34TH USENIX SECURITY SYMPOSIUM

蚂蚁集团 ANT GROUP | 数字科技

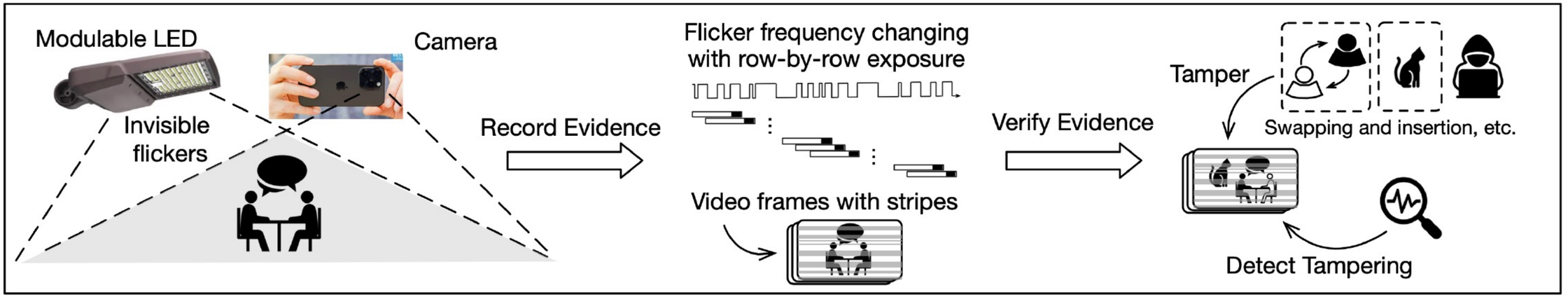
RollingEvidence: Autoregressive Video Evidence via Rolling Shutter Effect

Feng Qian*†, Lingfeng Zhang*, Tao Luo, Shiqi Xu, Zhijun Yu, and Wei Wang

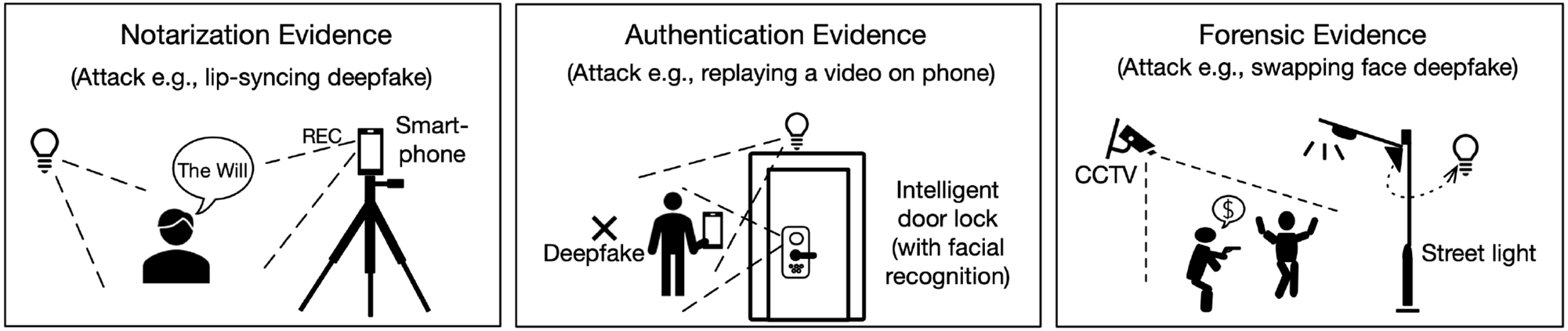
Ant Group

{*youzhi.qf, zhanglingfeng.zlf, xiyu.lt, xushiqi.xsq, poli.yzj, wei.wangwwei*}@antgroup.com

