The not-so-silent type

Breaking network crypto in almost every popular Chinese keyboard app

Jeffrey Knockel (Citizen Lab) and Mona Wang (Princeton University)

What’s a Chinese IME?



What’s a Chinese IME?



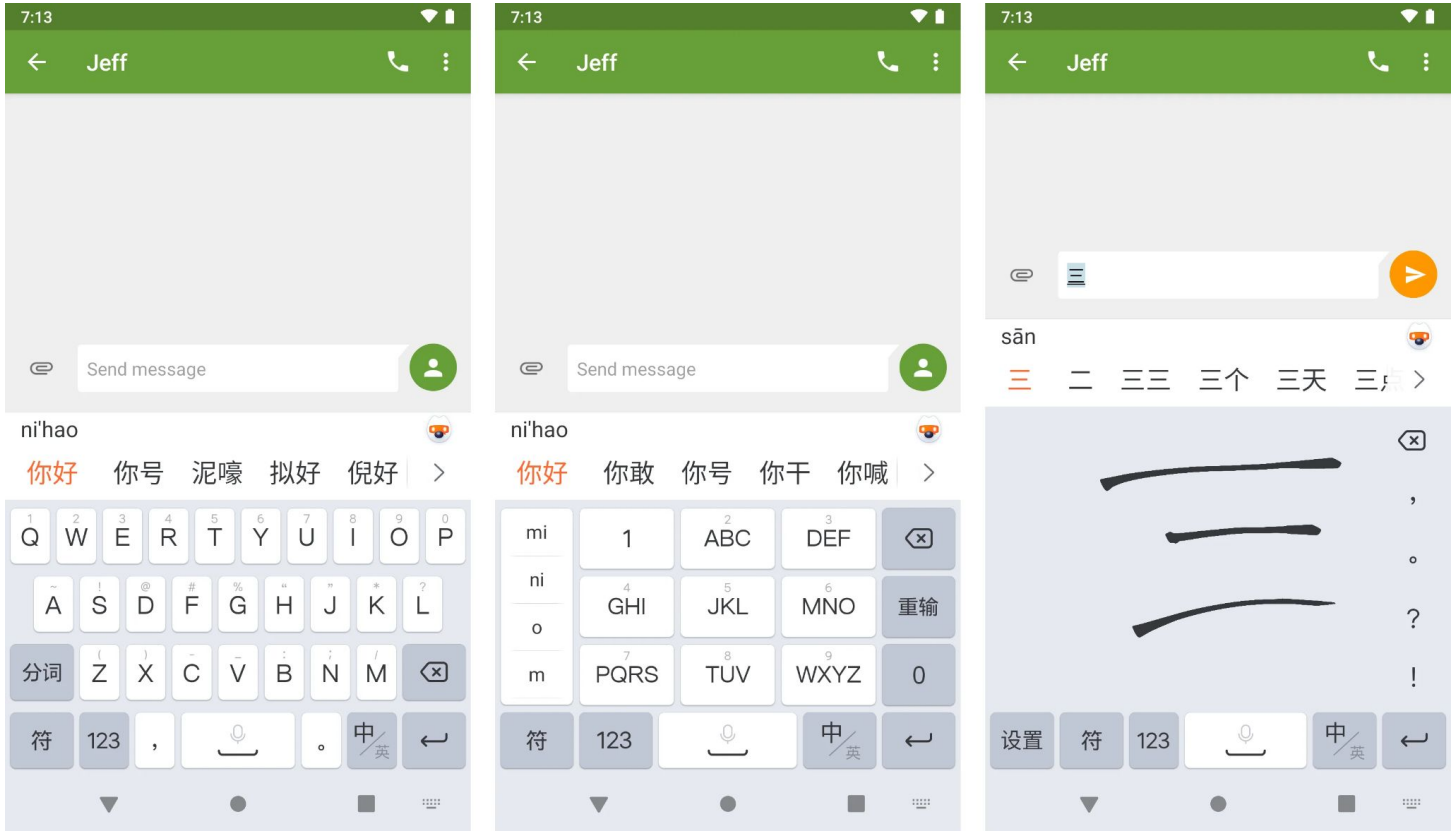
What’s a Chinese IME?



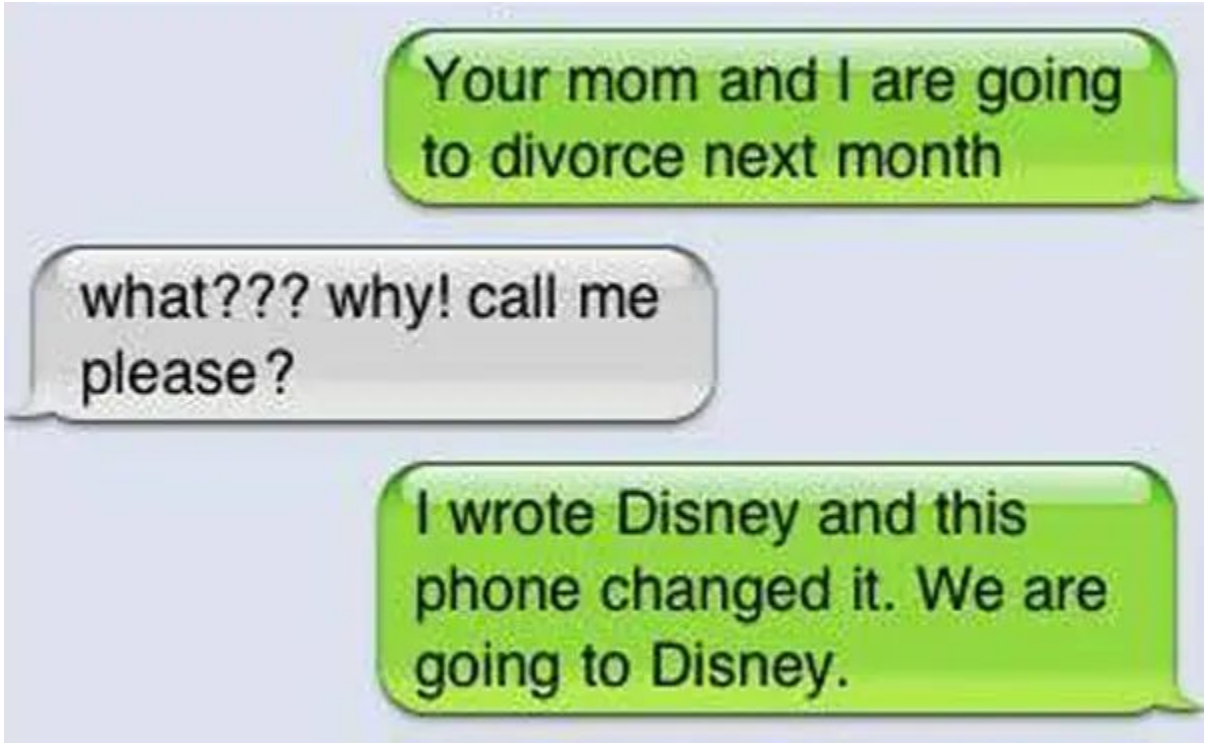
What’s a Chinese IME?



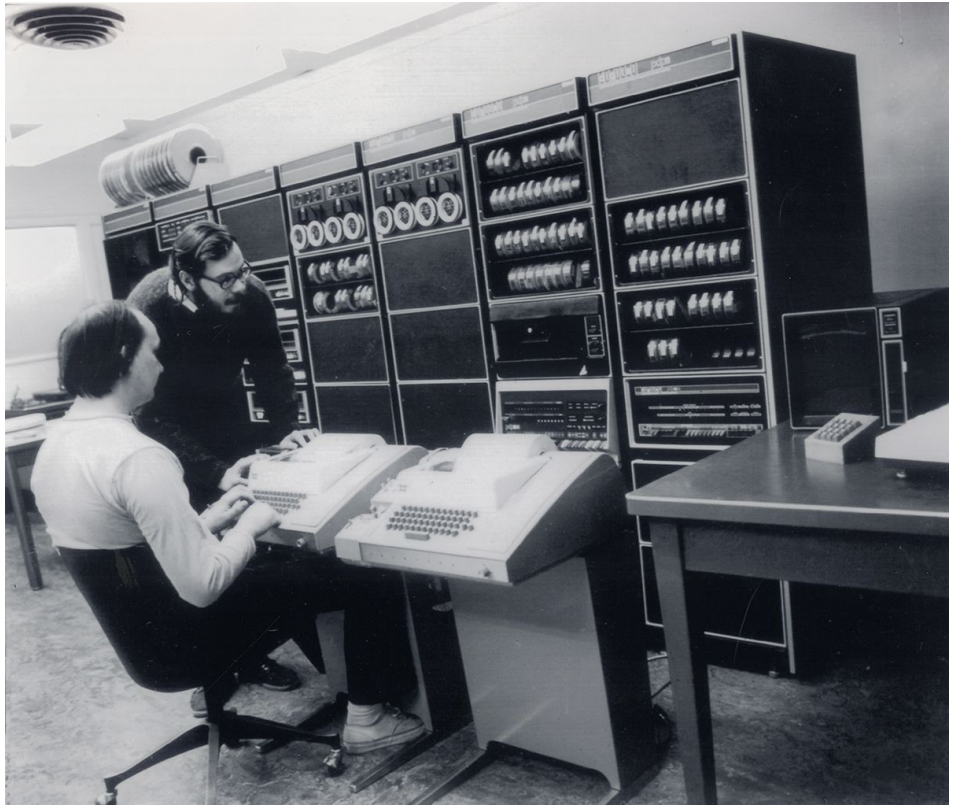
How does one type/input Chinese?



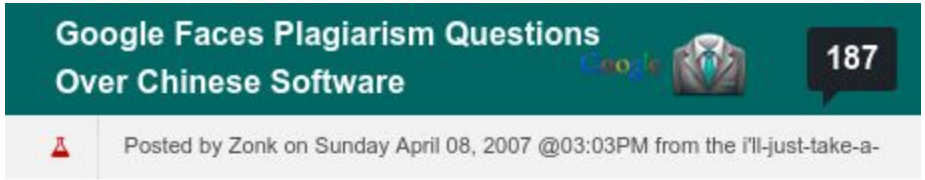
The advent of “cloud-based” prediction



The advent of “cloud-based” prediction



The advent of “cloud-based” prediction



Google Faces Plagiarism Questions Over Chinese Software

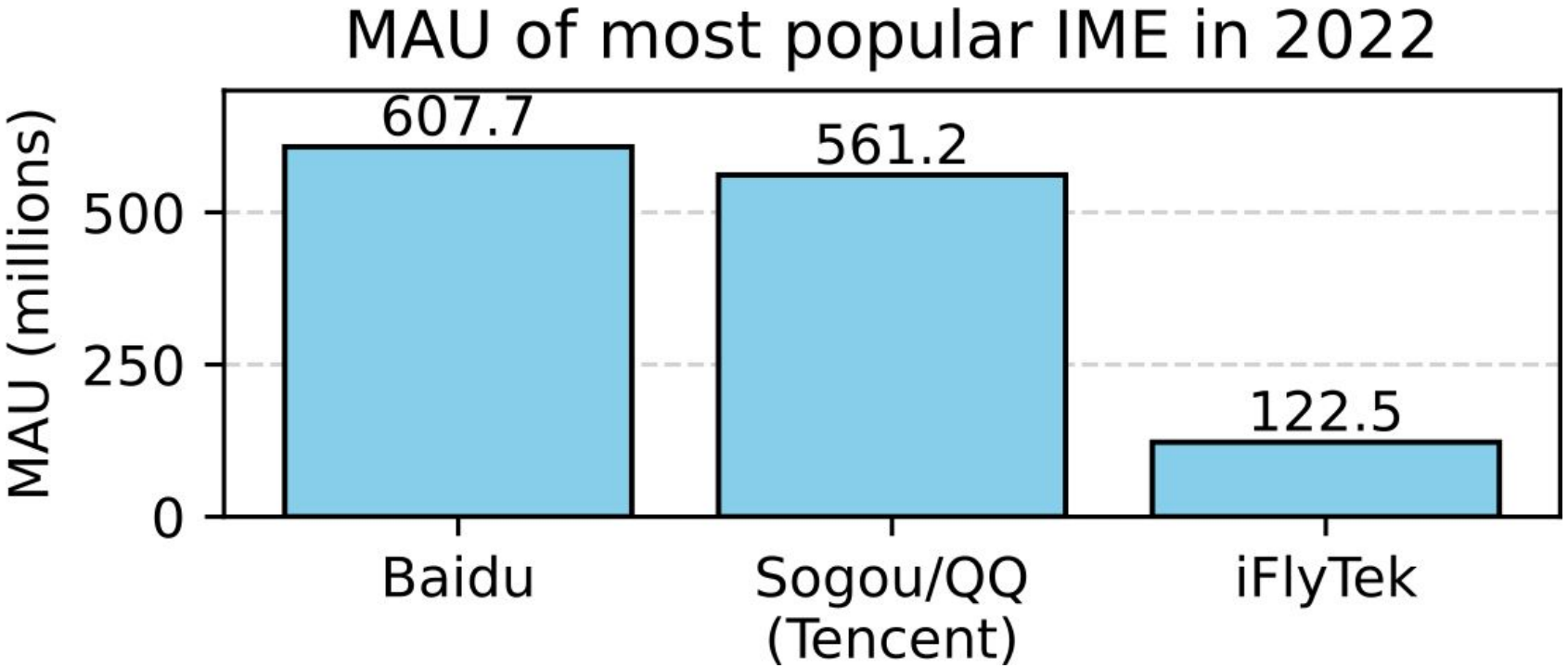
187

Posted by Zonk on Sunday April 08, 2007 @03:03PM from the i'll-just-take-a-

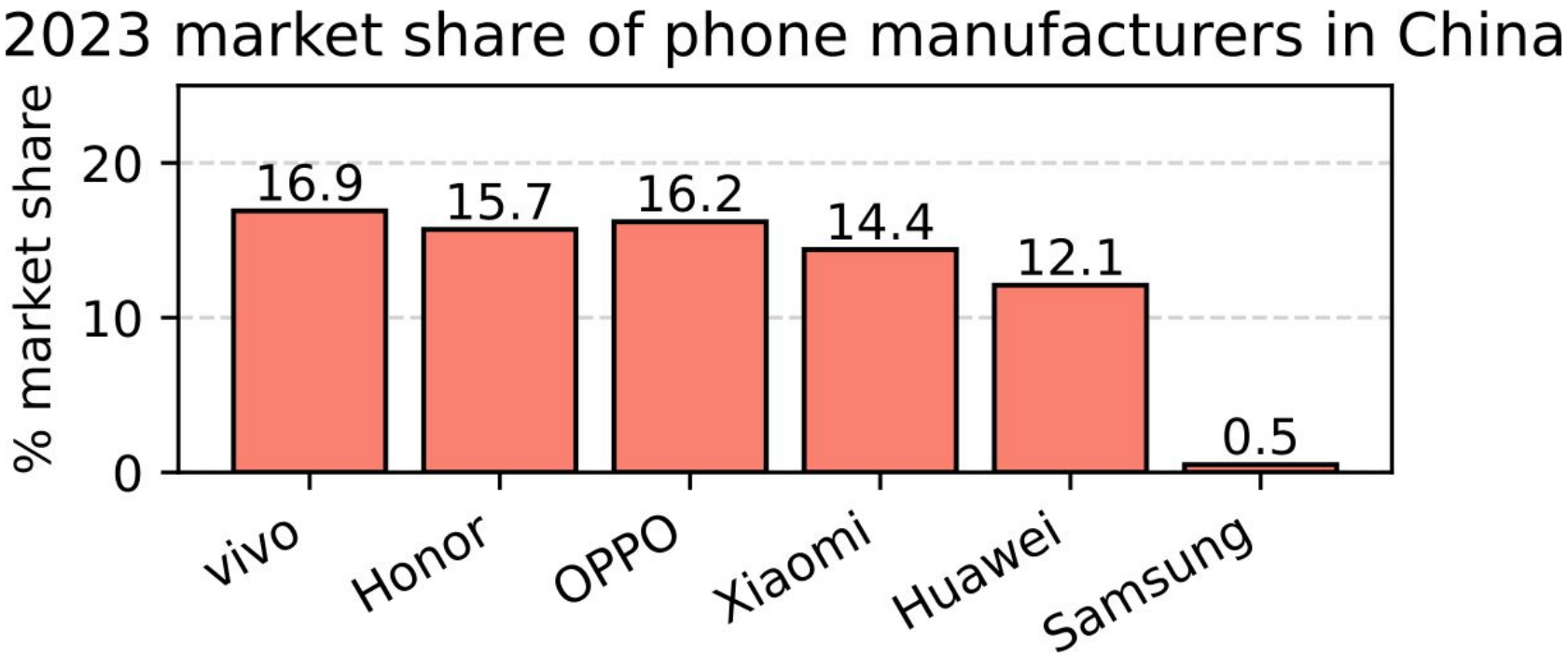
yaohua2000 writes

> "Google's laboratory in China has launched its first product, a Pinyin Input Method Editor. The software allows the romanized characters to be translated to more traditional Chinese symbols , via entering on a QWERTY keyboard. Users soon discovered that the data Google used for the product was unusually similar to the data used by a Chinese rival, Sogou. Google has evaded the question about software similarities, reports PC World. 'The similarities, which included an error involving the name of a celebrity, were noted on a Google Labs discussion board about its Pinyin IME. Users noted that entering the Pinyin pinggong into the Google IME incorrectly produced the name of Feng Gong, an actor and comedian.'"

