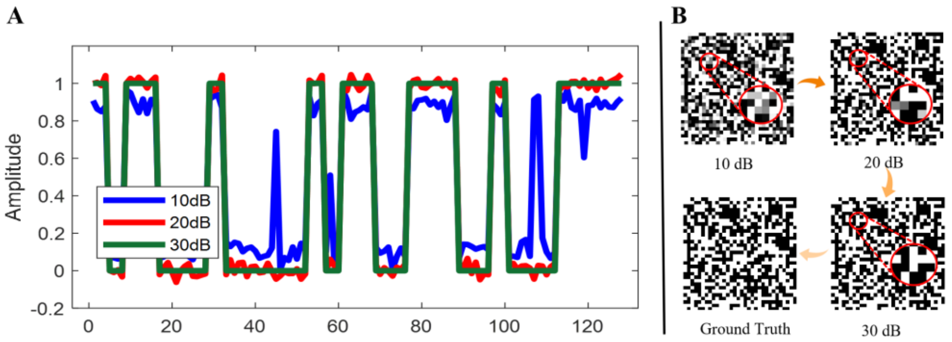


Fig S7. Optimized results of patterns in (A) 1D domain and (B) 2D domain.

Fig. S8 also provides a detailed comparison of cracked plaintext achieved by DDA when it comes to different reconstructing error, which is simulated in terms of SNR. The result show that even with the pattern optimization by deep learning, the recovering effect goes worse as the SNR decreases.

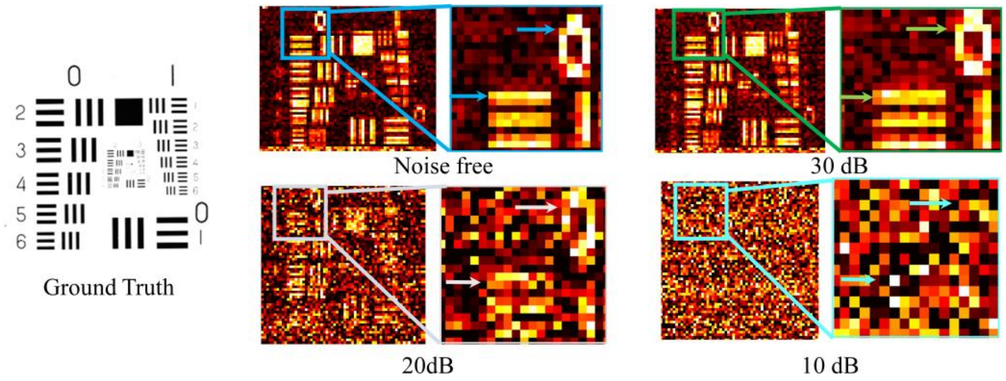


Fig S8. Detailed comparison of DDA.