Overview

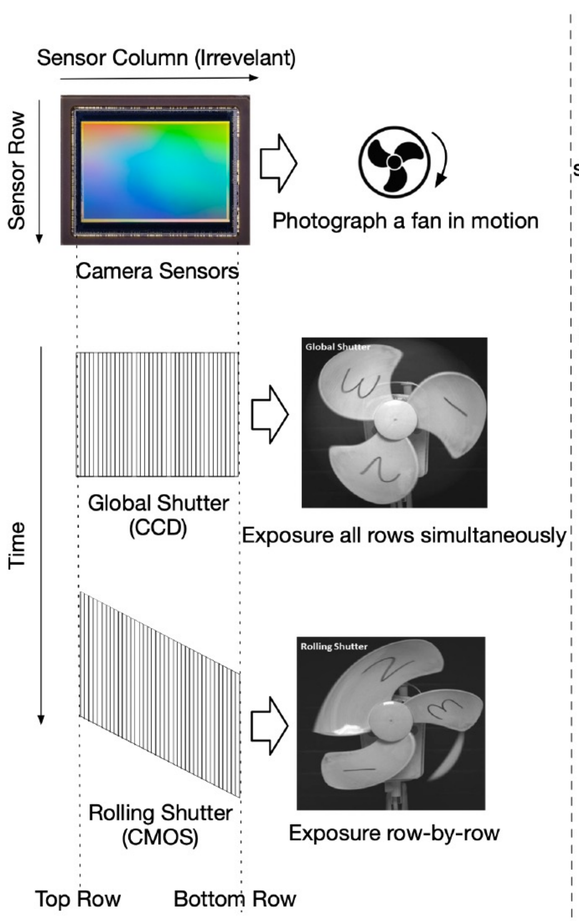


RollingEvidence: Rolling shutter effect as evidence encoding channel

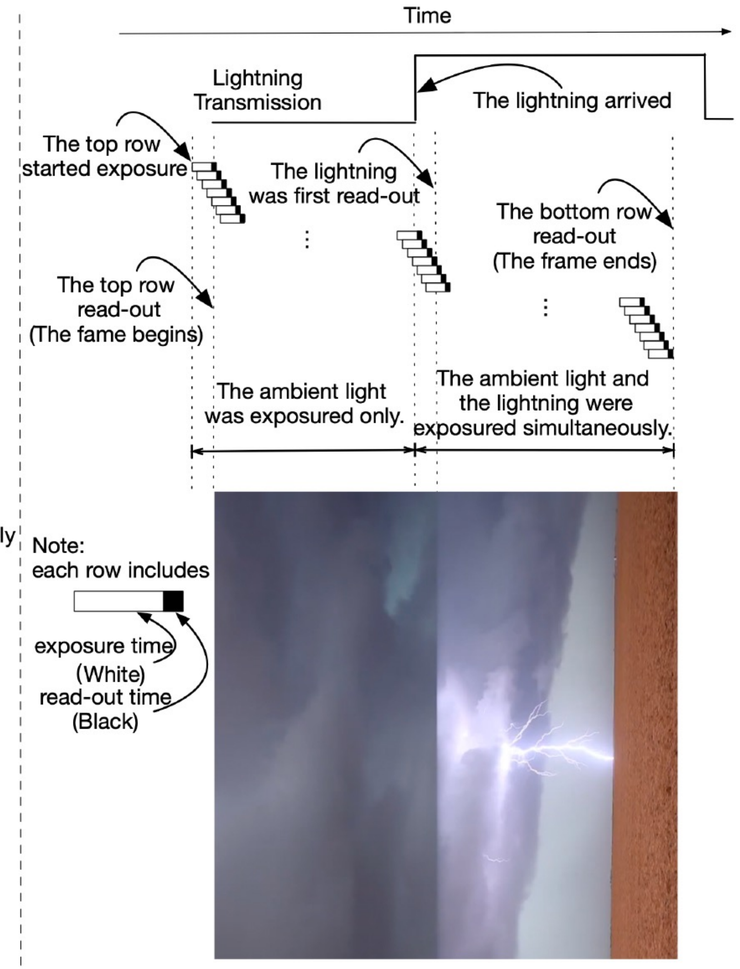


Application scenarios of RollingEvidence

Embedding Probes via Rolling Shutter Effect



(a) Rolling shutter effect on fan motion (spatial aliasing)



(b) Rolling shutter effect on lightning (temporal aliasing)

By utilizing RS's temporal aliasing,

$$I(x,y) = \Psi_{pe}\left((L_1 + L_2(y)) \cdot J(x,y)\right)$$

$$L_2(y) = \int_{t_{y-1}}^{t_y} L_{LED} \cdot F(t) \cdot \varphi_{rs}(t; T_r, T_e)\, dt$$

we autoregressively embed probes into videos.

Two types of rolling shutter effect (RS): (a) spatial aliasing and (b) temporal aliasing.

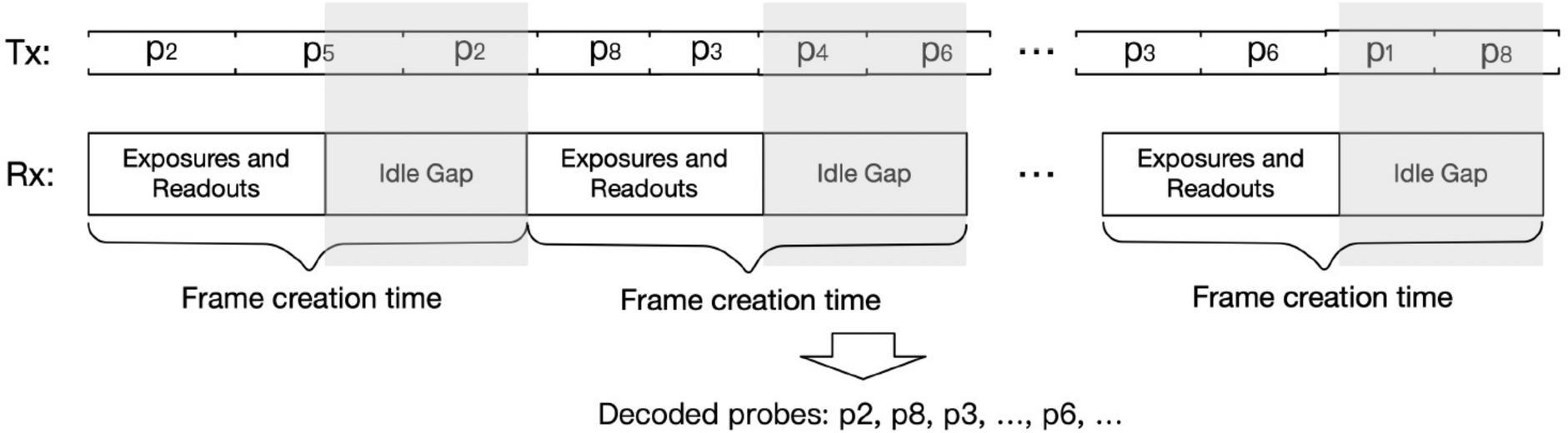
(The images are adopted from https://www.youtube.com/watch?v=4IUgiB4Z6CI and https://www.youtube.com/watch?v=8FvwhFuifaQ)

Probe Package Design

RollingEvidence adopts frequency-shift keying (FSK) with probes surrounded by splitter frequency

E.g., $\ldots f_{sp} p_1 f_{sp} p_2 f_{sp} \ldots$ with $p_1 = [f_0, f_{15}, f_3]$ and $p_2 = [f_{12}, f_9]$ $(f_i \in \mathcal{F}$ and $p_j \in \mathcal{P})$

We disregard concerns about communication issues, this allows for a more compact and high-dimensional probe definition. We employ $|\mathcal{P}|$ = 4,096 (frequency permutations from unigram to four-gram) with $|\mathcal{F}|$ =16.



Light Frequencies for Different Cameras

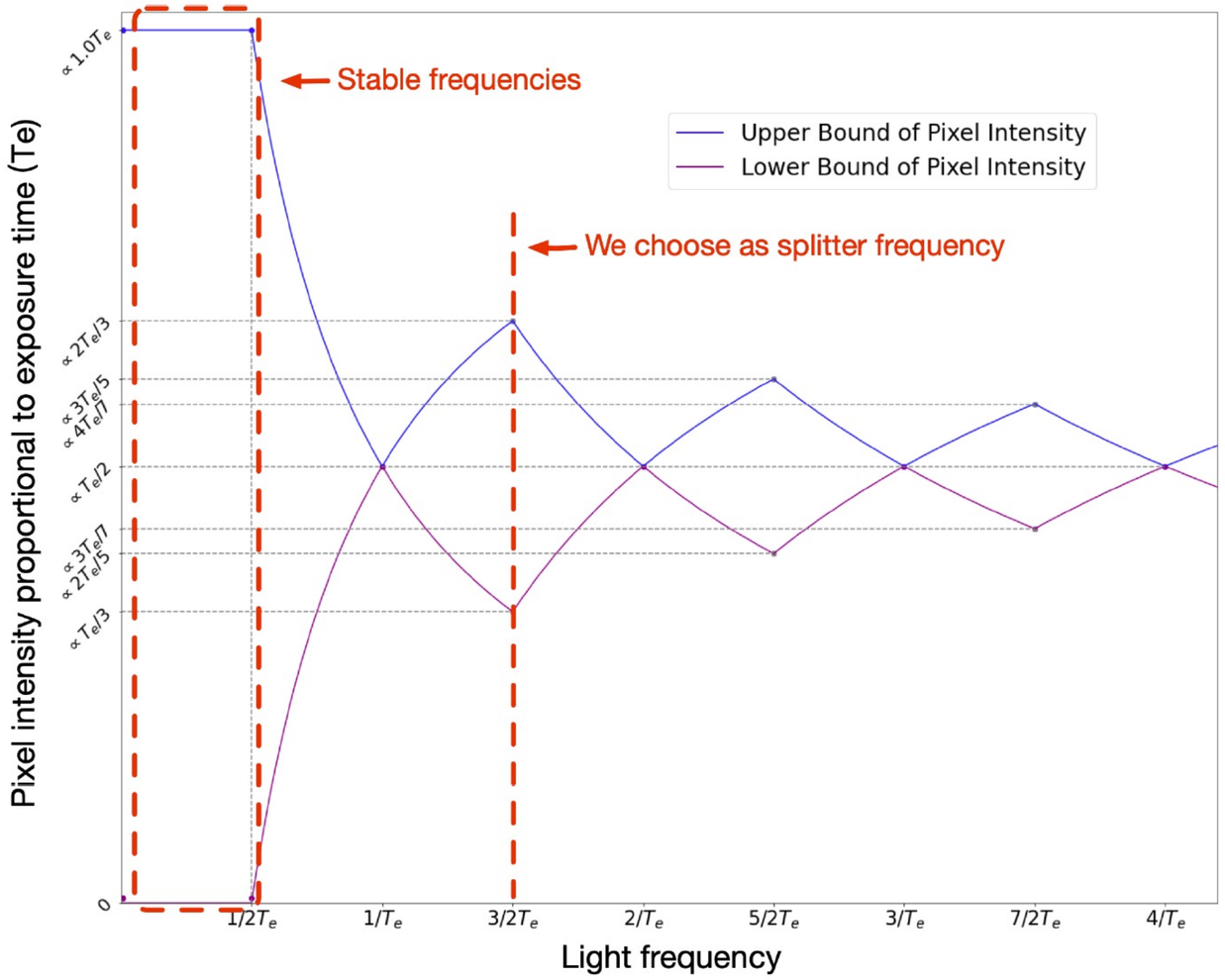
For cameras with different readout times, we fix the exposure time and adjust frequency dictionary in setup, ensuring consistent stripe patterns for deep network.