Landscape of Chinese IMEs



Landscape of Chinese IMEs



They are keyloggers



This talk is **not** about how they are keyloggers



Threat model

- Attacks on Sogou: *Active* network adversary
- All other attacks?

All other attacks: Passive network adversary



Adversaries

- Anyone on your network
- Your ISP
- The server’s ISP
- Every network in between
- Every state actor in between



Where is X-KEYSCORE?



Approximately 150 sites

Over 700 servers

Success Stories

* UCWeb mobile browser identification
  * Discovered by GCHQ analyst during DSD workshop
  * Chinese mobile web browser – leaks IMSI, MSISDN, IMEI and device characteristics



UCWeb – XKS Microplugin



UCWeb

Help | Actions | Reports | View | Map View

	State	ID	Datetime	Highlights	Datetime End	Browser Version	Email Address	Handset Model	IMEI	IMSI	Global Title	Platform	Active User/T	Casenotation
1		1	2012-05-13 02:29:20		2012-05-13 02:29:23	8.0.3.107	[redacted]123movies	nokiae90-1	[redacted]		9379900100	java		E9DHL00000M0000
2		3	2012-05-13 06:00:59		2012-05-13 06:01:00	8.0.3.107	[redacted]123movies	nokiae90-1			9379900100	java		E9DHL00000M0000
3		4	2012-05-13 19:39:11		2012-05-13 19:39:11	7.9.3.103		HTC A510e				android		E9BDE00000M0000
4		2	2012-05-14 12:29:53		2012-05-14 12:29:53	8.0.4.121	[redacted]djgol	NokiaE72-1				sis		E9DHL00000M0000
5		5	2012-05-14 17:46:46		2012-05-14 17:46:46	8.0.4.121	[redacted]mobimasti	NokiaX6-00				sis		H5H125221450000
6		6	2012-05-15 18:28:19		2012-05-15 18:28:19	8.0.4.121	[redacted]mobimasti	NokiaX6-00			93781090013	sis		H5H125221450000
7		7	2012-05-15 20:02:5[redacted]		2012-05-15 20:02:5[redacted]	8.0.4.121	[redacted]mobimasti	NokiaX6-00			93781090013	sis		H5H1252214500[redacted]

Will VPNs save us?

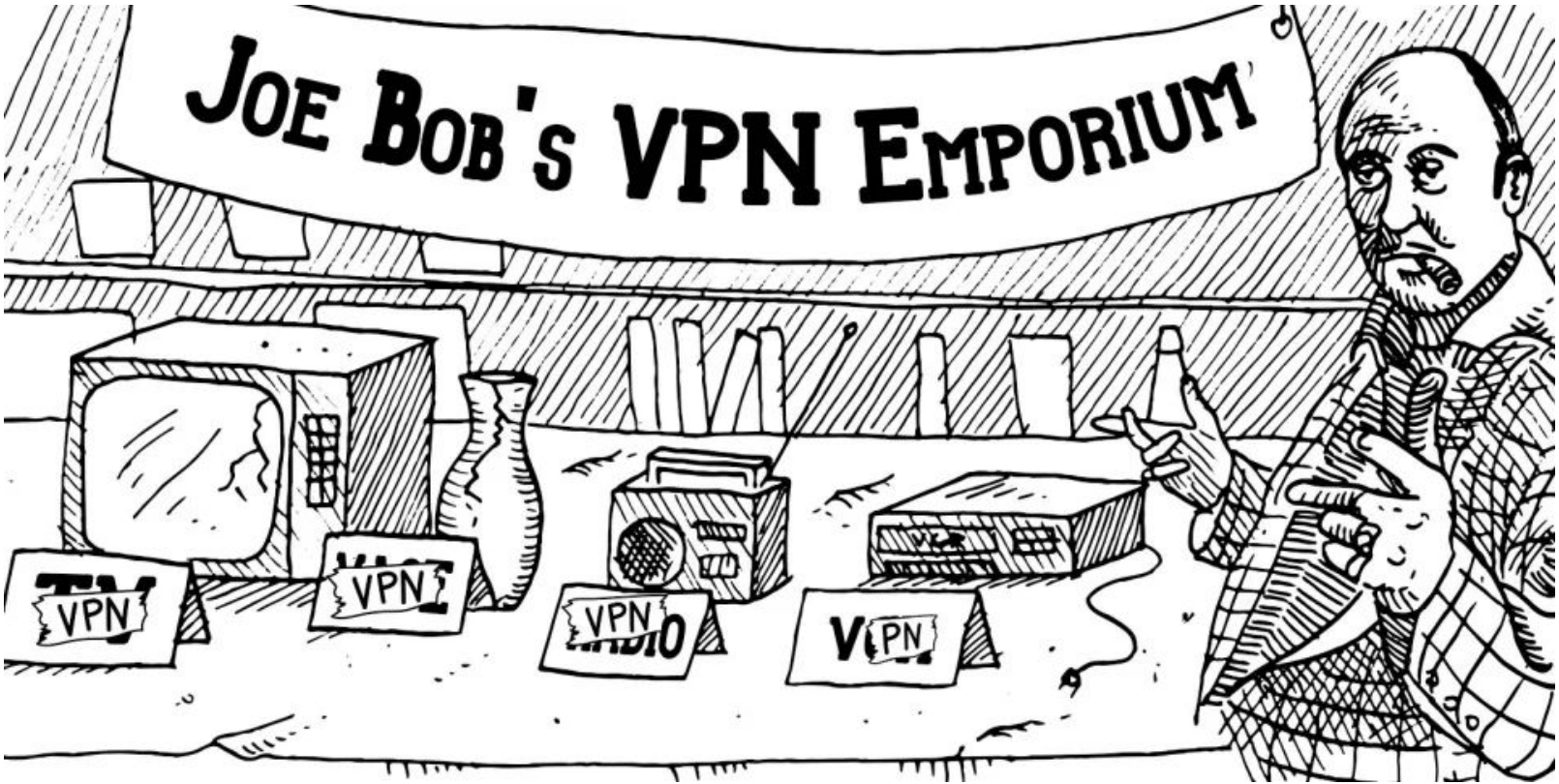


Illustration by D. Thomas Magee

We are **not** here to show off how clever our attacks are

- The problem is **how easy** our attacks are

- But… some of our attacks *are* cool :)

Legend	
✘✘	working exploit created to decrypt transmitted keystrokes for both **active and passive** eavesdroppers
✘	working exploit created to decrypt transmitted keystrokes for an **active** eavesdropper
!	weaknesses present in cryptography implementation
✔	no known issues
N/A	product not offered or not present on device analyzed

Keyboard developer	Android	iOS	Windows
Tencent	✘	✔	✘
Baidu	!	!	✘✘
iFlytek	✘✘	✔	✔

Legend	
✘✘	working exploit created to decrypt transmitted keystrokes for both **active and passive** eavesdroppers
✘	working exploit created to decrypt transmitted keystrokes for an **active** eavesdropper
!	weaknesses present in cryptography implementation
✔	no known issues
N/A	product not offered or not present on device analyzed

Device manufacturer	Own	Sogou	Baidu	iFlytek	iOS	Windows
Samsung	✘✘	✔*	✘✘	N/A	N/A	N/A
Huawei	✔*	✔	N/A	N/A	N/A	N/A
Xiaomi	N/A	✘*	✘✘	✘✘	N/A	N/A
OPPO	N/A	✘	✘✘*	N/A	N/A	N/A
Vivo	✔*	✘	N/A	N/A	N/A	N/A
Honor	N/A	N/A	✘✘*	N/A	N/A	N/A

* Default keyboard on device

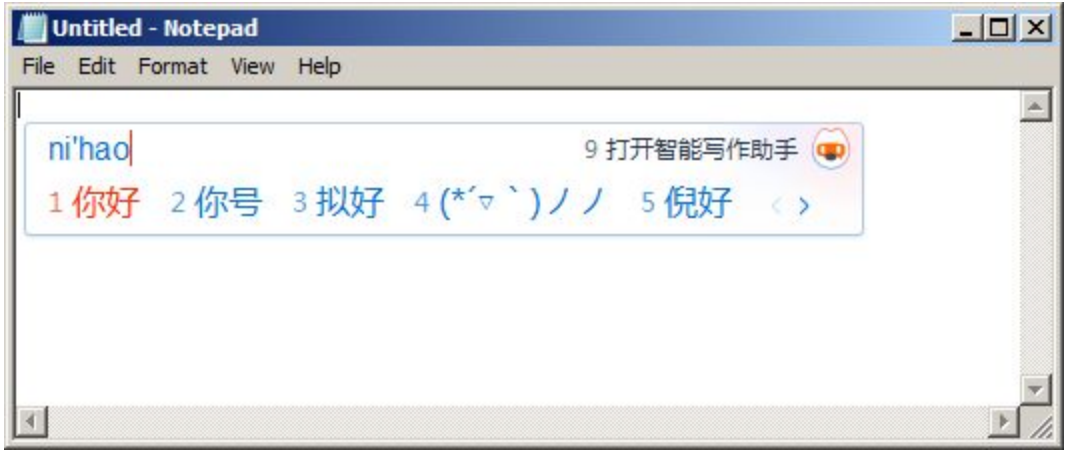
We considered an app secure if it used TLS

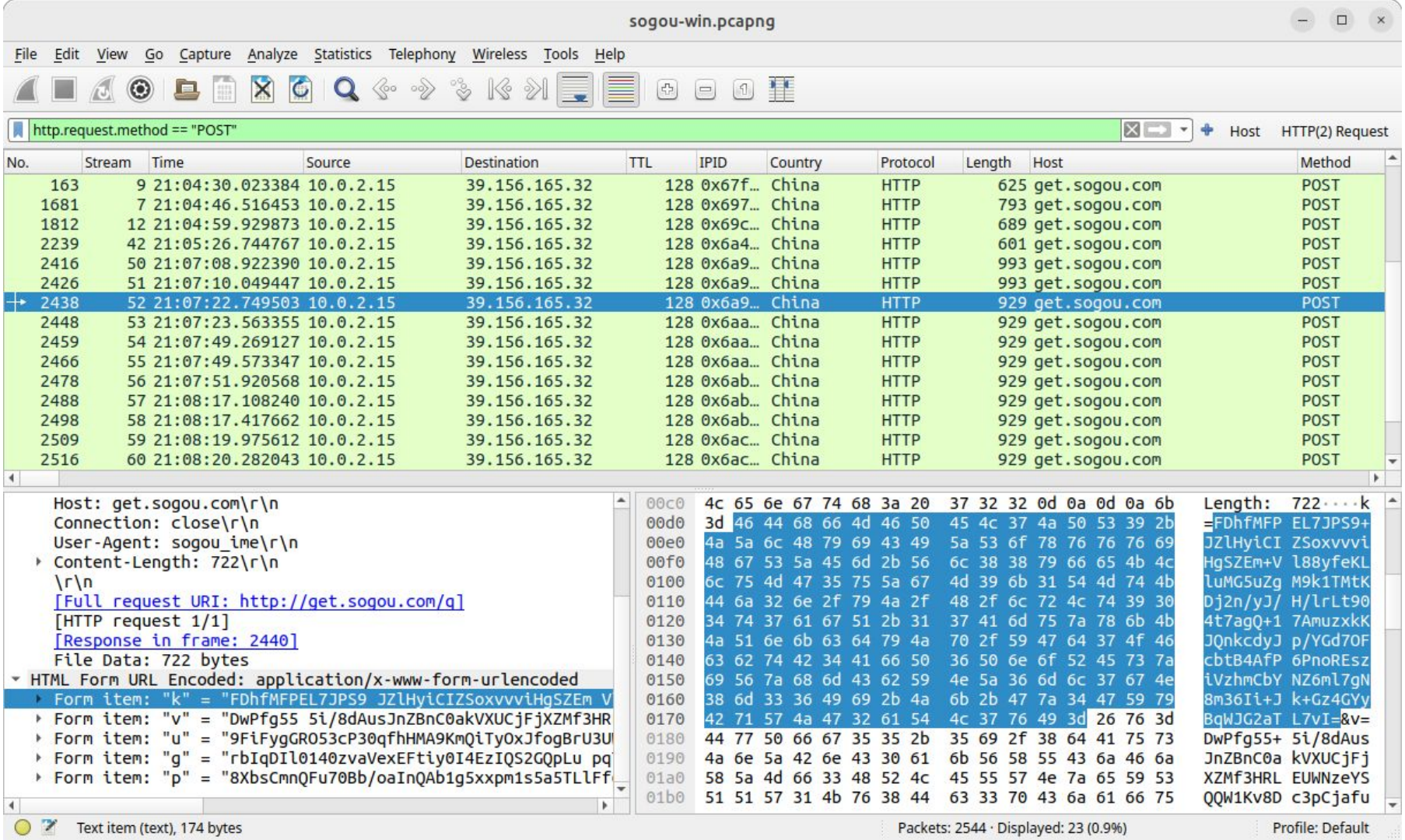


Keyboard Craptography Part 1: Sogou (The “Encrypt Wall”)

