Regarding Baidu IME's encryption protocol, we stand by our position that it is significantly weaker than state-of-the-art encryption protocols.

Regarding performance and reliability, we understand your performance constraints. In fact, many international network cryptography developers have found the same issues with TCP+TLS. Now, cryptography experts recommend QUIC+TLS, which has one fewer round trip for connection establishment. With session resumption, QUIC+TLS can also establish connections within 0RTT. We have also noticed that many other popular applications like WeChat are using QUIC+TLS under certain circumstances.

Here is a blog post about this topic:
https://blog.cloudflare.com/even-faster-connection-establishment-with-quic-0-rtt-resumption

In addition, many network middleboxes and firewalls block unknown UDP or TCP protocols. Due to the growing popularity of QUIC, middleboxes may be less likely to interfere with QUIC traffic, making it a more reliable network transport.

Regarding privacy and security, QUIC+TLS provides much stronger security and privacy guarantees. Given the sensitive nature of the data being transmitted, we cannot recommend Baidu's current encryption protocol.

In our report, we detail multiple issues related to Baidu IME's current encryption protocol. In particular, we note that it lacks the following cryptographic properties:

- Lack of CPA-security (due to IV and key re-use)
- Lack of diffusion (due to the use of its nonstandard CTR mode)
- Lack of forward secrecy
- Lack of message integrity

All four of these cryptographic properties are standard expectations of modern transit encryption.

We did not mention the term CPA-security in the report, but we will explain here. Because the mode of encryption is deterministic (as the key and IV are re-used), it cannot be secure against a chosen-plaintext attack, which is the baseline for provable security in a cryptographic scheme.

This means that Baidu IME's encryption scheme leaks information. An attacker could identify, for instance, if two ciphertexts encrypted by Baidu contain the same underlying plaintext; e.g., by comparing the ciphertexts of two encrypted messages, if they match, that means that the plaintexts are the same. Similarly, if any 16-byte ciphertext blocks are the same, that means that the underlying plaintexts are the same.

The sensitive data encrypted by Baidu IME (namely, keystrokes) is highly valuable to a potential attacker. The information leaked via this deterministic encryption mode is significant enough that a sufficiently motivated attacker could learn more about what a particular user is typing.

As a visual example, we performed an entropy analysis of Baidu IME payloads as compared to TLS1.2 data packets and QUIC+TLS data packets, with *N*=1,000 samples each, demonstrated in *Figure 1*.

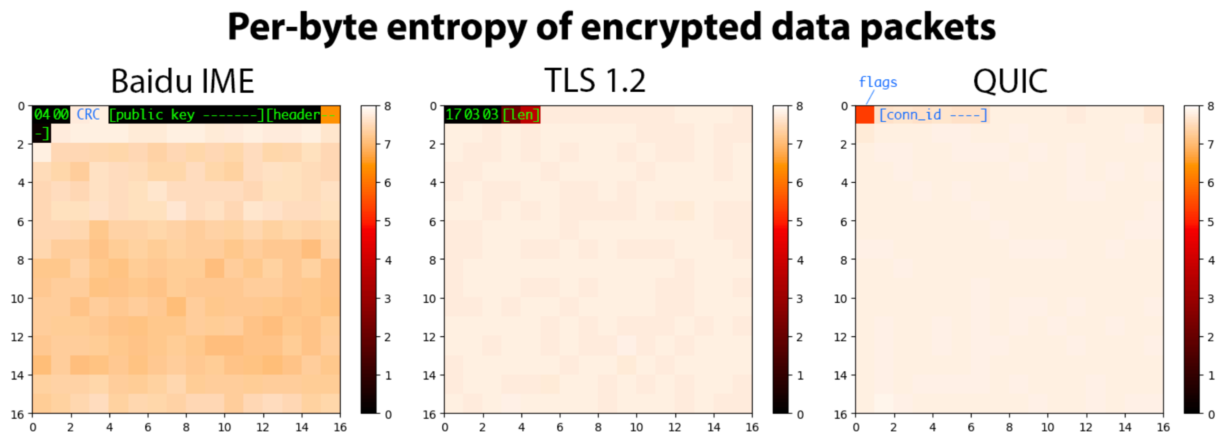

Figure 1: From left-to-right, an annotated visual representation of each byte's entropy for the first 256 bytes of Baidu, TLS1.2, and QUIC data payloads, with N=*1,000 samples each. Differing annotation colors are intended for readability and contrast.*

To generate these images, we calculate the Shannon entropy of each byte of each payload, across *N* observed payloads. The top-left square represents the first byte of the payload, the square in the first row, second column is the second byte of the payload, etc. In the Baidu image, the first two bytes are black because they are always "0x04 0x00", so the entropy is 0. In the TLS image, the first three bytes also have 0 entropy because TLS 1.2 data records always begin with "0x17 0x03 0x03". In the QUIC image, the first byte has less entropy because it is used for various flags; then the following 6 bytes have lower entropy since they represent the *connection ID*.

The maximum entropy for a square is $\log_2 2^8 = 8$ bits since there are 2^8 possible values of a byte. The code to generate these graphs is available here.