$$\mathcal{F} = \left\{ f_i = \frac{1}{T_r w_i} \,\middle|\, w_i = w_0 + 5i \right\}_{0 \le i < L}$$

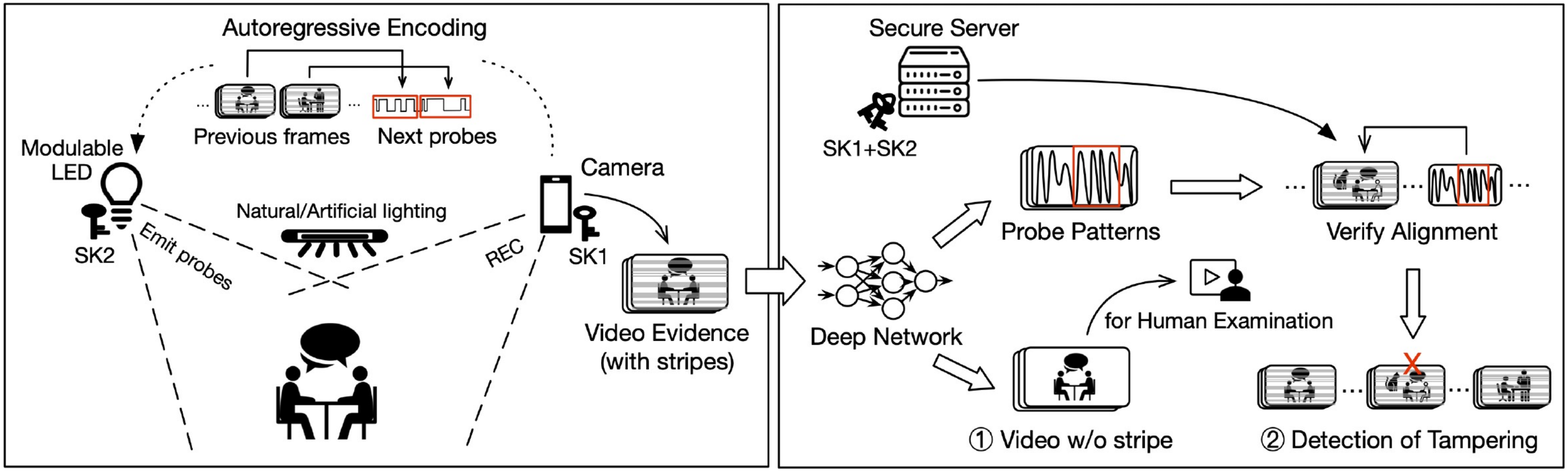
(in prototype system, L=16 and w_0=100)

The upper bound and lower bound of stripe pixel intensity (proportion to exposure time) with light frequency.

$$B_{upp} \propto \begin{cases} T_e, & 0 < f \le \dfrac{1}{2T_e}, \\ \max\left(\dfrac{k+1}{2f}, T_e - \dfrac{k+1}{2f} \right), & \dfrac{2k+1}{2T_e} < f \le \dfrac{2k+3}{2T_e}. \end{cases}$$