- Windows
- Android
- iOS

Sogou Windows




sogou-win.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http.request.method == "POST"
Host HTTP(2) Request
No. Stream Time Source Destination TTL IPID Country Protocol Length Host Method
163 9 21:04:30.023384 10.0.2.15 39.156.165.32 128 0x67f... China HTTP 625 get.sogou.com POST
1681 7 21:04:46.516453 10.0.2.15 39.156.165.32 128 0x697... China HTTP 793 get.sogou.com POST
1812 12 21:04:59.929873 10.0.2.15 39.156.165.32 128 0x69c... China HTTP 689 get.sogou.com POST
2239 42 21:05:26.744767 10.0.2.15 39.156.165.32 128 0x6a4... China HTTP 601 get.sogou.com POST
2416 50 21:07:08.922390 10.0.2.15 39.156.165.32 128 0x6a9... China HTTP 993 get.sogou.com POST
2426 51 21:07:10.049447 10.0.2.15 39.156.165.32 128 0x6a9... China HTTP 993 get.sogou.com POST
2438 52 21:07:22.749503 10.0.2.15 39.156.165.32 128 0x6a9... China HTTP 929 get.sogou.com POST
2448 53 21:07:23.563355 10.0.2.15 39.156.165.32 128 0x6aa... China HTTP 929 get.sogou.com POST
2459 54 21:07:49.269127 10.0.2.15 39.156.165.32 128 0x6aa... China HTTP 929 get.sogou.com POST
2466 55 21:07:49.573347 10.0.2.15 39.156.165.32 128 0x6aa... China HTTP 929 get.sogou.com POST
2478 56 21:07:51.920568 10.0.2.15 39.156.165.32 128 0x6ab... China HTTP 929 get.sogou.com POST
2488 57 21:08:17.108240 10.0.2.15 39.156.165.32 128 0x6ab... China HTTP 929 get.sogou.com POST
2498 58 21:08:17.417662 10.0.2.15 39.156.165.32 128 0x6ab... China HTTP 929 get.sogou.com POST
2509 59 21:08:19.975612 10.0.2.15 39.156.165.32 128 0x6ac... China HTTP 929 get.sogou.com POST
2516 60 21:08:20.282043 10.0.2.15 39.156.165.32 128 0x6ac... China HTTP 929 get.sogou.com POST
Host: get.sogou.com\r\n
Connection: close\r\n
User-Agent: sogou_ime\r\n
Content-Length: 722\r\n
\r\n
[Full request URI: http://get.sogou.com/q]
[HTTP request 1/1]
[Response in frame: 2440]
File Data: 722 bytes
HTML Form URL Encoded: application/x-www-form-urlencoded
Form item: "k" = "FDhfMFPEL7JPS9 JZlHyiCIZSoxvvviHgSZEm V
Form item: "v" = "DwPfg55 5i/8dAusJnZBnC0akVXUCjFjXZMf3HR
Form item: "u" = "9FiFygGRO53cP30qfhHMA9KmQiTyOxJfogBrU3U
Form item: "g" = "rbIqDIl0140zvaVexEFtiy0I4EzIQS2GQpLu pq
Form item: "p" = "8XbsCmnQFu70Bb/oaInQAb1g5xxpm1s5a5TLlFf
00c0 4c 65 6e 67 74 68 3a 20 37 32 32 0d 0a 0d 0a 6b Length: 722····k
00d0 3d 46 44 68 66 4d 46 50 45 4c 37 4a 50 53 39 2b =FDhfMFP EL7JPS9+
00e0 4a 5a 6c 48 79 69 43 49 5a 53 6f 78 76 76 76 69 JZlHyiCI ZSoxvvvi
00f0 48 67 53 5a 45 6d 2b 56 6c 38 38 79 66 65 4b 4c HgSZEm+V l88yfeKL
0100 6c 75 4d 47 35 75 5a 67 4d 39 6b 31 54 4d 74 4b luMG5uZg M9k1TMtK
0110 44 6a 32 6e 2f 79 4a 2f 48 2f 6c 72 4c 74 39 30 Dj2n/yJ/ H/lrLt90
0120 34 74 37 61 67 51 2b 31 37 41 6d 75 7a 78 6b 4b 4t7agQ+1 7AmuzxkK
0130 4a 51 6e 6b 63 64 79 4a 70 2f 59 47 64 37 4f 46 JQnkcdyJ p/YGd7OF
0140 63 62 74 42 34 41 66 50 36 50 6e 6f 52 45 73 7a cbtB4AfP 6PnoREsz
0150 69 56 7a 68 6d 43 62 59 4e 5a 36 6d 6c 37 67 4e iVzhmCbY NZ6ml7gN
0160 38 6d 33 36 49 69 2b 4a 6b 2b 47 7a 34 47 59 79 8m36Ii+J k+Gz4GYy
0170 42 71 57 4a 47 32 61 54 4c 37 76 49 3d 26 76 3d BqWJG2aT L7vI=&v=
0180 44 77 50 66 67 35 35 2b 35 69 2f 38 64 41 75 73 DwPfg55+ 5i/8dAus
0190 4a 6e 5a 42 6e 43 30 61 6b 56 58 55 43 6a 46 6a JnZBnC0a kVXUCjFj
01a0 58 5a 4d 66 33 48 52 4c 45 55 57 4e 7a 65 59 53 XZMf3HRL EUWNzeYS
01b0 51 51 57 31 4b 76 38 44 63 33 70 43 6a 61 66 75 QQW1Kv8D c3pCjafu
Text item (text), 174 bytes
Packets: 2544 · Displayed: 23 (0.9%)
Profile: Default

Sogou Craptography for Windows

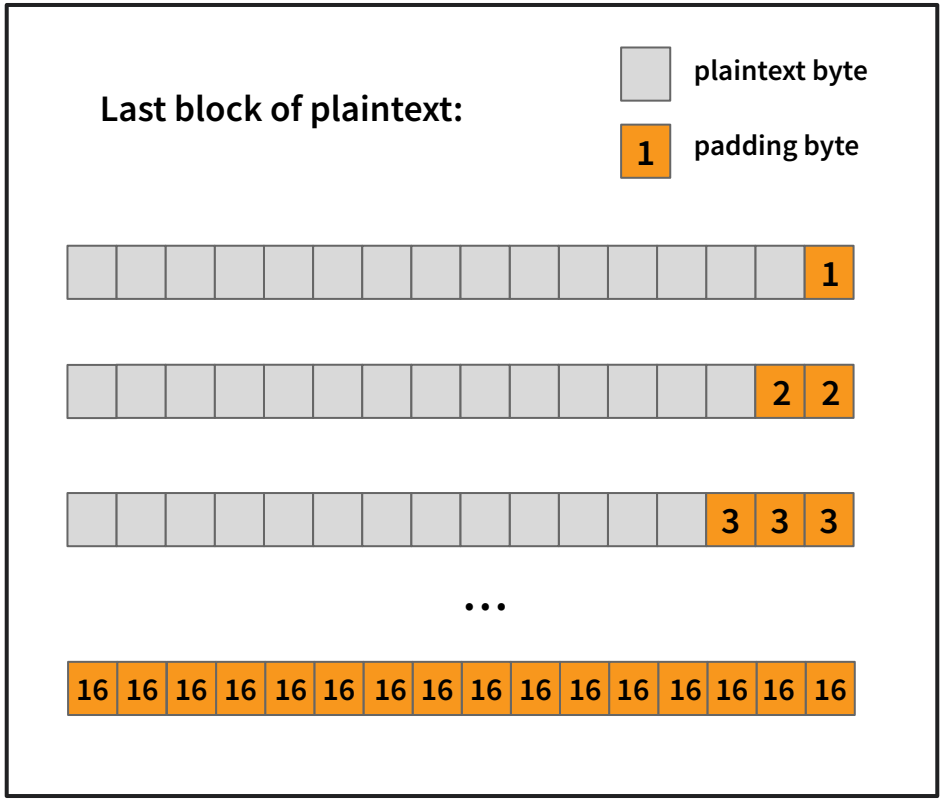
- K – AES key
- V – IV
- U – tunneled URL
- G – tunneled GET params
- P – tunneled POST params

Sogou Craptography for Windows

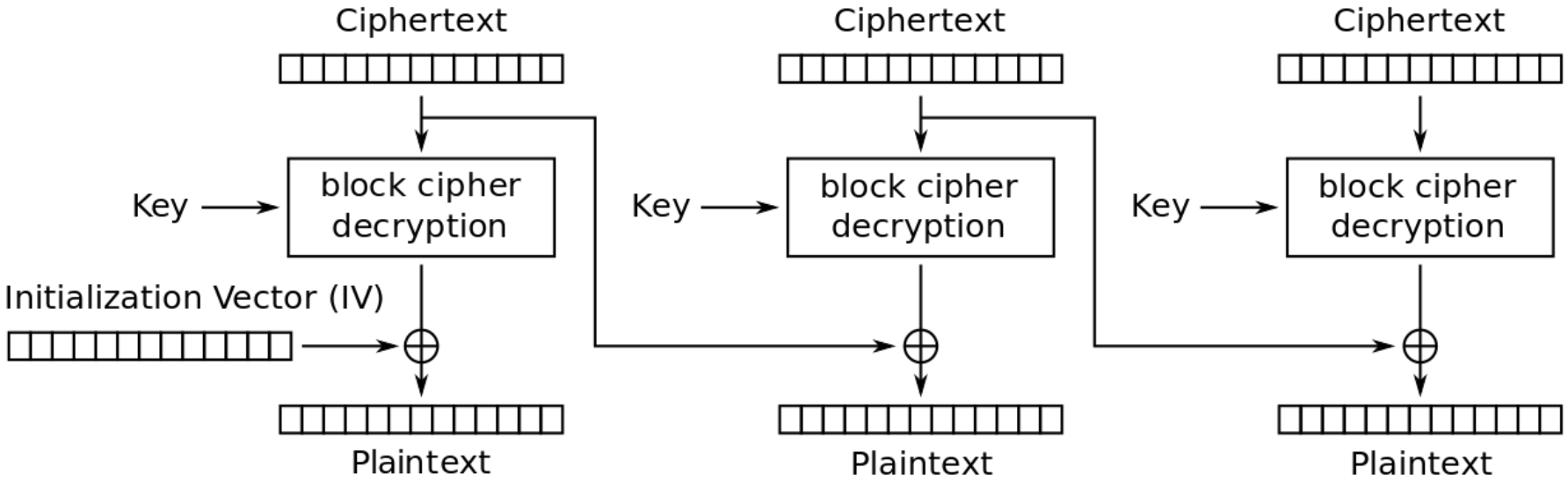
- K – AES key *k* (encrypted with 1024-bit public RSA key)
- V – IV *v* (encrypted with 1024-bit public RSA key)
- U – tunneled URL (encrypted with *k* and *v*)
- G – tunneled GET params (encrypted with *k* and *v*)
- P – tunneled POST params (encrypted with *k* and *v*)

CBC padding oracle attack

- Padding oracle?
- Distinct responses for good/bad padding

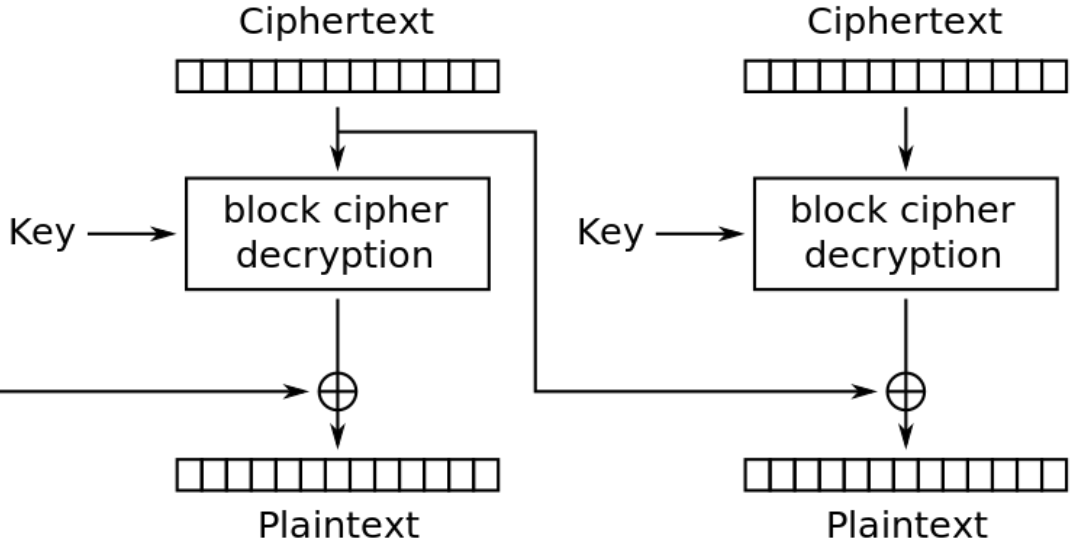


CBC padding oracle attack



Cipher Block Chaining (CBC) mode decryption

CBC padding oracle attack



$$\text{ciphertext}_{n-1} \oplus \text{Decrypt}(\text{ciphertext}_n) = \text{plaintext}_n$$

Chaining (CBC) mode decryption

XOR algebra

Since $X \oplus X = 0$…

1. $ciphertext_{n-1} \oplus Decrypt(ciphertext_n) = plaintext_n$
2. $ciphertext_{n-1} \oplus Decrypt(ciphertext_n) \oplus plaintext_n = 0$
3. $Decrypt(ciphertext_n) = ciphertext_{n-1} \oplus plaintext_n$

$$ciphertext_{n-1} \oplus Decrypt(ciphertext_n) = plaintext_n$$

Find byte *b* such that

$$b \oplus Decrypt(ciphertext_n)[15] == 0x01 \Rightarrow Decrypt(ciphertext_n)[15] == b \oplus 0x01$$

To recover $plaintext_n[15]$

$$plaintext_n[15] = ciphertext_{n-1}[15] \oplus b \oplus 0x01$$

Set

$$ciphertext_{n-1}[15] = b \oplus 0x01 \oplus 0x02$$

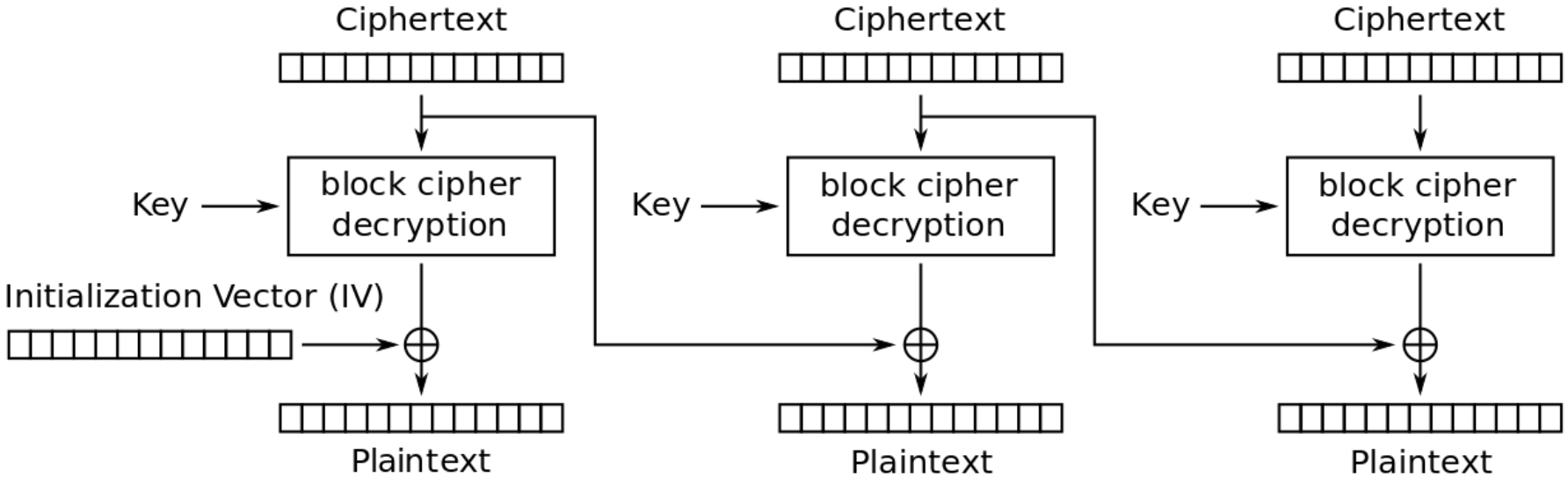
Find *b* such that

$$b \oplus Decrypt(ciphertext_n)[14] == 0x02$$

…

CBC padding oracle attack

- Problem 1: IV encrypted with RSA key
- Problem 2: plaintext is zlib-compressed



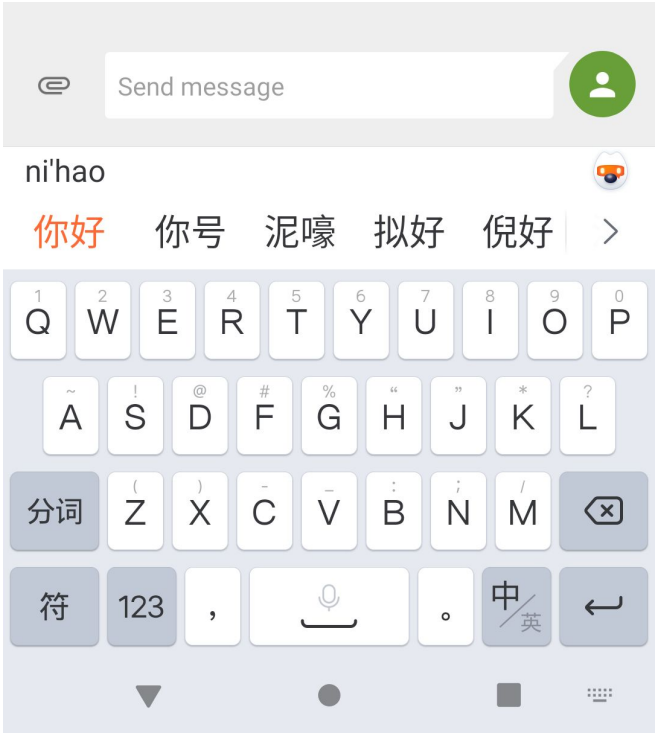
Cipher Block Chaining (CBC) mode decryption