To ensure cryptographic protocols do not leak any information about the underlying protocol, **encrypted data should be indistinguishable from random data**. This is the reasoning behind the CPA model of cryptanalysis. The above figure visually demonstrates that the entropy of data encrypted by Baidu is significantly lower than the state-of-the-art, meaning that there is a large amount of information leakage that could be exploited by a motivated actor.

Below, we have reproduced the relevant portion of the original report.

Privacy issues with key and IV re-use

Since the IV and key are both directly derived from the client key pair, the IV and key are reused until the application generates a new key pair. This only happens when the application restarts, such as when the user restarts the mobile device, the user switches to a different keyboard and back, or the keyboard app is evicted from memory. From our testing, we have observed the same key and IV in use for over 24 hours. There are various issues that arise from key and IV reuse.

Re-using the same IV and key means that the same inputs will encrypt to the same encrypted ciphertext. Additionally, due to the way the block cipher is constructed, if blocks in the same positions of the plaintexts are the same, they will encrypt to the same ciphertext blocks. As an example, if the second block of two plaintexts are the same, the second block of the corresponding ciphertexts will be the same.

Weakness in cipher mode

The electronic codebook (ECB) cipher mode is notorious for having the undesirable property that equivalent plaintext blocks encrypt to equivalent ciphertext blocks, allowing patterns in the plaintext to be revealed in the ciphertext (see Figure 2 for an illustration).

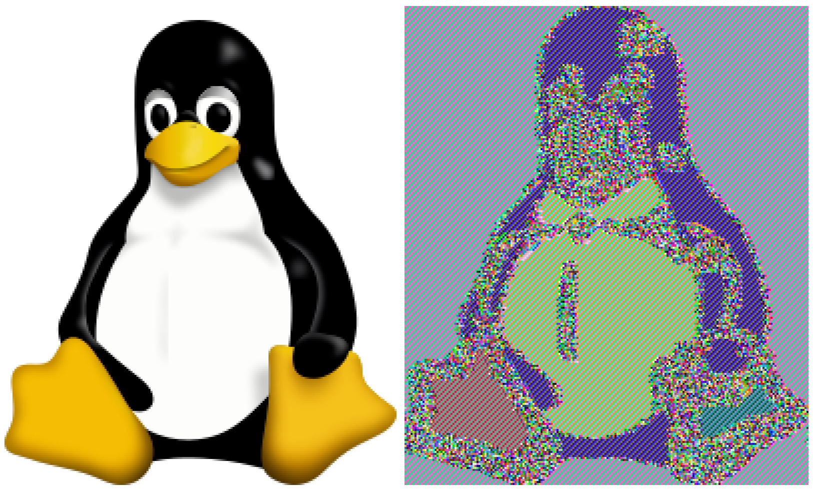
Figure 2: When a bitmap image (left) is encrypted in ECB mode, patterns in the image are still visible in the ciphertext (right). Adapted from these figures.

While BCTR mode used by Baidu does not as flagrantly reveal patterns to the same extent as ECB mode, there do exist circumstances in which patterns in the plaintext can still be revealed in the ciphertext. Specifically, there exist circumstances in which there exists a counter-like pattern in the plaintext which can be revealed by the ciphertext (see Figure 3 for an example). These circumstances are possible due to the fact that (IV + *i*) is XORed with each plaintext block *i* and then encrypted, unlike ordinary CTR mode which encrypts (IV + *i*) and XORs it with the plaintext. Thus, when using BCTR mode, if the plaintext exhibits similar counting patterns as

(IV + *i*), then for multiple blocks the value ((IV + *i*) XOR plaintext block *i*) may be equivalent and thus encrypt to an equivalent ciphertext.

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

Figure 3: When encrypted with the randomly generated key “\x96f\x08\xd1o\x80\x82\x86\xa7\xb7\xdaC\x96\xee\xd1\xa2” and IV “H[T\x92\x0c\x80\xa6)o\x95\xe5\xc5j=\xe2” using Baidu’s modified CTR mode, the above plaintext blocks in positions 0 and 1 encrypt to the same ciphertext.

More generally, BCTR mode fails to provide the cryptographic property of diffusion. Specifically, if an algorithm provides diffusion, then, when we change a single bit of the plaintext, we expect half of the bits of the ciphertext to change. However, the example in Figure 3 illustrates a case where changing a single bit of the plaintext caused zero bits of the ciphertext to change, a clear violation of the expectations of this property. The property of diffusion is vital in secure cryptographic algorithms so that patterns in the plaintext are not visible as patterns in the ciphertext.

Other privacy and security weaknesses

There are other weaknesses in the custom encryption protocol designed by Baidu IME that are not consistent with the expected standards for a modern encryption protocol used by hundreds of millions of devices.

Forward secrecy issues with static Diffie-Hellman

The use of a pinned static server key means that the cipher is not forward secret, a property of other modern network encryption ciphers like TLS. If the server key is ever revealed, any past message where the shared secret was generated with that key can be successfully decrypted.

Lack of message integrity

There are no cryptographically secure message integrity checks, which means that a network attacker may freely modify the ciphertext. There is a CRC32 checksum calculated and included with the plaintext data, but a CRC32 checksum does not provide cryptographic integrity, as it is easy to generate CRC32 checksum collisions. Therefore, modifying the ciphertext may be possible. In combination with the issue concerning key and IV reuse, this protocol may be vulnerable to a swapped block attack.