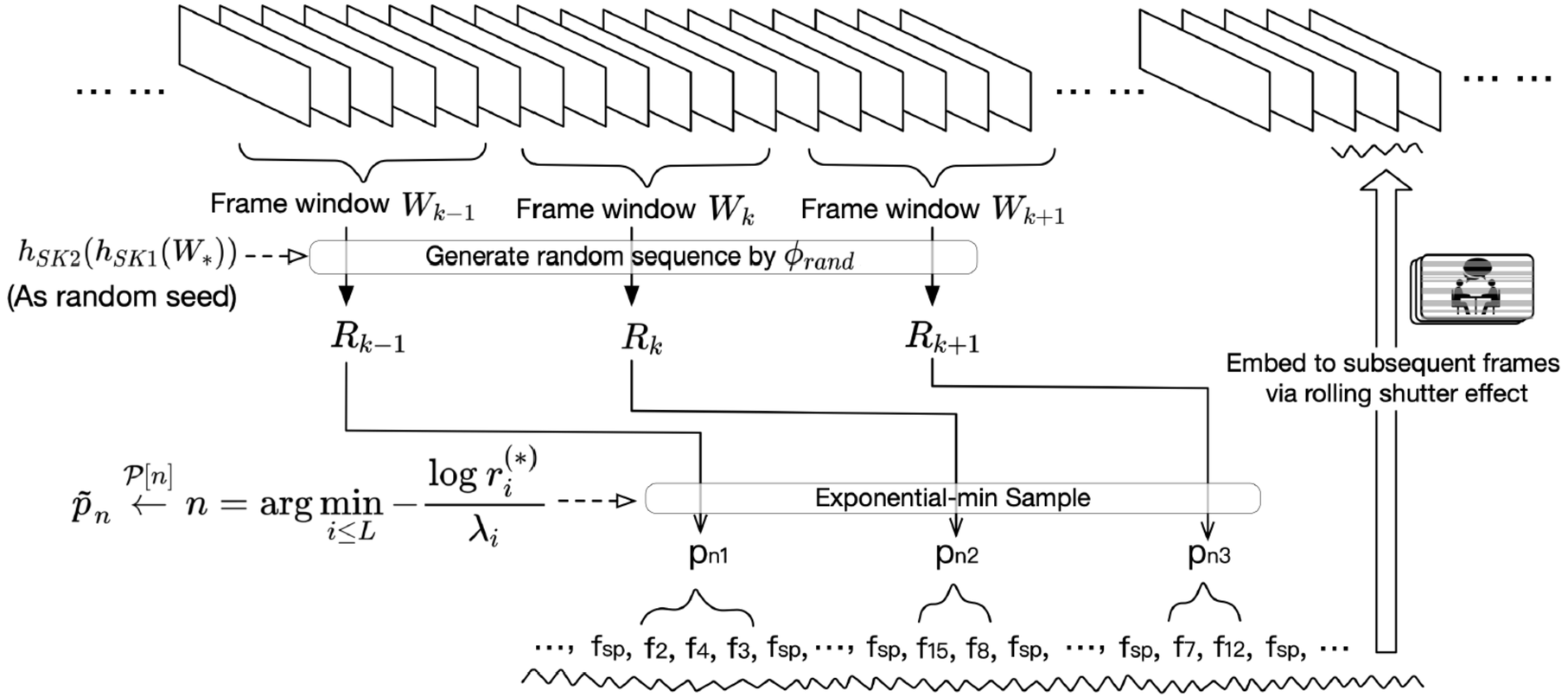


Two-step work processes of RollingEvidence



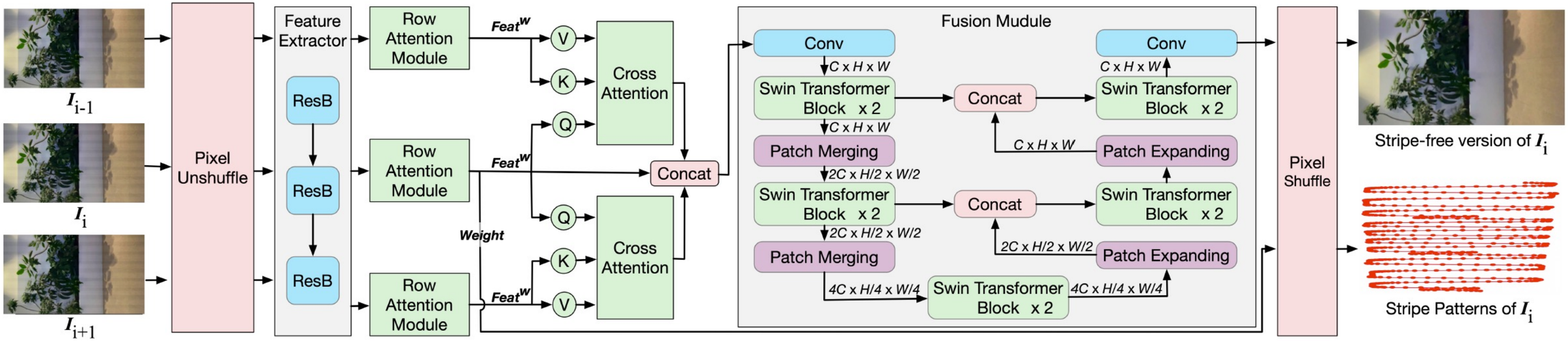
Recording (left) and verification (right).

Autoregressive and Stochastic Encoding of Probes

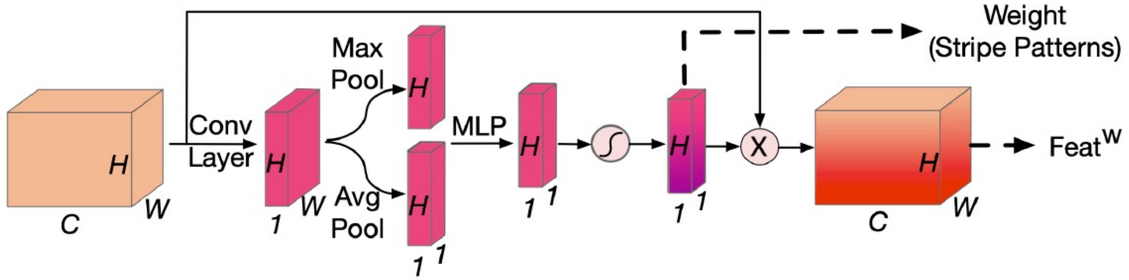


Forward encoding: use previous frame windows to encode subsequent frames.