CBC padding oracle attack

- The beginning of the URL field “U” is predictable during typing
- Same key, IV used for U, G, and P
- Set IV = all zeros
- Attack first block per normal, yielding $\text{plaintext}_1 \oplus \text{IV}$
- We know plaintext_1 (!)
- We can recover the IV

```
1 {
  1: 1
  2 {
    1 {
      2: "1111_sogou_pinyin_guanwang_13.4e_1111"
      3: "13.4.0.7561"
      5: 3
      7: 1
      8: "13.4.0.7561"
    }
    7: "nihaohaohaohaohaohaohaozdaasdfffaahellocanyoureadthis"
    16: 11
    17 {
      3 {
        1: 2
        2: 1
      }
      9: 1
      10: 1
    }
    19 {
      4: "0"
    }
[...]
```

Sogou Android



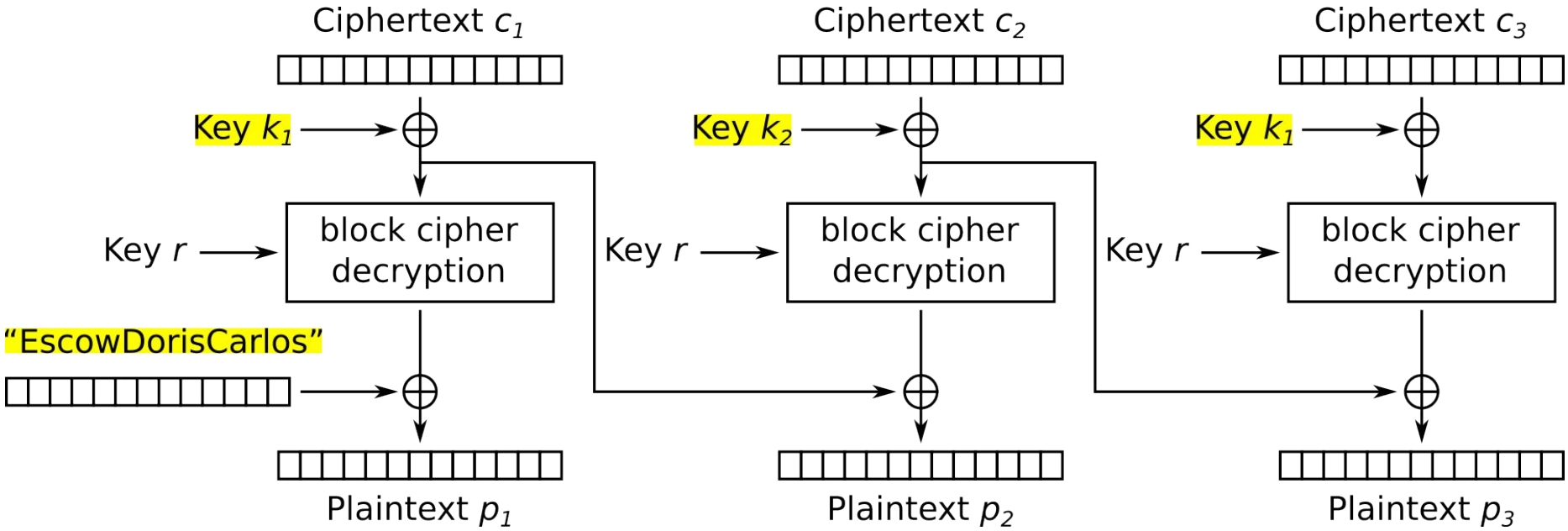
Sogou Craptography for Android

- K – AES key *k* (encrypted with 1024-bit public RSA key)
- V – IV *v* (not encrypted this time)
- U – tunneled URL (encrypted with *k* and *v*)
- G – tunneled GET params (encrypted with *k* and *v*)
- P – tunneled POST params (encrypted with *k* and *v*)

Sogou Craptography for Android

- K – AES key *k* (encrypted with 1024-bit public RSA key)
- V – IV *v* (not encrypted this time)
- U – tunneled URL (encrypted with *k* and *v*)
- G – tunneled GET params (encrypted with *k* and *v*)
- P – tunneled POST params (encrypted with *k* and *v*)
- R – *k* ⊕ another AES key *r*
- S – *k* ⊕ (*s* encrypted with *r* and “EscowDorisCarlos”)
- E – *k* ⊕ (*e* encrypted with *r* and “EscowDorisCarlos”)
- F – *k* ⊕ (*f* encrypted with *r* and “EscowDorisCarlos”)

Sogou Craptography for Android



Sogou Craptography for Android

- Let 256-bit k = 128-bit k_1 || 128-bit k_2
- After attack, first plaintext blocks of S, E, F:
- $k_2 \oplus s$
- $k_2 \oplus e$
- $k_2 \oplus f$
- s is predictable
- We can recover k_2
- We can recover e, f, and r_2

```
1 {
  1: "com.android.messaging"
  2: "11.20"
  4: 1
  6: "android_sweb"
  8: "Google"
  10: "android_sweb"
  11: "11.20"
  14: "30"
  18: "-1"
  22: "5682b3aa4fa7bd40d776c93a35a77c6d"
}
2 {
  1: 0xbff0000000000000
  2: 0xbff0000000000000
  3: "-1"
}
3: 1
4: "canyoureadthis"
11 {
  1: "onekeyimageenable"
  2: "1"
}
```

Sogou Craptography for iOS?

- Key generation?
- Key from Unix time in seconds
- If you know the time, you know the key
- IV from Unix time in seconds
- Key likely to be IV
- The IV is transmitted unencrypted!

```
void __cdecl +[DataEncryptor randomizeAesKeyIv:keyLen:iv:ivLen:](
        id a1,
        SEL a2,
        unsigned __int8 *key,
        ssize_t key_len,
        unsigned __int8 *iv,
        ssize_t iv_len)
{
  unsigned __int8 *iv_; // x20
  unsigned __int8 *key_; // x22
  unsigned int key_seed; // w0
  unsigned int iv_seed; // w0

  if ( key )
  {
    iv_ = iv;
    if ( iv )
    {
      key_ = key;
      key_seed = time(0LL);
      srand(key_seed);
      if ( key_len >= 1 )
      {
        do
        {
          *key_++ = rand();
          --key_len;
        }
        while ( key_len );
      }
      iv_seed = time(0LL);
      srand(iv_seed);
      if ( iv_len >= 1 )
      {
        do
        {
          *iv_++ = rand();
          --iv_len;
        }
        while ( iv_len );
      }
    }
  }
}
```

Sogou Craptography for iOS?

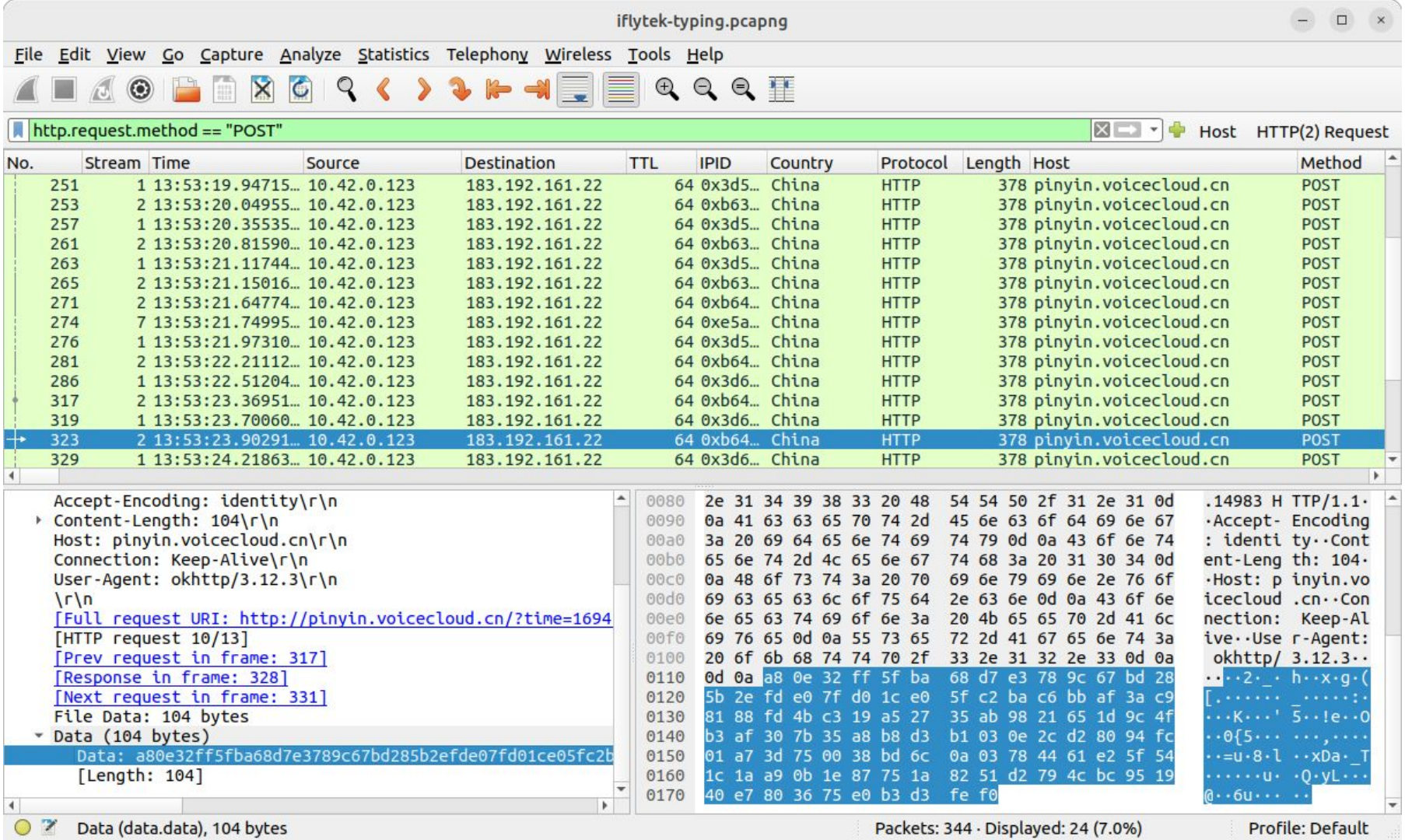
- BUT, all of this is wrapped in TLS!

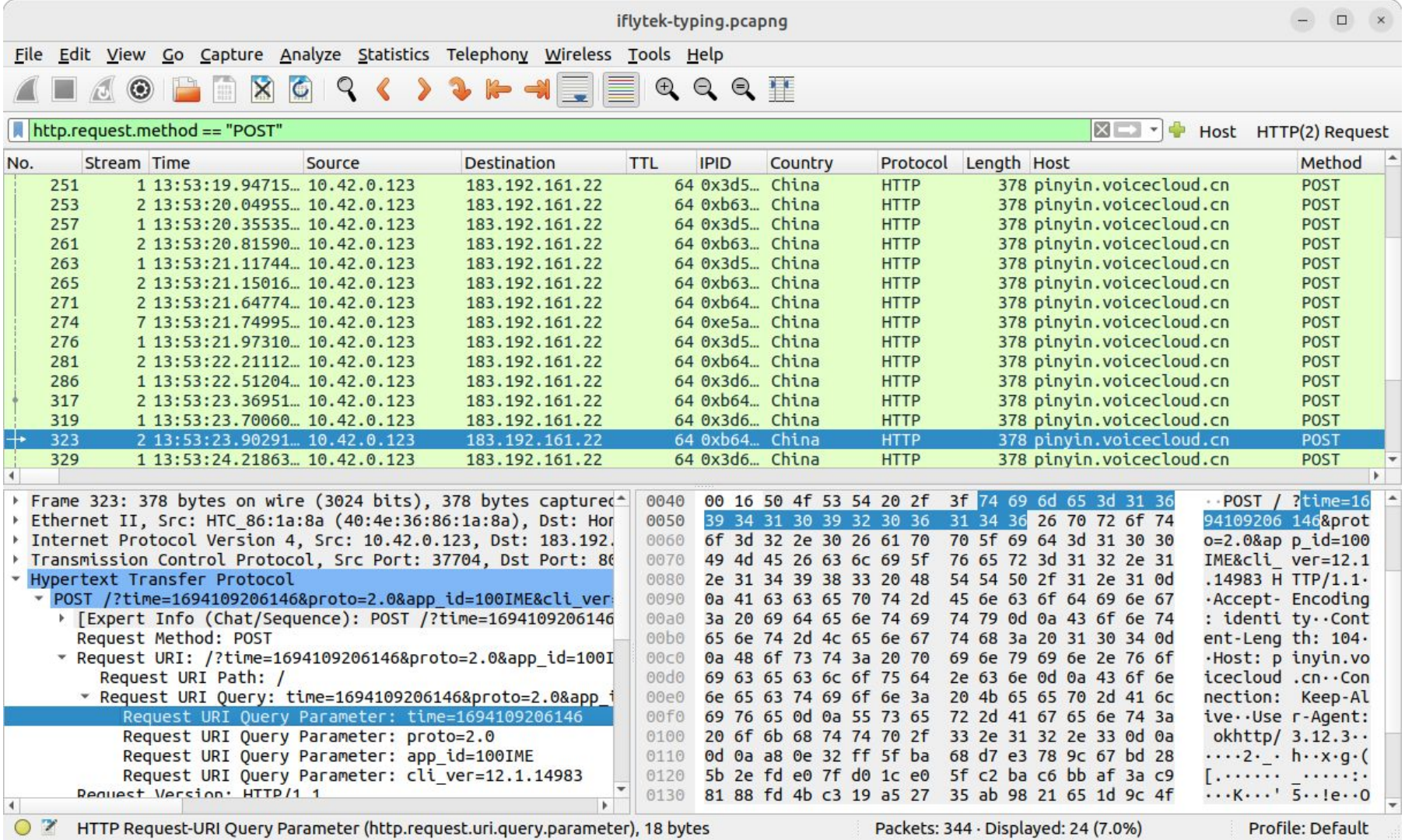
```
void __cdecl +[DataEncryptor randomizeAesKeyIv:keyLen:iv:ivLen:](
        id a1,
        SEL a2,
        unsigned __int8 *key,
        ssize_t key_len,
        unsigned __int8 *iv,
        ssize_t iv_len)
{
  unsigned __int8 *iv_; // x20
  unsigned __int8 *key_; // x22
  unsigned int key_seed; // w0
  unsigned int iv_seed; // w0

  if ( key )
  {
    iv_ = iv;
    if ( iv )
    {
      key_ = key;
      key_seed = time(0LL);
      srand(key_seed);
      if ( key_len >= 1 )
      {
        do
        {
          *key_++ = rand();
          --key_len;
        }
        while ( key_len );
      }
      iv_seed = time(0LL);
      srand(iv_seed);
      if ( iv_len >= 1 )
      {
        do
        {
          *iv_++ = rand();
          --iv_len;
        }
        while ( iv_len );
      }
    }
  }
}
```