Neural Network for Stipe Extraction and Elimination

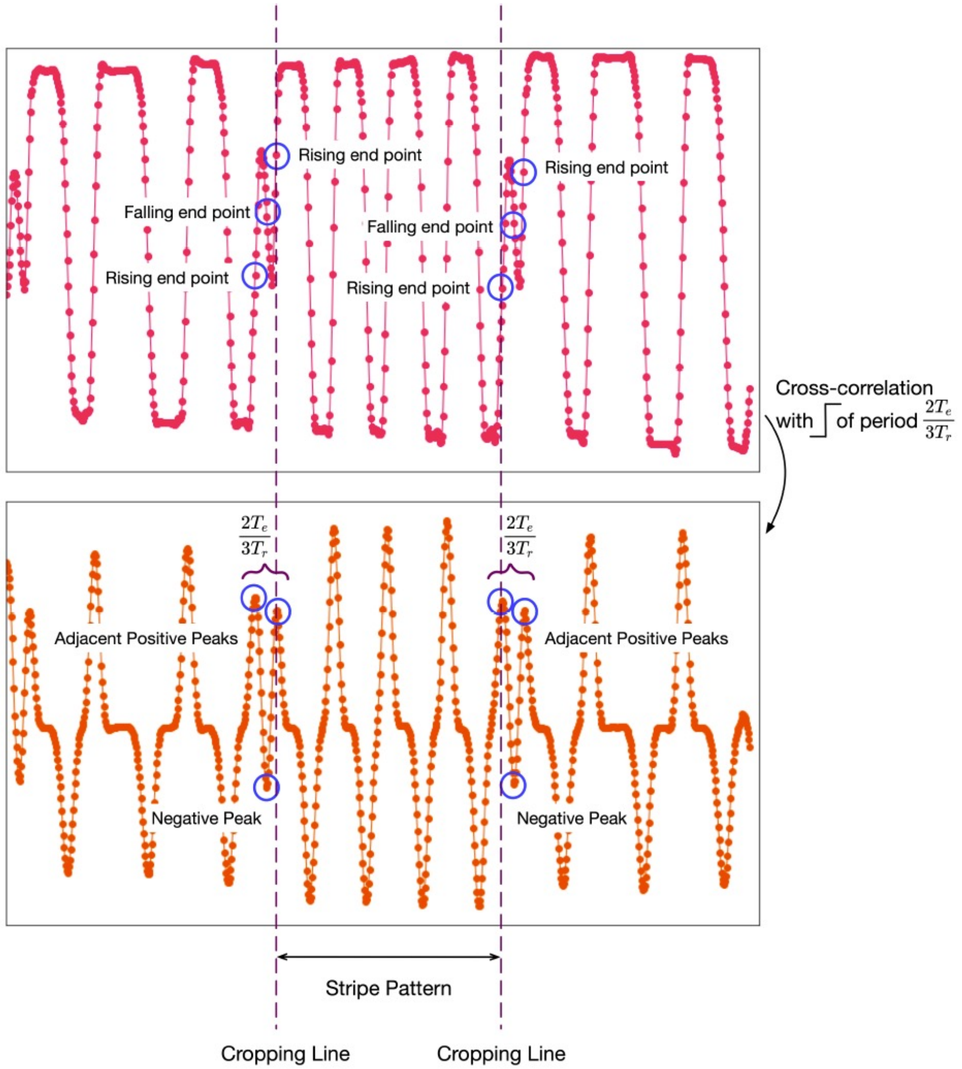


Network architecture for extracting stripe intensity curve and simultaneously producing a stripe-free frame.



Row attention module.

Stripe Pattern Cropping and Probe Decoding

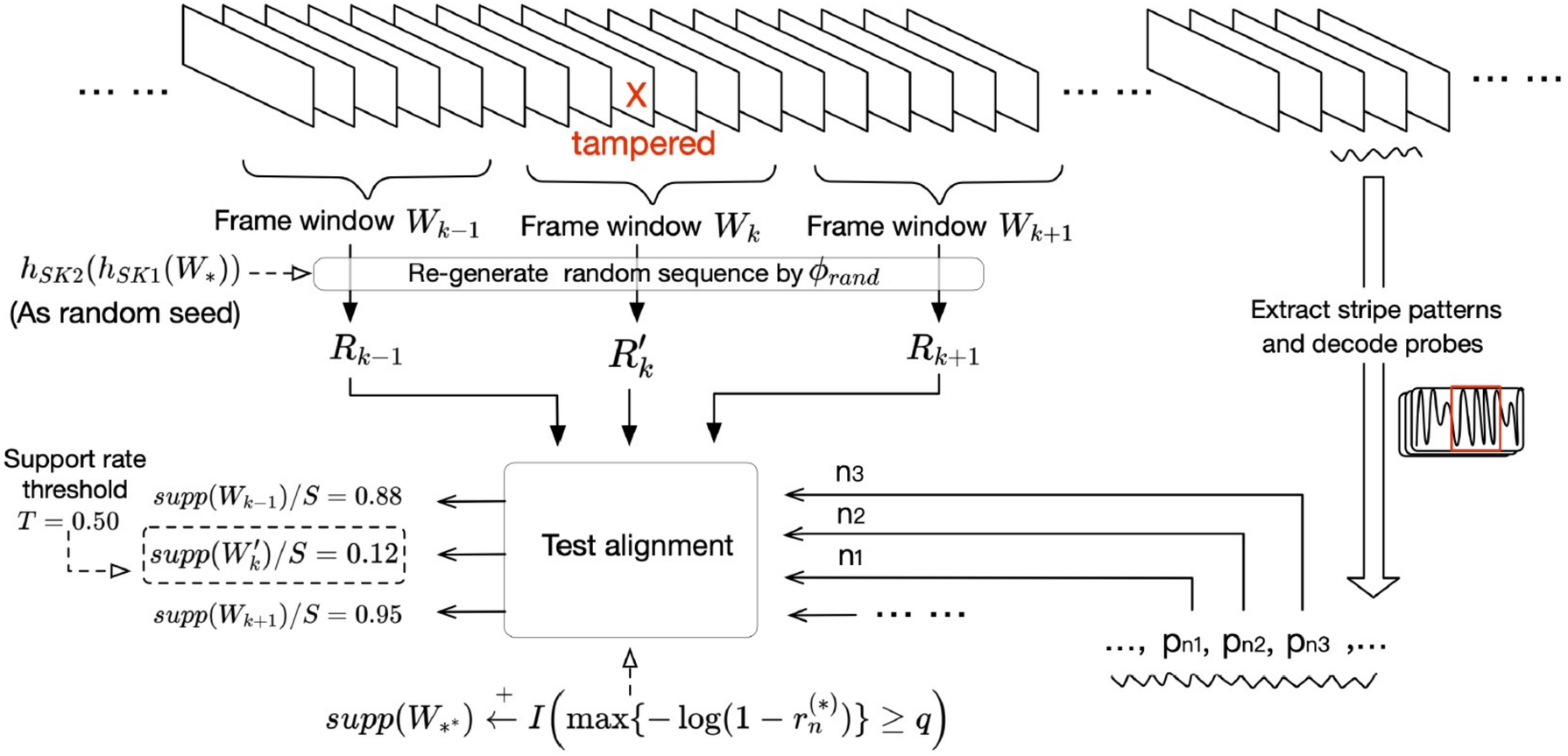


Determining the crop lines for stripe pattern.

Stripe Pattern Cropping. Starting with the extracted intensity curve, we perform a cross-correlation with splitter pattern (a square wave, see Figure 7). The AMPD algorithm [85] is applied to find local peaks. The distance between each negative peak and its adjacent positive peaks near $2T_e/3T_r$ defines a splitter pattern for determining vertical cropping lines.

Probe Decoding. To decode the cropped stripe patterns into probes, we use ResNet34 [29] for multi-class classification by replacing 2D convolutions with 1D and using cross-entropy loss. Trained on a prelabeled dataset, this classifier assigns each stripe pattern to one of 4096 categories, decoding at least one probe per frame according to its maximum width design.

Tampering Detection with Exponential-min Implication



Backward detection: decoded probes provide support to previous frame windows.

Theoretical Limits on Tampering Significance

Theorem 3.1. *The expected value of the random variable r_n, dependent on the arbitrarily sampled probe $\tilde{p}_n$, is $\frac{1}{2}$. In contrast, the expected value of r_n with respect to the selected probe $\tilde{p}_n$, as delineated in Formula 6, lies within the interval $\left[\frac{1}{2}, \frac{L}{1+L}\right]$. Given that λ_i is independently sampled from $\mathcal{U}(0,1)$ and subsequently normalized, for sufficiently large L, the empirical expectation of r_n approximates 1.*

Theorem 3.2. *Given the frame window size S and the average number of probes Q embedded in a single frame, we can get the following results with sufficiently large SQ: (1) The mean of random values $\overline{R_k}$ obtained from an individual frame window (e.g., the kth frame window) using an arbitrarily sampling schema follows a Bates distribution with $E[\overline{R_k}] = \frac{1}{2}$ and $Var[\overline{R_k}] = \frac{1}{12SQ}$; (2) Using a selection schema according to Formula 6, the mean value approximately follows a Gaussian distribution with $\frac{1}{2} \leq E[\overline{R_k}] \leq \frac{L}{1+L}$ and $E\left[Var[\overline{R_k}]\right] < \frac{3}{4SQ}$, which is a tight bound with large SQ.*

Theoretical bounds for detecting abnormalities.

Threat Model

Security Assumptions: We focus on addressing security concerns within our system by reasonably assuming that cryptographic keys and nonces are securely distributed from the server to the devices and they cannot be stolen from the server.

Attacker Capabilities: We assume that the attacker is highly skilled and has substantial computational resources to use deep learning to extract stripes from genuine videos and inversely synthesize stripes and fuse them into fake videos.

Attack Goal: Manipulate existing videos (AI-generated or third-party videos) or tamper with RollingEvidence-captured videos using forgery techniques to evade verification.

Attack Scenario I: Both devices are safe.

- attack vector (a): insertion, deletion, and modification
- attack vector (b): alter fake frames to get same digests
- attack vector (c): copy stripes into fake frames
- attack vector (d): guess SK1 and SK2

Attack Scenario II: The LED is compromised.

- attack vector (e): already know SK2, guess SK1

Attack Scenario III: The camera is compromised

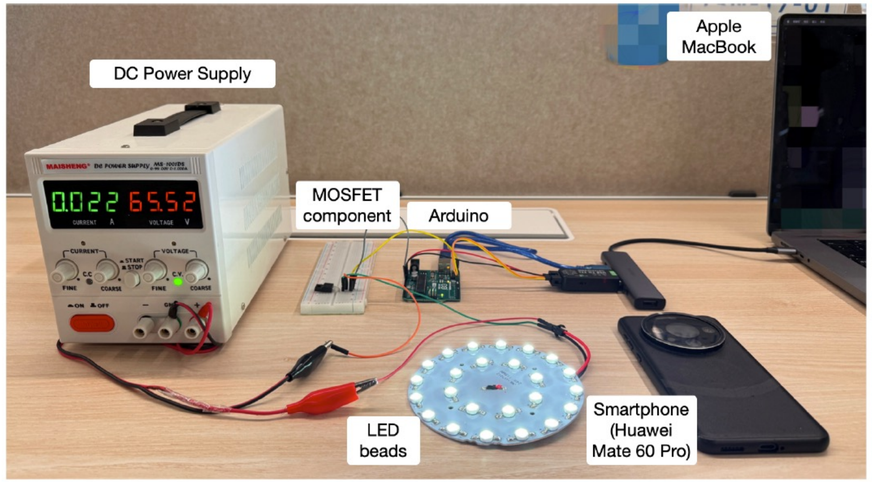
- attack vector (f): already know SK1, spoof the LED (by copying stripes from genuine frames into forged frames, and generating fake probes based on manipulated frames).

Threat Model

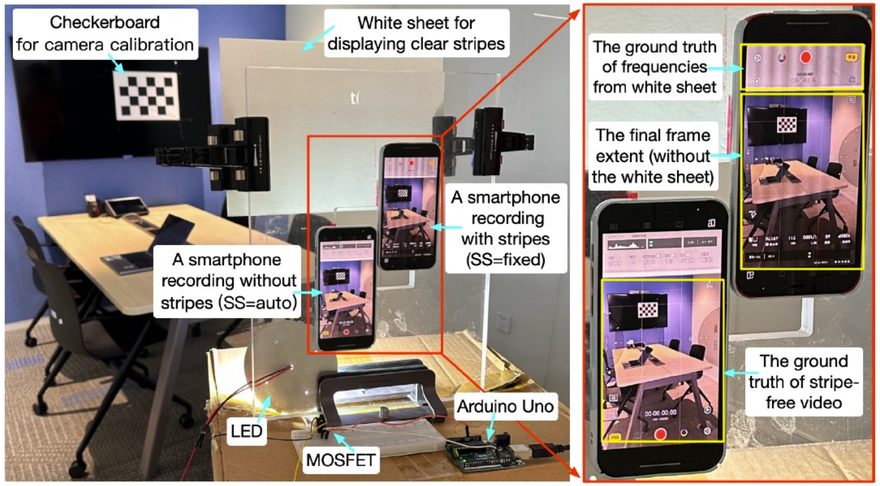
System Defense Summary:

- For attack vectors (a), (b) and (c), the difficulty matches that of carrying out a preimage attack on SHA-256 or comparable security level algorithms with a resistance strength of 128 bits.
- For attack vectors (d) and (e), the brute force search for the size of key space of 2^{256} is needed.
- The attack vector (f) requires real-time operations synchronized with the LED, hence cooperating a server-side timestamp nonce in probes can prevent a previously recorded from being replaced by a newly filmed one

Prototype System and Datasets



(a) The equipment for the RollingEvidence prototype.



(b) The equipment setup for self-labeled dataset (with a zoom-in view of the red section).

Illustration of (a) RollingEvidence prototype, and (b) equipment for recording self-labeled datasets.

No.	# of Samples	Dur. (mins)	Filmed Scenes	Camera Devices
T1[1]	12K	60	12 different scenes[2]	Huawei Mate 60 Pro and iPhone 12
T2	67K	70	Meeting room and Recreation area	Huawei Mate 60 Pro
D1[1]	16K	80	16 different scenes [2]	Huawei Mate 60 Pro and iPhone 12
D2	315K	180	Meeting room and Recreation area	Huawei Mate 60 Pro, Xiaomi 15 Pro and iPhone 12
D3	9k	2,400	16 different scenes [2]	Huawei Mate 60 Pro and Xiaomi 15 Pro
D4-1	1,560	270	Human faces with face-swapping	Huawei Mate 60 Pro
D4-2	840	210	Human faces with lip-syncing	Huawei Mate 60 Pro
D5-1	4,320	1,080	Meeting room	Huawei Mate 60 Pro
D5-2	3,840	960	Garden	Huawei Mate 60 Pro
D6	3,840	1,920	Meeting room	Huawei Mate 60 Pro
D7	3,840	960	Meeting room	Huawei Mate 60 Pro
D8	3,600	840	Recreation area	Huawei Mate 60 Pro

[1] These videos are recorded with the equipment shown in Figure 8b.
[2] Figure 9 shows these scenes (dataset T1 contains only first 12 scenes).

Description of all datasets used in experiments.

Performance of Tampering Detection

Type	Scene	Accuracy (%)	FRR (%)	FAR (%)
Static Scene	Recreation Area	99.70	0.00	0.40
	Meeting Room	99.71	0.00	0.39
	Plants	99.32	0.00	0.91
	Reception Desk	99.43	0.00	0.76
	Kitchen	99.84	0.00	0.22
	Wooden Rack	99.64	0.00	0.48
	Cubby Locker	99.62	0.00	0.51
	Lawn[1]	99.41	0.00	0.78
	Garden[1]	99.78	0.00	0.29
	Coffee Room	99.84	0.00	0.22
Dynamic Scene	Presentation	99.66	0.00	0.46
	Printing	99.44	0.00	0.75
	Desk Work	99.83	0.00	0.23
	Billboard[1,2]	99.31	0.00	0.92
	Break Room[2]	99.35	0.00	0.86
	Walking	99.42	0.58	0.58
Both	Total	99.59	0.04	0.53

[1] These videos are recorded outdoors.
[2] These videos involve camera motion during filming.