Keyboard Craptography Part 2: iFlytek

- Android
- iOS
- Windows


iflytek-typing.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http.request.method == "POST"
Host HTTP(2) Request
No. Stream Time Source Destination TTL IPID Country Protocol Length Host Method
251 1 13:53:19.94715… 10.42.0.123 183.192.161.22 64 0x3d5… China HTTP 378 pinyin.voicecloud.cn POST
253 2 13:53:20.04955… 10.42.0.123 183.192.161.22 64 0xb63… China HTTP 378 pinyin.voicecloud.cn POST
257 1 13:53:20.35535… 10.42.0.123 183.192.161.22 64 0x3d5… China HTTP 378 pinyin.voicecloud.cn POST
261 2 13:53:20.81590… 10.42.0.123 183.192.161.22 64 0xb63… China HTTP 378 pinyin.voicecloud.cn POST
263 1 13:53:21.11744… 10.42.0.123 183.192.161.22 64 0x3d5… China HTTP 378 pinyin.voicecloud.cn POST
265 2 13:53:21.15016… 10.42.0.123 183.192.161.22 64 0xb63… China HTTP 378 pinyin.voicecloud.cn POST
271 2 13:53:21.64774… 10.42.0.123 183.192.161.22 64 0xb64… China HTTP 378 pinyin.voicecloud.cn POST
274 7 13:53:21.74995… 10.42.0.123 183.192.161.22 64 0xe5a… China HTTP 378 pinyin.voicecloud.cn POST
276 1 13:53:21.97310… 10.42.0.123 183.192.161.22 64 0x3d5… China HTTP 378 pinyin.voicecloud.cn POST
281 2 13:53:22.21112… 10.42.0.123 183.192.161.22 64 0xb64… China HTTP 378 pinyin.voicecloud.cn POST
286 1 13:53:22.51204… 10.42.0.123 183.192.161.22 64 0x3d6… China HTTP 378 pinyin.voicecloud.cn POST
317 2 13:53:23.36951… 10.42.0.123 183.192.161.22 64 0xb64… China HTTP 378 pinyin.voicecloud.cn POST
319 1 13:53:23.70060… 10.42.0.123 183.192.161.22 64 0x3d6… China HTTP 378 pinyin.voicecloud.cn POST
323 2 13:53:23.90291… 10.42.0.123 183.192.161.22 64 0xb64… China HTTP 378 pinyin.voicecloud.cn POST
329 1 13:53:24.21863… 10.42.0.123 183.192.161.22 64 0x3d6… China HTTP 378 pinyin.voicecloud.cn POST
Accept-Encoding: identity\r\n
Content-Length: 104\r\n
Host: pinyin.voicecloud.cn\r\n
Connection: Keep-Alive\r\n
User-Agent: okhttp/3.12.3\r\n
\r\n
[Full request URI: http://pinyin.voicecloud.cn/?time=1694
[HTTP request 10/13]
[Prev request in frame: 317]
[Response in frame: 328]
[Next request in frame: 331]
File Data: 104 bytes
Data (104 bytes)
Data: a80e32ff5fba68d7e3789c67bd285b2efde07fd01ce05fc2b
[Length: 104]
0080 2e 31 34 39 38 33 20 48 54 54 50 2f 31 2e 31 0d .14983 H TTP/1.1·
0090 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 ·Accept- Encoding
00a0 3a 20 69 64 65 6e 74 69 74 79 0d 0a 43 6f 6e 74 : identi ty··Cont
00b0 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 31 30 34 0d ent-Leng th: 104·
00c0 0a 48 6f 73 74 3a 20 70 69 6e 79 69 6e 2e 76 6f ·Host: p inyin.vo
00d0 69 63 65 63 6c 6f 75 64 2e 63 6e 0d 0a 43 6f 6e icecloud .cn··Con
00e0 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c nection: Keep-Al
00f0 69 76 65 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a ive··Use r-Agent:
0100 20 6f 6b 68 74 74 70 2f 33 2e 31 32 2e 33 0d 0a okhttp/ 3.12.3··
0110 0d 0a a8 0e 32 ff 5f ba 68 d7 e3 78 9c 67 bd 28 ··2·_· h··x·g·(
0120 5b 2e fd e0 7f d0 1c e0 5f c2 ba c6 bb af 3a c9 [·····_ ·····:·
0130 81 88 fd 4b c3 19 a5 27 35 ab 98 21 65 1d 9c 4f ···K···' 5··!e··O
0140 b3 af 30 7b 35 a8 b8 d3 b1 03 0e 2c d2 80 94 fc ··0{5··· ···,····
0150 01 a7 3d 75 00 38 bd 6c 0a 03 78 44 61 e2 5f 54 ··=u·8·l ··xDa·_T
0160 1c 1a a9 0b 1e 87 75 1a 82 51 d2 79 4c bc 95 19 ·····u· ·Q·yL···
0170 40 e7 80 36 75 e0 b3 d3 fe f0 @··6u··· ··
Data (data.data), 104 bytes
Packets: 344 · Displayed: 24 (7.0%)
Profile: Default


iflytek-typing.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http.request.method == "POST"
Host HTTP(2) Request
No. Stream Time Source Destination TTL IPID Country Protocol Length Host Method
251 1 13:53:19.94715… 10.42.0.123 183.192.161.22 64 0x3d5… China HTTP 378 pinyin.voicecloud.cn POST
253 2 13:53:20.04955… 10.42.0.123 183.192.161.22 64 0xb63… China HTTP 378 pinyin.voicecloud.cn POST
257 1 13:53:20.35535… 10.42.0.123 183.192.161.22 64 0x3d5… China HTTP 378 pinyin.voicecloud.cn POST
261 2 13:53:20.81590… 10.42.0.123 183.192.161.22 64 0xb63… China HTTP 378 pinyin.voicecloud.cn POST
263 1 13:53:21.11744… 10.42.0.123 183.192.161.22 64 0x3d5… China HTTP 378 pinyin.voicecloud.cn POST
265 2 13:53:21.15016… 10.42.0.123 183.192.161.22 64 0xb63… China HTTP 378 pinyin.voicecloud.cn POST
271 2 13:53:21.64774… 10.42.0.123 183.192.161.22 64 0xb64… China HTTP 378 pinyin.voicecloud.cn POST
274 7 13:53:21.74995… 10.42.0.123 183.192.161.22 64 0xe5a… China HTTP 378 pinyin.voicecloud.cn POST
276 1 13:53:21.97310… 10.42.0.123 183.192.161.22 64 0x3d5… China HTTP 378 pinyin.voicecloud.cn POST
281 2 13:53:22.21112… 10.42.0.123 183.192.161.22 64 0xb64… China HTTP 378 pinyin.voicecloud.cn POST
286 1 13:53:22.51204… 10.42.0.123 183.192.161.22 64 0x3d6… China HTTP 378 pinyin.voicecloud.cn POST
317 2 13:53:23.36951… 10.42.0.123 183.192.161.22 64 0xb64… China HTTP 378 pinyin.voicecloud.cn POST
319 1 13:53:23.70060… 10.42.0.123 183.192.161.22 64 0x3d6… China HTTP 378 pinyin.voicecloud.cn POST
323 2 13:53:23.90291… 10.42.0.123 183.192.161.22 64 0xb64… China HTTP 378 pinyin.voicecloud.cn POST
329 1 13:53:24.21863… 10.42.0.123 183.192.161.22 64 0x3d6… China HTTP 378 pinyin.voicecloud.cn POST
Frame 323: 378 bytes on wire (3024 bits), 378 bytes captured
Ethernet II, Src: HTC_86:1a:8a (40:4e:36:86:1a:8a), Dst: Hon
Internet Protocol Version 4, Src: 10.42.0.123, Dst: 183.192.
Transmission Control Protocol, Src Port: 37704, Dst Port: 80
Hypertext Transfer Protocol
POST /?time=1694109206146&proto=2.0&app_id=100IME&cli_ver
[Expert Info (Chat/Sequence): POST /?time=1694109206146
Request Method: POST
Request URI: /?time=1694109206146&proto=2.0&app_id=100I
Request URI Path: /
Request URI Query: time=1694109206146&proto=2.0&app_i
Request URI Query Parameter: time=1694109206146
Request URI Query Parameter: proto=2.0
Request URI Query Parameter: app_id=100IME
Request URI Query Parameter: cli_ver=12.1.14983
Request Version: HTTP/1.1
0040 00 16 50 4f 53 54 20 2f 3f 74 69 6d 65 3d 31 36 ··POST / ?time=16
0050 39 34 31 30 39 32 30 36 31 34 36 26 70 72 6f 74 94109206 146&prot
0060 6f 3d 32 2e 30 26 61 70 70 5f 69 64 3d 31 30 30 o=2.0&ap p_id=100
0070 49 4d 45 26 63 6c 69 5f 76 65 72 3d 31 32 2e 31 IME&cli_ ver=12.1
0080 2e 31 34 39 38 33 20 48 54 54 50 2f 31 2e 31 0d .14983 H TTP/1.1·
0090 0a 41 63 63 65 70 74 2d 45 6e 63 6f 64 69 6e 67 ·Accept- Encoding
00a0 3a 20 69 64 65 6e 74 69 74 79 0d 0a 43 6f 6e 74 : identi ty··Cont
00b0 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 31 30 34 0d ent-Leng th: 104·
00c0 0a 48 6f 73 74 3a 20 70 69 6e 79 69 6e 2e 76 6f ·Host: p inyin.vo
00d0 69 63 65 63 6c 6f 75 64 2e 63 6e 0d 0a 43 6f 6e icecloud .cn··Con
00e0 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c nection: Keep-Al
00f0 69 76 65 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a ive··Use r-Agent:
0100 20 6f 6b 68 74 74 70 2f 33 2e 31 32 2e 33 0d 0a okhttp/ 3.12.3··
0110 0d 0a a8 0e 32 ff 5f ba 68 d7 e3 78 9c 67 bd 28 ····2·_· h··x·g·(
0120 5b 2e fd e0 7f d0 1c e0 5f c2 ba c6 bb af 3a c9 [····· _·····:·
0130 81 88 fd 4b c3 19 a5 27 35 ab 98 21 65 1d 9c 4f ···K···' 5··!e··O
HTTP Request-URI Query Parameter (http.request.uri.query.parameter), 18 bytes
Packets: 344 · Displayed: 24 (7.0%)
Profile: Default

iFlytek craptography for Android

- Encrypted using DES key *k* in ECB mode.

iFlytek craptography for Android

- Encrypted using DES key *k* in ECB mode.
- How is *k* derived?

```
__int64 generate_key_seed()
{
  return now_nanoseconds() / 1000000;
}
```

[...]

```
  if ( ptr && length )
  {
    v17 = (_DWORD)length + 8;
    output = malloc((int)length + 8);
    memset(output, 0, v17);
    sprintf((char *)key, "%08llu", (key_seed % 0x5F5E100) ^ 0x1001111);
    v19 = DES_ECB_Encrypt(ptr, (__int64)output, key, (unsigned int)length);
    free(ptr);
```

iFlytek craptography for Android

- Encrypted using DES key *k* in ECB mode.
- How is *k* derived?

$$k = \text{b'\%08u'} \% ((s \% 0x5F5E100) \wedge 0x1001111)$$

where *s* is the current Unix time in milliseconds.

iFlytek craptography for Android

- Who can decrypt everything you type?
- Anyone who knows the time
- Just in case you don’t know the time, they tell you the time for you…

```
1: 0
2: 0
3: 49
4: "xxxxx"
5: 0
7 {
    1: "app_id"
    2: "100IME"
}
7 {
    1: "uid"
    2: "2308170317523964l8"
}
7 {
    1: "cli_ver"
    2: "12.1.14983"
}
7 {
    1: "net_type"
    2: "wifi"
}
7 {
    1: "OS"
    2: "android"
}
8: 8
```

iFlytek craptography for iOS and Windows?