Detection performance of video tampering involving insertion, removal, and alteration operations.

Tamperer	Detector	ACC (%)	FRR (%)	FAR (%)
SimSwap	Rethinking	87.92/88.75	37.50/37.50	3.61/2.50
	AVFF	90.21/91.04	7.50/7.50	10.56/9.44
	RollingEvidence	100.00	0.00	0.00
E4S	Rethinking	88.54/87.08	37.50/37.50	2.78/4.72
	AVFF	92.71/91.67	7.50/7.50	7.22/8.61
	RollingEvidence	100.00	0.00	0.00

* In results marked by -/-, the number before slash refers to striped videos, and the number after to strip-free videos derived from deep network.

Face swapping detection results of various methods.

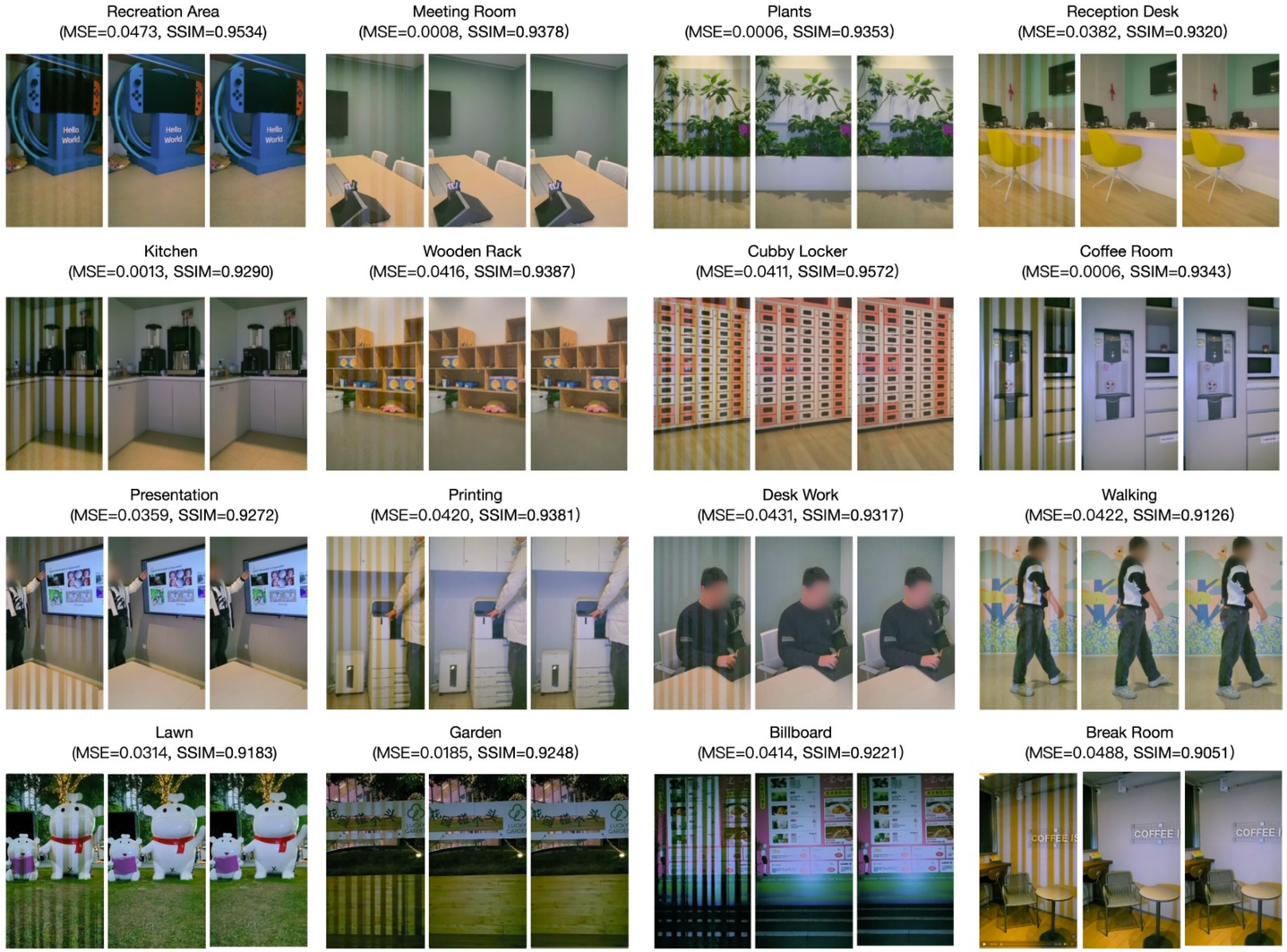
Tamperer	Detector	ACC (%)	FRR (%)	FAR (%)
LatentSync	Rethinking	82.28	37.50	11.02
	AVFF	93.67	7.50	5.93
	RollingEvidence	100.00	0.00	0.00
SadTalker	Rethinking	85.02	37.50	7.34
	AVFF	94.09	7.50	5.37
	RollingEvidence	100.00	0.00	0.00

Lip syncing detection results of various methods.

Camera Device	ACC (%)	Precision (%)	Recall (%)
iPhone 12	94.52	95.39	94.50
Xiaomi 15 Pro	96.43	97.24	96.40
Huawei Mate 60 Pro	97.55	98.48	97.54

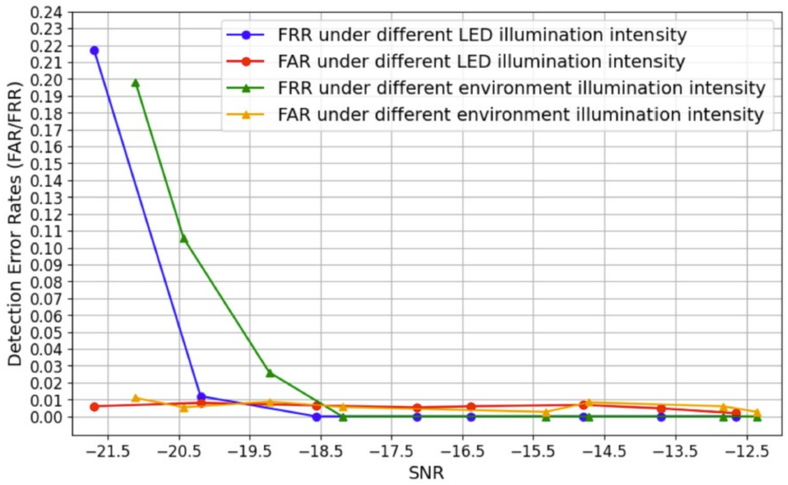
Deep network performance of probe decoding.