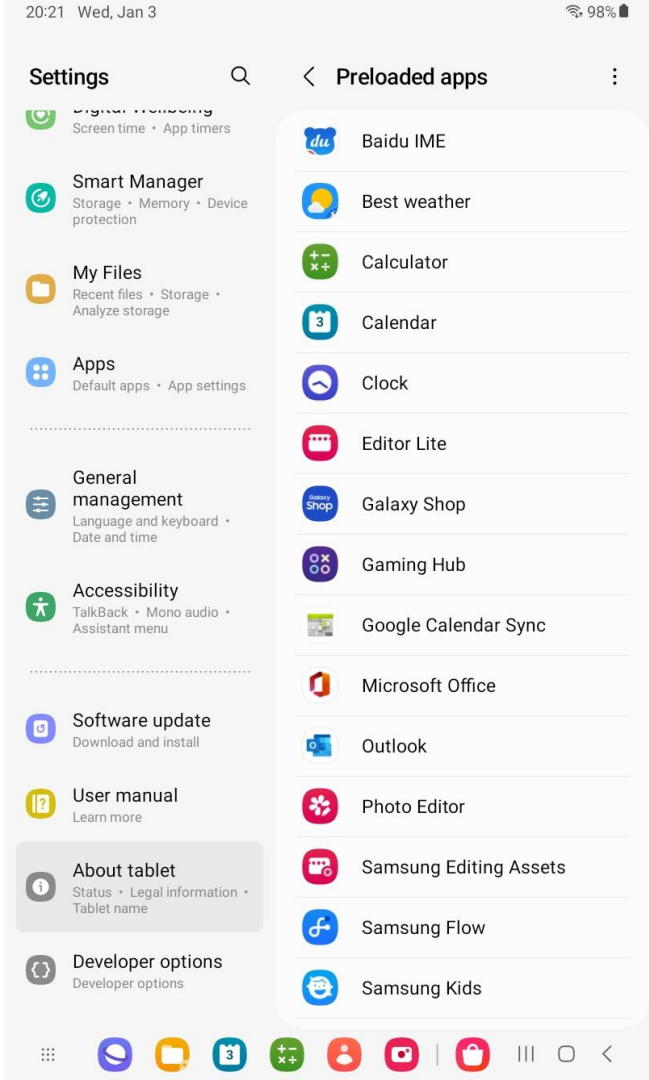
- Same problems but… wrapped in TLS

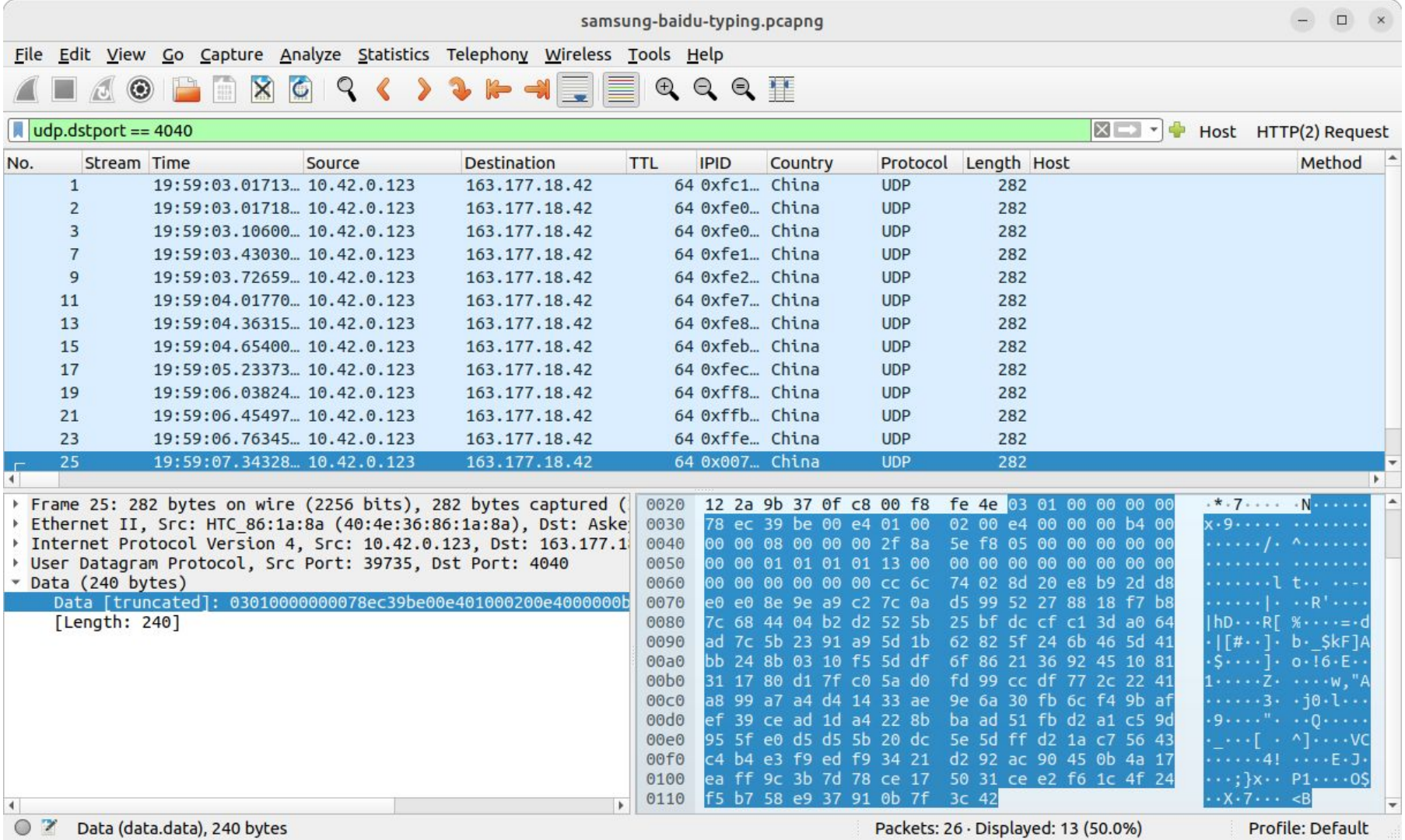


Keyboard Craptography Part 3: Baidu

- Samsung
- Windows
- Android
- iOS

Baidu craptography for Samsung




samsung-baidu-typing.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.dstport == 4040
Host
HTTP(2) Request
No. Stream Time Source Destination TTL IPID Country Protocol Length Host Method
1 19:59:03.01713… 10.42.0.123 163.177.18.42 64 0xfc1… China UDP 282
2 19:59:03.01718… 10.42.0.123 163.177.18.42 64 0xfe0… China UDP 282
3 19:59:03.10600… 10.42.0.123 163.177.18.42 64 0xfe0… China UDP 282
7 19:59:03.43030… 10.42.0.123 163.177.18.42 64 0xfe1… China UDP 282
9 19:59:03.72659… 10.42.0.123 163.177.18.42 64 0xfe2… China UDP 282
11 19:59:04.01770… 10.42.0.123 163.177.18.42 64 0xfe7… China UDP 282
13 19:59:04.36315… 10.42.0.123 163.177.18.42 64 0xfe8… China UDP 282
15 19:59:04.65400… 10.42.0.123 163.177.18.42 64 0xfeb… China UDP 282
17 19:59:05.23373… 10.42.0.123 163.177.18.42 64 0xfec… China UDP 282
19 19:59:06.03824… 10.42.0.123 163.177.18.42 64 0xff8… China UDP 282
21 19:59:06.45497… 10.42.0.123 163.177.18.42 64 0xffb… China UDP 282
23 19:59:06.76345… 10.42.0.123 163.177.18.42 64 0xffe… China UDP 282
25 19:59:07.34328… 10.42.0.123 163.177.18.42 64 0x007… China UDP 282
Frame 25: 282 bytes on wire (2256 bits), 282 bytes captured (
Ethernet II, Src: HTC_86:1a:8a (40:4e:36:86:1a:8a), Dst: Aske
Internet Protocol Version 4, Src: 10.42.0.123, Dst: 163.177.1
User Datagram Protocol, Src Port: 39735, Dst Port: 4040
Data (240 bytes)
Data [truncated]: 03010000000078ec39be00e401000200e4000000b
[Length: 240]
0020 12 2a 9b 37 0f c8 00 f8 fe 4e 03 01 00 00 00 00
0030 78 ec 39 be 00 e4 01 00 02 00 e4 00 00 00 b4 00
0040 00 00 08 00 00 00 2f 8a 5e f8 05 00 00 00 00 00
0050 00 00 01 01 01 01 13 00 00 00 00 00 00 00 00 00
0060 00 00 00 00 00 00 cc 6c 74 02 8d 20 e8 b9 2d d8
0070 e0 e0 8e 9e a9 c2 7c 0a d5 99 52 27 88 18 f7 b8
0080 7c 68 44 04 b2 d2 52 5b 25 bf dc cf c1 3d a0 64
0090 ad 7c 5b 23 91 a9 5d 1b 62 82 5f 24 6b 46 5d 41
00a0 bb 24 8b 03 10 f5 5d df 6f 86 21 36 92 45 10 81
00b0 31 17 80 d1 7f c0 5a d0 fd 99 cc df 77 2c 22 41
00c0 a8 99 a7 a4 d4 14 33 ae 9e 6a 30 fb 6c f4 9b af
00d0 ef 39 ce ad 1d a4 22 8b ba ad 51 fb d2 a1 c5 9d
00e0 95 5f e0 d5 d5 5b 20 dc 5e 5d ff d2 1a c7 56 43
00f0 c4 b4 e3 f9 ed f9 34 21 d2 92 ac 90 45 0b 4a 17
0100 ea ff 9c 3b 7d 78 ce 17 50 31 ce e2 f6 1c 4f 24
0110 f5 b7 58 e9 37 91 0b 7f 3c 42
Data (data.data), 240 bytes
Packets: 26 · Displayed: 13 (50.0%)
Profile: Default

```
signed int __fastcall GA10_aesv1_encrypt(struct s_cloud_tool *cloud_tool, int *plain_crypt_buf, int crypt_len)
{
  struct s_cloud_tool *cloud_tool_; // r5@1
  int *plain_crypt_buf_; // r4@1
  int plain_len_; // r7@1
  signed int result; // r0@2
  int uninit1; // r2@4
  int uninit2; // r3@4
  int i; // r2@5
  unsigned __int8 fixed_key[16]; // [sp+4h] [bp-2Ch]@6

  cloud_tool_ = cloud_tool;
  plain_crypt_buf_ = plain_crypt_buf;
  plain_len_ = crypt_len;
  GA60(cloud_tool);
  if ( !cloud_tool_
    || (unsigned __int8)plain_crypt_buf_ & 3
    || get_crypt_len((int)cloud_tool_, *plain_crypt_buf_) != plain_len_ )
  {
    return -1;
  }
  GA61_generate_key(cloud_tool_, plain_crypt_buf_, uninit1, uninit2);
  GB02_encrypt(
    cloud_tool_->aes,
    (unsigned int *)plain_crypt_buf_ + 5,
    *plain_crypt_buf_,
    (unsigned __int8 *)plain_crypt_buf_ + 4);
  i = (unsigned __int8)plain_crypt_buf_ & 3;
  do
  {
    fixed_key[i] = ~(_BYTE)i ^ ((unsigned int)(1937 * i) >> (i & 3)) * (i + 11);
    ++i;
  }
  while ( i != 16 );
  GB02_encrypt(cloud_tool_->aes, (unsigned int *)plain_crypt_buf_ + 1, 16, fixed_key);
  result = 0;
  *plain_crypt_buf_ ^= plain_crypt_buf_[4] ^ plain_crypt_buf_[1] ^ plain_crypt_buf_[2] ^ plain_crypt_buf_[3];
  return result;
}
```

Baidu craptography for Samsung

- Randomly generate “AES” key k_1
- “Generate” “AES” key k_2
- “AES”-encrypt k_1 with k_2
- Encrypt message with k_1
- Transmit encrypted k_1 and encrypted message

Baidu craptography for Samsung

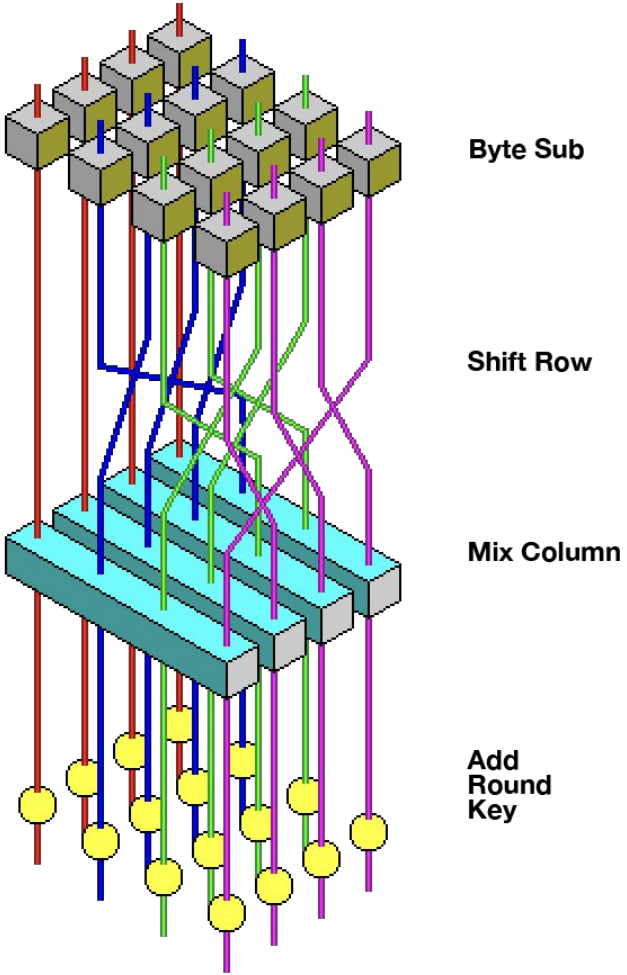
- How does recipient know k_2?

```
void __cdecl generate_static_key(unsigned __int8 *output, bool flag)
{
  unsigned int i; // eax
  unsigned int v3; // edi
  unsigned int v4; // edx

  i = 0;
  v3 = 0;
  do
  {
    v4 = v3;
    v3 += !flag + 1937;
    output[i] = ~(_BYTE)i ^ ((i + 11) * (v4 >> (i & 3)));
    ++i;
  }
  while ( i < 16 );
}
```

Baidu craptography for Samsung

- “AES”?
- Modified AES with additional permutations…
- Security through obscurity…



```
{0: [800,
     1276,
     10,
     0,
     '92F8EE78F1DDCBE74CFEB1166F70883D%7C0',
     'a1|SM-T220-gta7litewifi|320',
     '8.5.20.4',
     'com.android.settings.intelligence',
     '1012497q',
     b'',
     ['2你好惨又热大腿'],
     b''],
 1: [0, b'', 'nihaocanyoureadthis']}
```

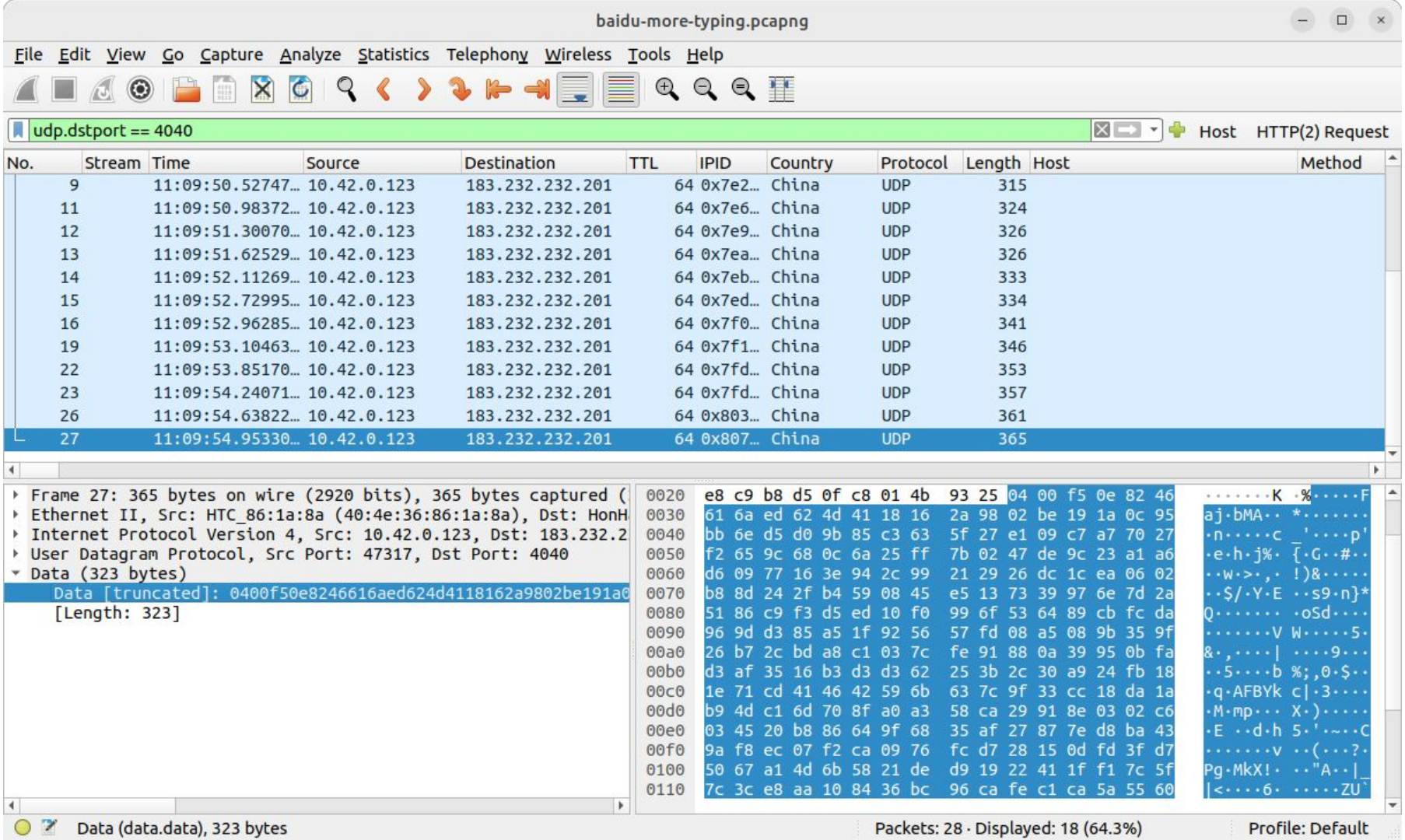
Baidu craptography for Windows

- Mostly cosmetic differences versus Baidu on Samsung
- “AES”v2
- Instead of additional permutations…
- One fewer round

```
[...]
2 {
    1: "nihaocanyoureadthis"
    5: 3407918
  }
3 {
    1: 107
    2: 10
    5: 1
  }
4 {
    1: "1133d4c64afbf1feda85d3c497dd6164|0"
    2: "wn1||0"
    3: "6.0.3.44"
    4: "notepad.exe"
  }
[...]
```

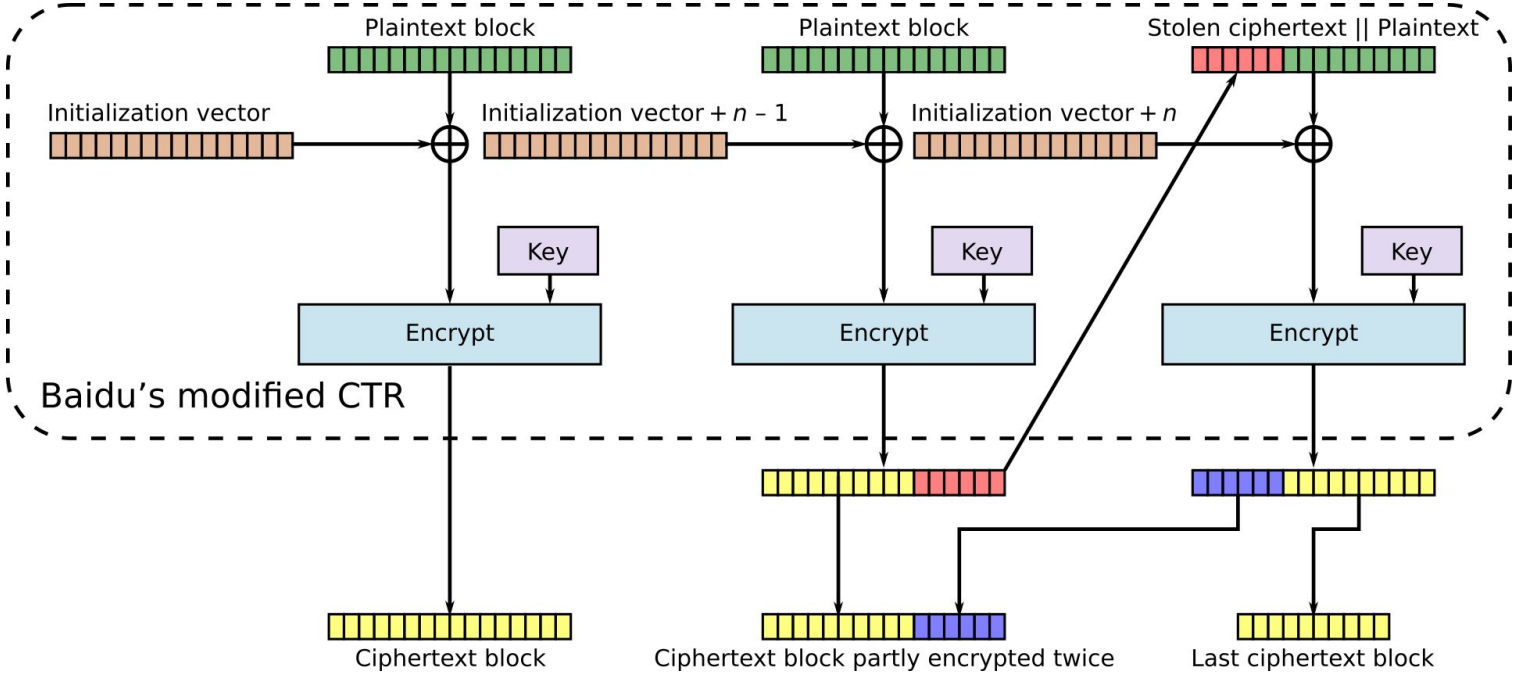
Baidu craptography for Android and iOS

- Uses an upgraded protocol


baidu-more-typing.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
udp.dstport == 4040
Host
HTTP(2) Request
No. Stream Time Source Destination TTL IPID Country Protocol Length Host Method
9 11:09:50.52747… 10.42.0.123 183.232.232.201 64 0x7e2… China UDP 315
11 11:09:50.98372… 10.42.0.123 183.232.232.201 64 0x7e6… China UDP 324
12 11:09:51.30070… 10.42.0.123 183.232.232.201 64 0x7e9… China UDP 326
13 11:09:51.62529… 10.42.0.123 183.232.232.201 64 0x7ea… China UDP 326
14 11:09:52.11269… 10.42.0.123 183.232.232.201 64 0x7eb… China UDP 333
15 11:09:52.72995… 10.42.0.123 183.232.232.201 64 0x7ed… China UDP 334
16 11:09:52.96285… 10.42.0.123 183.232.232.201 64 0x7f0… China UDP 341
19 11:09:53.10463… 10.42.0.123 183.232.232.201 64 0x7f1… China UDP 346
22 11:09:53.85170… 10.42.0.123 183.232.232.201 64 0x7fd… China UDP 353
23 11:09:54.24071… 10.42.0.123 183.232.232.201 64 0x7fd… China UDP 357
26 11:09:54.63822… 10.42.0.123 183.232.232.201 64 0x803… China UDP 361
27 11:09:54.95330… 10.42.0.123 183.232.232.201 64 0x807… China UDP 365
Frame 27: 365 bytes on wire (2920 bits), 365 bytes captured (
Ethernet II, Src: HTC_86:1a:8a (40:4e:36:86:1a:8a), Dst: HonH
Internet Protocol Version 4, Src: 10.42.0.123, Dst: 183.232.2
User Datagram Protocol, Src Port: 47317, Dst Port: 4040
Data (323 bytes)
Data [truncated]: 0400f50e8246616aed624d4118162a9802be191a0
[Length: 323]
0020 e8 c9 b8 d5 0f c8 01 4b 93 25 04 00 f5 0e 82 46
0030 61 6a ed 62 4d 41 18 16 2a 98 02 be 19 1a 0c 95
0040 bb 6e d5 d0 9b 85 c3 63 5f 27 e1 09 c7 a7 70 27
0050 f2 65 9c 68 0c 6a 25 ff 7b 02 47 de 9c 23 a1 a6
0060 d6 09 77 16 3e 94 2c 99 21 29 26 dc 1c ea 06 02
0070 b8 8d 24 2f b4 59 08 45 e5 13 73 39 97 6e 7d 2a
0080 51 86 c9 f3 d5 ed 10 f0 99 6f 53 64 89 cb fc da
0090 96 9d d3 85 a5 1f 92 56 57 fd 08 a5 08 9b 35 9f
00a0 26 b7 2c bd a8 c1 03 7c fe 91 88 0a 39 95 0b fa
00b0 d3 af 35 16 b3 d3 d3 62 25 3b 2c 30 a9 24 fb 18
00c0 1e 71 cd 41 46 42 59 6b 63 7c 9f 33 cc 18 da 1a
00d0 b9 4d c1 6d 70 8f a0 a3 58 ca 29 91 8e 03 02 c6
00e0 03 45 20 b8 86 64 9f 68 35 af 27 87 7e d8 ba 43
00f0 9a f8 ec 07 f2 ca 09 76 fc d7 28 15 0d fd 3f d7
0100 50 67 a1 4d 6b 58 21 de d9 19 22 41 1f f1 7c 5f
0110 7c 3c e8 aa 10 84 36 bc 96 ca fe c1 ca 5a 55 60
Data (data.data), 323 bytes
Packets: 28 · Displayed: 18 (64.3%)
Profile: Default

Baidu craptography for Android and iOS

- “AES”v3
- Uses modified CTR mode



Baidu craptography for Android and iOS

- Modified CTR mode fails to have cryptographic diffusion

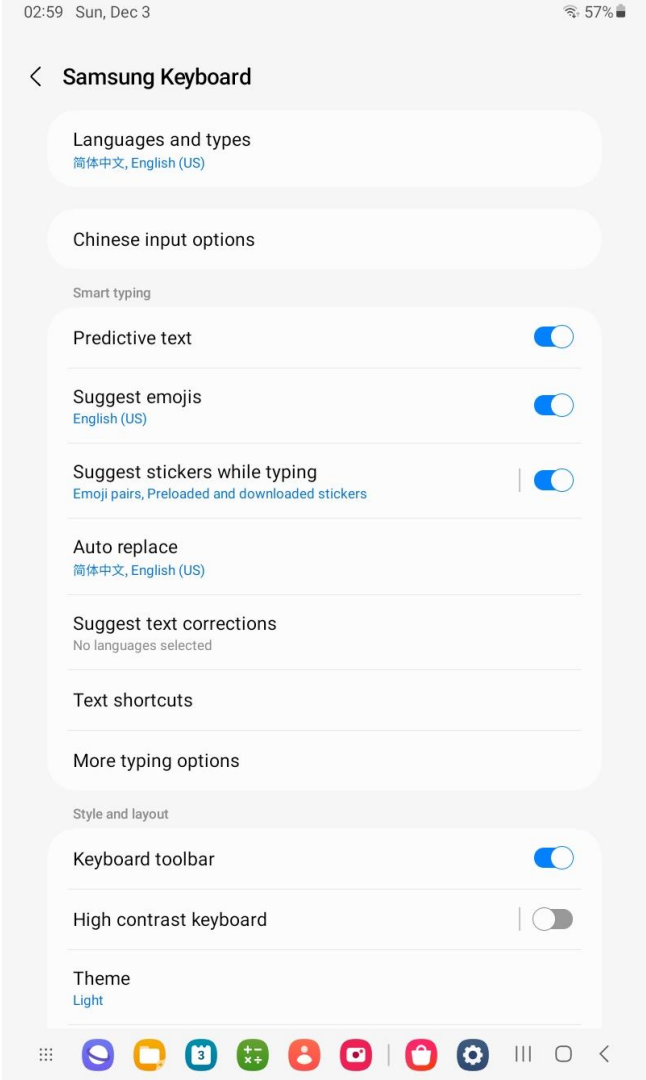
Block	Plaintext																Ciphertext															
0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	e2	d4	00	1c	c6	5d	80	33	0c	b9	48	7d	d5	27	72	7a
1	01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	e2	d4	00	1c	c6	5d	80	33	0c	b9	48	7d	d5	27	72	7a

- IV and key re-use: plaintexts with same first N blocks will encrypt to the same first N ciphertext blocks

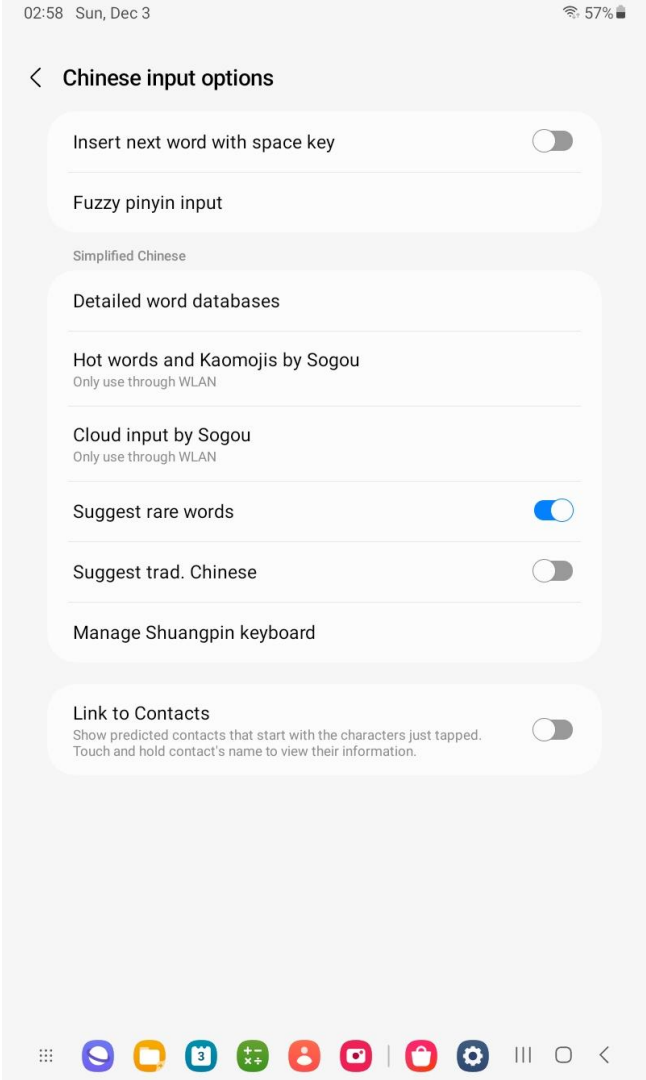
Baidu craptography for Android and iOS

- Static elliptic-curve Diffie-Hellman! But…
- IV and key are re-used across application lifetime
- No forward secrecy (pinned static server key)
- Lack of message integrity (only CRC32)
- Modified CTR mode fails to have cryptographic diffusion

Keyboard Craptography Part 4: Samsung IME



Keyboard Craptography Part 4: Samsung IME



Wireshark · Follow HTTP Stream (tcp.stream eq 1) · samsung.pcapng

```
1602841941J.POST /web_ime/mobile_pb.php?durtot=327&h=8f2bc112-bbec-3f96-86ca-652e98316ad8&r
=android_oem_samsung_open&v=8.13.10038.413173&s=&e=&i=&fc=0&base=dW5rbm93biswLjArMC4w&ext_v
er=0 HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0 (Linux; U; Android 13; SM-T220 Build/TP1A.220624.014)
Host: shouji.sogou.com
Connection: Keep-Alive
Accept-Encoding: gzip
Content-Length: 167


\
$8f2bc112-bbec-3f96-86ca-652e98316ad8..android_oem_samsung_open..8.13.10038.413173".999(.8.
".com.tencent.mobileqq:.nihaocanyoureadthis..
..
......(.H.......
.0".327HTTP/1.1 200 OK
Date: Sat, 07 Oct 2023 19:10:13 GMT
Content-Type: application/octet-stream
Content-Length: 200
Connection: keep-alive

.........i2V
.`O}Y.`fTr.e.a.d.t.h.i.s.....u...s.............................
.............(..0
h..:.
.`O}Y.`fTr.e.a.d.t.h.i.s...BD
.......Kf..[........... ..
.......Kf..[.....,....... .."
1602841941J.
```

12 client pkts, 12 server pkts, 23 turns.

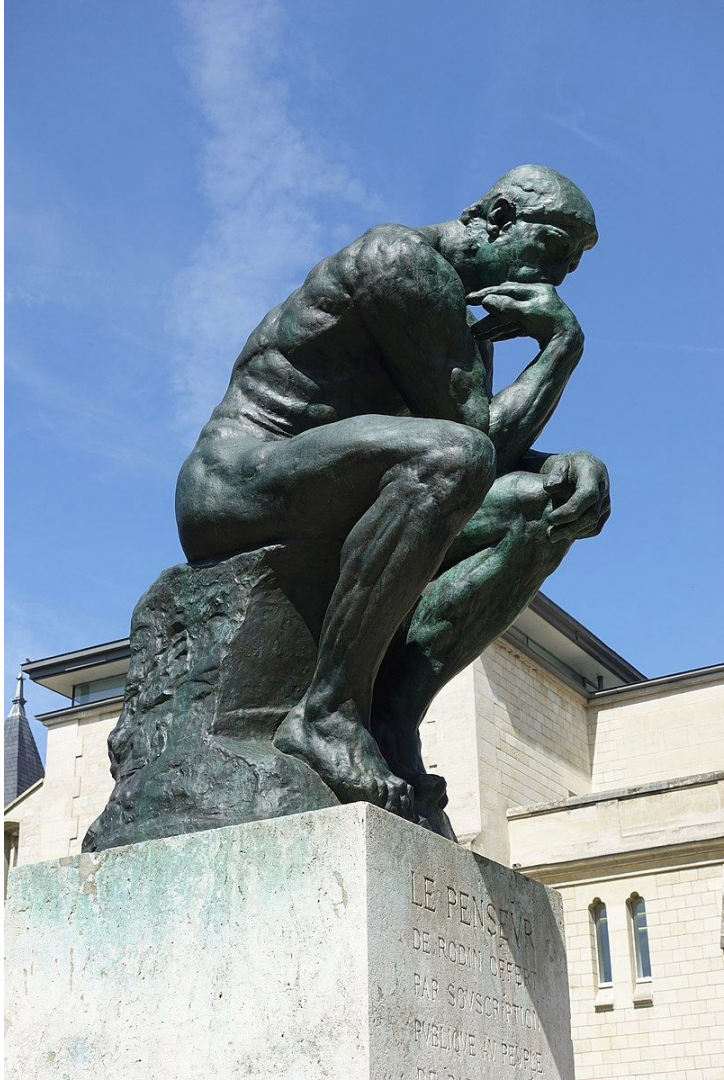
Entire conversation (11 kB) | Show data as ASCII | Stream 1

Find: | Find Next

Help | Filter Out This Stream | Print | Save as... | Back | Close

Is it even craptography?

If there is no cryptography?



Generalizing attacks

Discussed attacks can generally also be applied to…

- incoming data
- spoofing data
- modifying data

“Please do not make it public”

Comment



tsrc审核 2023-06-25 17:10:23

Thank you for your interest in Tencent security. There is no low or low security risk for this issue. We look forward to your next more exciting report.



tsrc审核 2023-06-26 11:52:30

Sorry, my previous reply was wrong, we are dealing with this vulnerability, please do not make it public, thank you very much for your report

Legend	
✘✘	working exploit created to decrypt transmitted keystrokes for both **active and passive** eavesdroppers
✘	working exploit created to decrypt transmitted keystrokes for an **active** eavesdropper
!	weaknesses present in cryptography implementation
✔	no known issues
N/A	product not offered or not present on device analyzed

Keyboard developer	Android	iOS	Windows
Tencent	✘	✔	✘
Baidu	!	!	✘✘
iFlytek	✘✘	✔	✔

Legend	
✘✘	working exploit created to decrypt transmitted keystrokes for both **active and passive** eavesdroppers
✘	working exploit created to decrypt transmitted keystrokes for an **active** eavesdropper
!	weaknesses present in cryptography implementation
✔	no known issues or all known issues fixed
N/A	product not offered or not present on device analyzed

Keyboard developer	Android	iOS	Windows
Tencent	✔	✔	✔
Baidu	!	!	!
iFlytek	✔	✔	✔

Legend	
✘✘	working exploit created to decrypt transmitted keystrokes for both **active and passive** eavesdroppers
✘	working exploit created to decrypt transmitted keystrokes for an **active** eavesdropper
!	weaknesses present in cryptography implementation
✔	no known issues
N/A	product not offered or not present on device analyzed

Device manufacturer	Own	Sogou	Baidu	iFlytek	iOS	Windows
Samsung	✘✘	✔*	✘✘	N/A	N/A	N/A
Huawei	✔*	✔	N/A	N/A	N/A	N/A
Xiaomi	N/A	✘*	✘✘	✘✘	N/A	N/A
OPPO	N/A	✘	✘✘*	N/A	N/A	N/A
Vivo	✔*	✘	N/A	N/A	N/A	N/A
Honor	N/A	N/A	✘✘*	N/A	N/A	N/A

* Default keyboard on device

Legend	
✘✘	working exploit created to decrypt transmitted keystrokes for both **active and passive** eavesdroppers
✘	working exploit created to decrypt transmitted keystrokes for an **active** eavesdropper
!	weaknesses present in cryptography implementation
✔	no known issues or all known issues fixed
N/A	product not offered or not present on device analyzed

Device manufacturer	Own	Sogou	Baidu	iFlytek	iOS	Windows
Samsung	✔	✔*	!	N/A	N/A	N/A
Huawei	✔*	✔	N/A	N/A	N/A	N/A
Xiaomi	N/A	✔*	!	✔	N/A	N/A
OPPO	N/A	✔	!*	N/A	N/A	N/A
Vivo	✔*	✔	N/A	N/A	N/A	N/A
Honor	N/A	N/A	✘✘*	N/A	N/A	N/A

* Default keyboard on device

Let’s zoom out a bit!