Performance of Verification Sub-modules

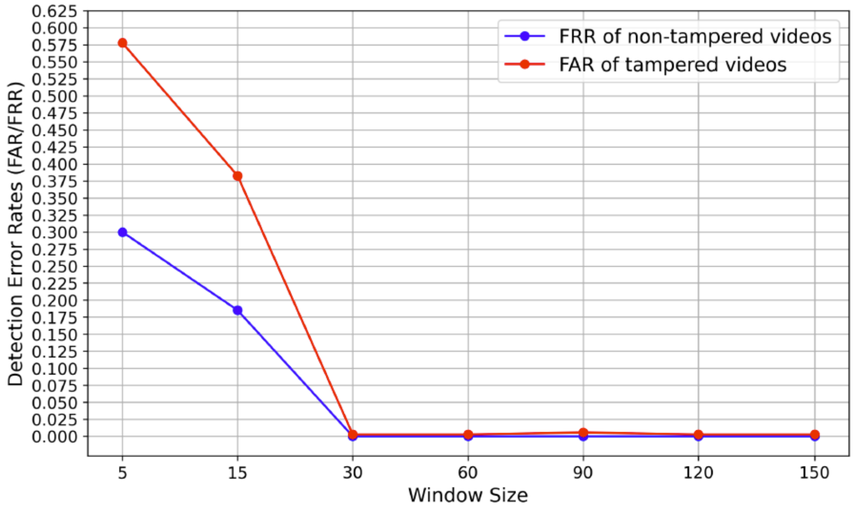


Performance for extracting intensity curves and generating stripe-free videos across 13 indoor and 3 outdoor locations (left: captured frames, middle: de-striped frames, right: ground truth); The last two scenes were captured with camera moving.

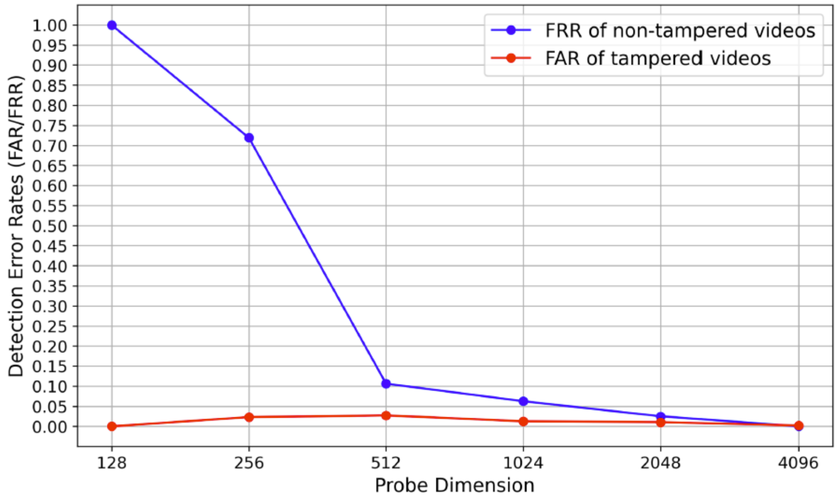
Impact of Parameters on System Performance



(a) Impact of signal-to-noise ratio (SNR) on system performance.



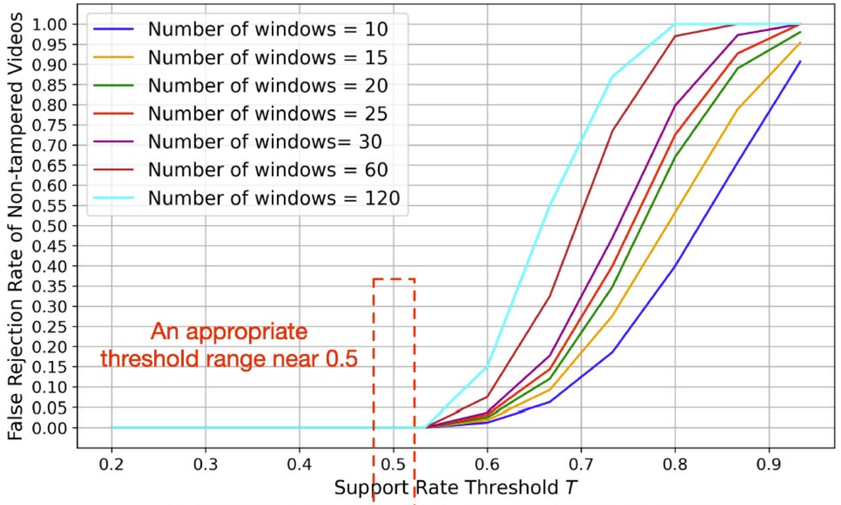
(b) Impact of window size on system performance.



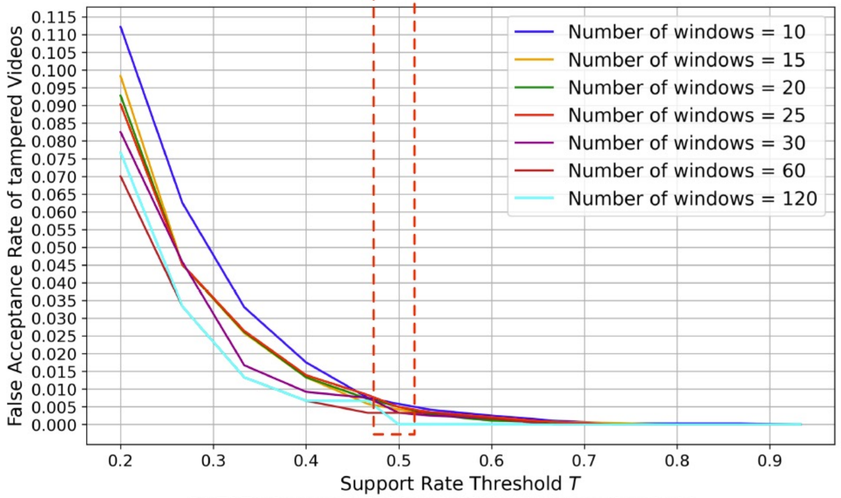
(c) Impact of probe dimension on system performance.

Impact of parameters on system performance.

Selecting Support Rate Threshold



(a) FRR of non-tampered videos with support rate threshold.



(b) FAR of tampered videos with support rate threshold.

Selecting an appropriate support rate threshold and the corresponding range of window numbers.

Conclusions

We summarize the contributions of our work as follows.

- We propose an active system that uses the rolling shutter effect to embed tamper-resistant probes in real-time video recording at the physical layer.
- We introduce an autoregressive encoding scheme that generates compact, high-dimensional probes using prior frames and device-specific keys for efficient tamper detection.
- We design a multitask deep network to extract embedded stripe patterns, decode probes, and detect tampered frames with theoretical and experimental validation.
- We implement a prototype and conduct extensive experiments to demonstrate the efficiency and strong security guarantees of our RollingEvidence framework.

Thanks!