Most downloaded apps in 2023?

1		
2		
3		
4		
5		
6		

7		
8		
9		
10		
11		
12		

Source: https://www.businessofapps.com/data/most-popular-apps/ (1/30/2024)

Most downloaded apps in 2023?

1	WeChat	1012
2	Alipay	901
3	Taobao	795
4	Pinduoduo	728
5	Instagram	696
6	Douyin	695

7	TikTok	654
8	QQ	583
9	Facebook	553
10	Baidu	491
11	Kuaishou	480
12	WhatsApp	475

Most Popular Apps Key Statistics

- Instagram was the most downloaded app globally in 2023, with 696 million downloads

Source: https://www.businessofapps.com/data/most-popular-apps/ (1/30/2024)

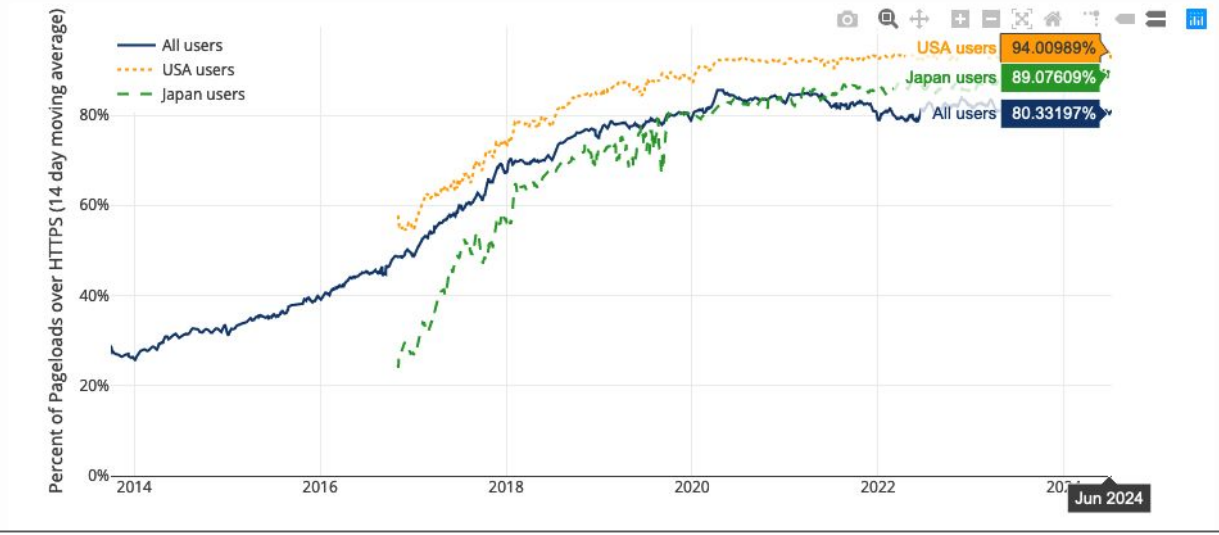
ELECTRONIC FRONTIER FOUNDATION

HTTPS Is Actually Everywhere

SEPTEMBER 21, 2021

Percentage of Web Pages Loaded by Firefox Using HTTPS

(14-day moving average, source: Firefox Telemetry)



How many always use HTTPS/TLS?

	WeChat	1012
	Alipay	901
	Taobao	795
	Pinduoduo	728
	Instagram	696
	Douyin	695

	TikTok	654
	QQ	583
	Facebook	553
	Baidu	491
	Kuaishou	480
	WhatsApp	475

Source: https://www.businessofapps.com/data/most-popular-apps/ (1/30/2024)

How many always use HTTPS/TLS?

❌	WeChat	1012
❌	Alipay	901
❌	Taobao	795
❌	Pinduoduo	728
✅	Instagram	696
✅	Douyin	695

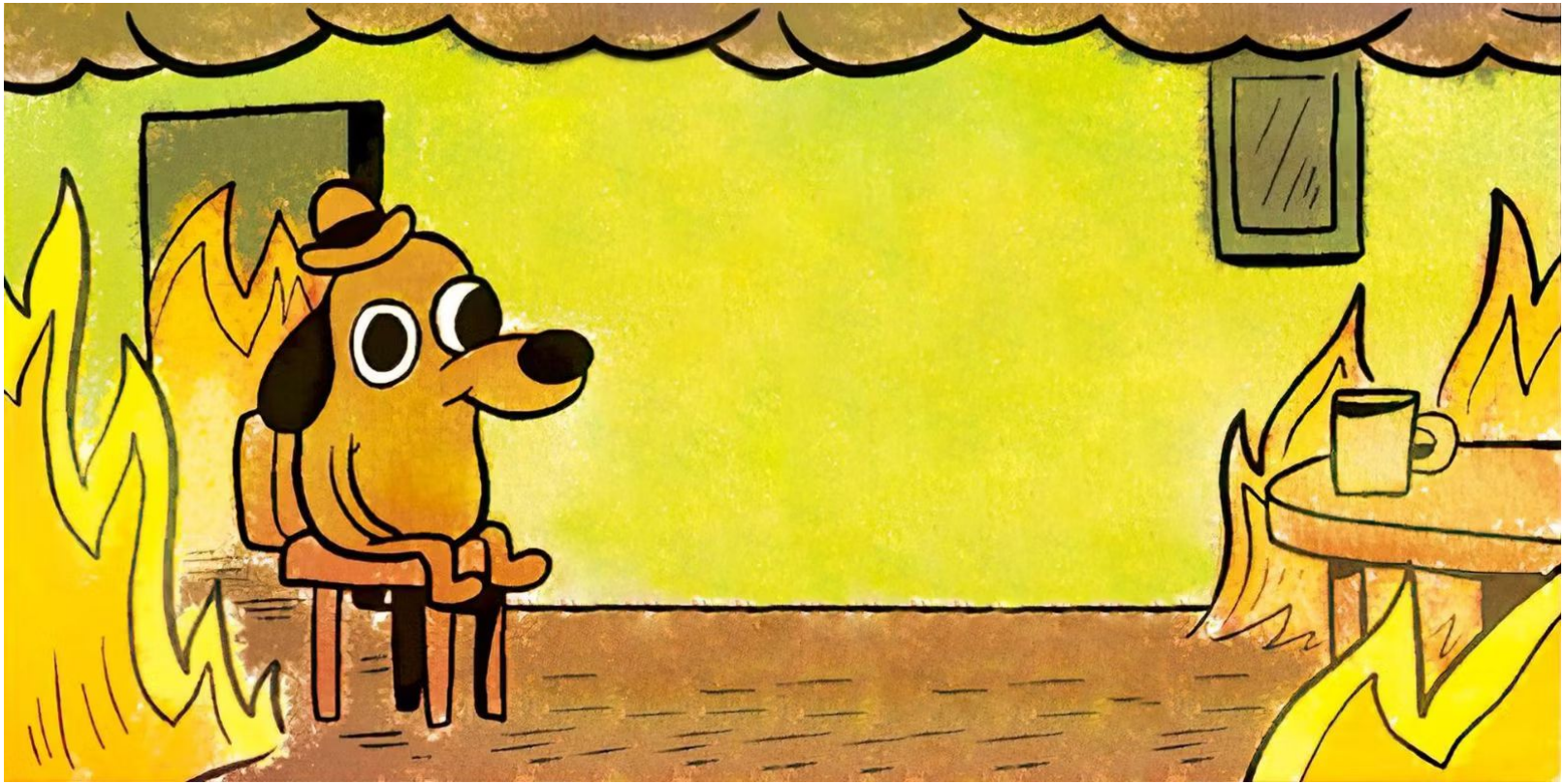
✅	TikTok	654
❌	QQ	583
✅	Facebook	553
❌	Baidu	491
❌	Kuaishou	480
✅	WhatsApp	475

*but they're also **not not** encrypting…

many of them are **using proprietary cryptography**

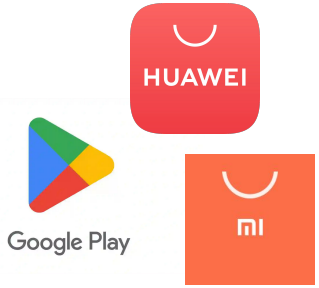
Source: https://www.businessofapps.com/data/most-popular-apps/ (1/30/2024)

Uh-oh

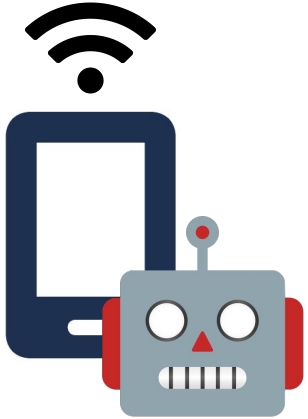


So far, researchers have mostly been conducting one-off studies analyzing proprietary cryptography in individual apps

Can we measure this **systematically**?











Download popular applications

Install app

Simulate user behavior

Isolate non-TLS traffic

Entropy analysis



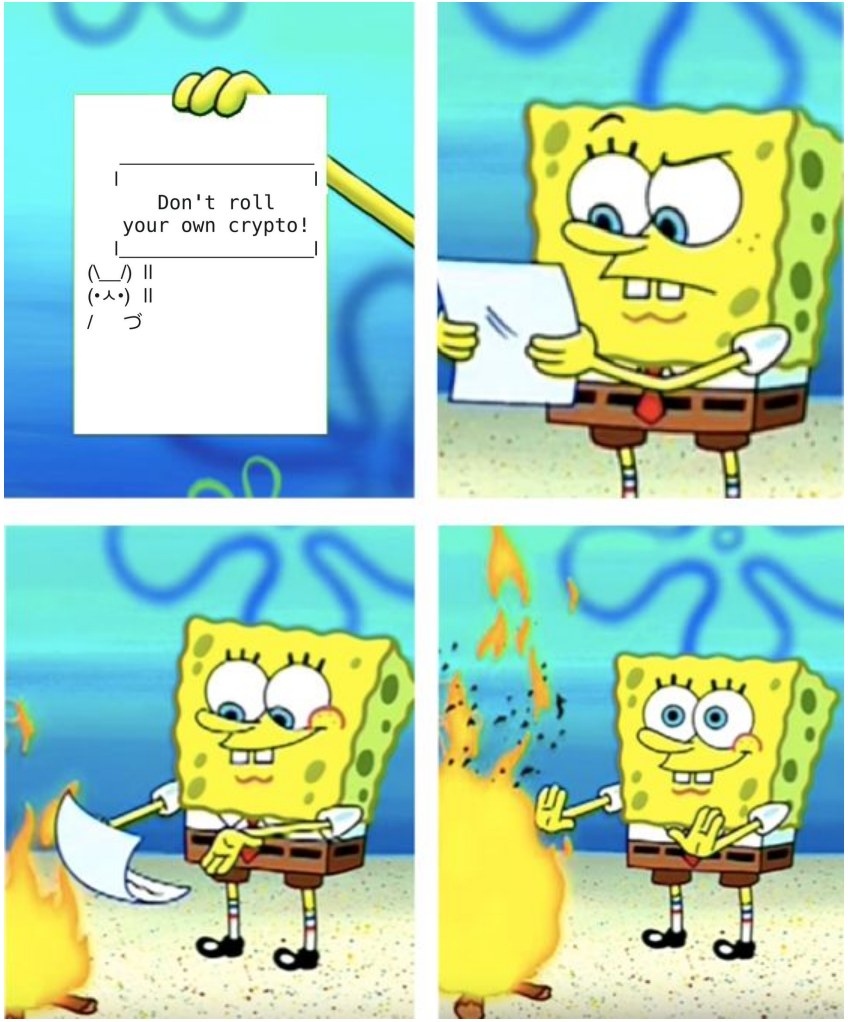
91% of top 45 apps exclusively used standard encryption (QUIC, TLS) to transmit data

---



4% of top 44 apps exclusively used standard encryption

54% used proprietary cryptography


Don't roll
your own crypto!
(__/) ||
(•ㅅ•) ||
/ づ

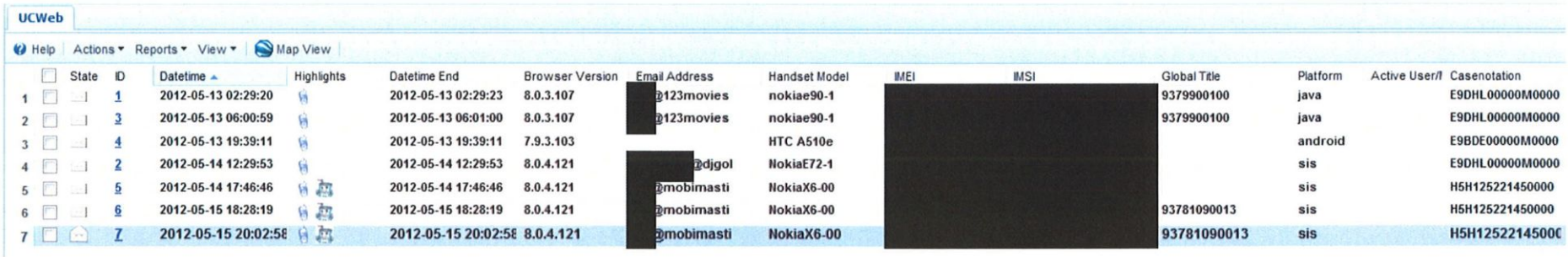
- Are these backdoors?

- Are these backdoors?
  - **No**

Success Stories

* UCWeb mobile browser identification
  * Discovered by GCHQ analyst during DSD workshop
  * Chinese mobile web browser – leaks IMSI, MSISDN, IMEI and device characteristics

UCWeb – XKS Microplugin



UCWeb

Help | Actions | Reports | View | Map View

	State	ID	Datetime	Highlights	Datetime End	Browser Version	Email Address	Handset Model	IMEI	IMSI	Global Title	Platform	Active User/I	Casenotation
1		1	2012-05-13 02:29:20		2012-05-13 02:29:23	8.0.3.107	[redacted]@123movies	nokiae90-1	[redacted]		9379900100	java		E9DHL00000M0000
2		3	2012-05-13 06:00:59		2012-05-13 06:01:00	8.0.3.107	[redacted]@123movies	nokiae90-1			9379900100	java		E9DHL00000M0000
3		4	2012-05-13 19:39:11		2012-05-13 19:39:11	7.9.3.103		HTC A510e				android		E9BDE00000M0000
4		2	2012-05-14 12:29:53		2012-05-14 12:29:53	8.0.4.121	[redacted]@djgol	NokiaE72-1				sis		E9DHL00000M0000
5		5	2012-05-14 17:46:46		2012-05-14 17:46:46	8.0.4.121	[redacted]@mobimasti	NokiaX6-00				sis		H5H125221450000
6		6	2012-05-15 18:28:19		2012-05-15 18:28:19	8.0.4.121	[redacted]@mobimasti	NokiaX6-00			93781090013	sis		H5H125221450000
7		7	2012-05-15 20:02:58		2012-05-15 20:02:58	8.0.4.121	[redacted]@mobimasti	NokiaX6-00			93781090013	sis		H5H12522145000

Why is this happening?

- Many of these applications became massively popular around the early 2010s– before TLS was de-facto standard
- Anti-scraping/competition
- Inertia
- ???

How do we stop it from being bad?

1. Find the problems
   - Security researchers should pay more attention to these **massively popular** but understudied apps
   - Any researcher that even looked at this traffic in Wireshark would know there is a problem

How do we stop it from being bad?

2. Report the problems

- Many did switch to TLS when we reported severe vulns, some did not
- We need to better engage with these companies and put pressure on them to design better products

How do we stop it from being bad?

3. Prevent future problems?

- Can platforms, app store enforcement, etc. impose restrictions on the nature of app's network access?
- "Don't roll your own crypto" – how do we spread this message?

For our full report…

https://citizenlab.ca/2024/04/vulnerabilities/

