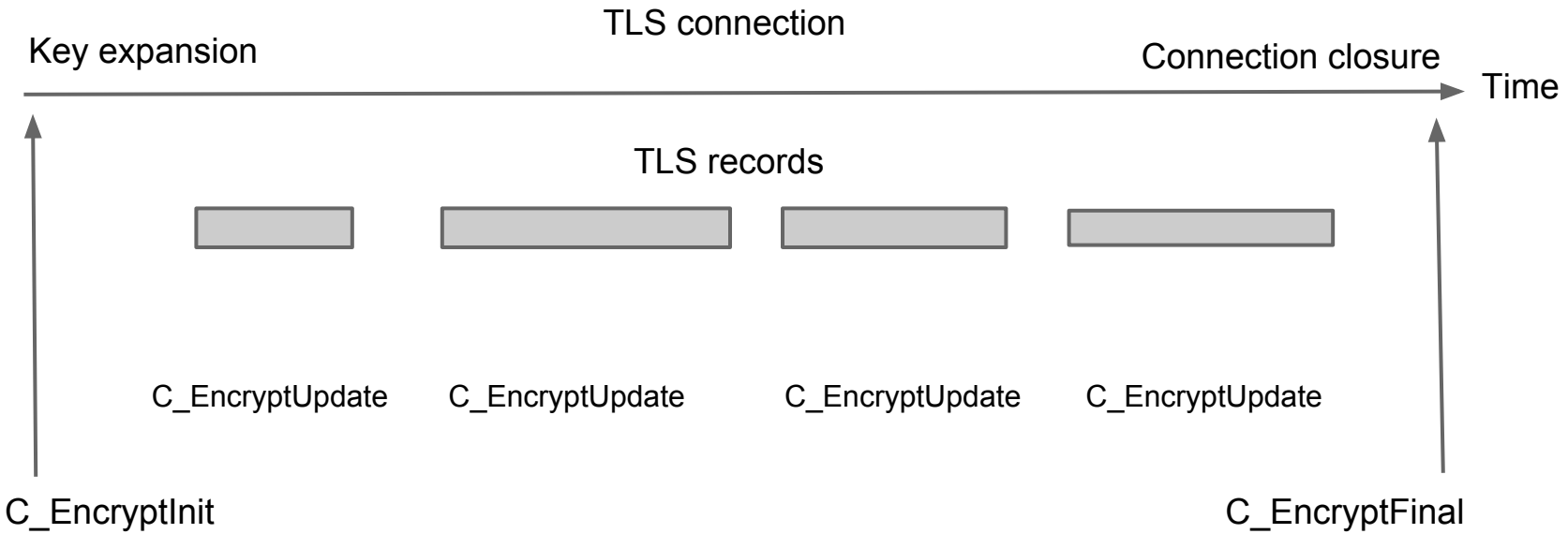
PKCS #11 encryption v2

Wan-Teh Chang <wtc@google.com>

Outline

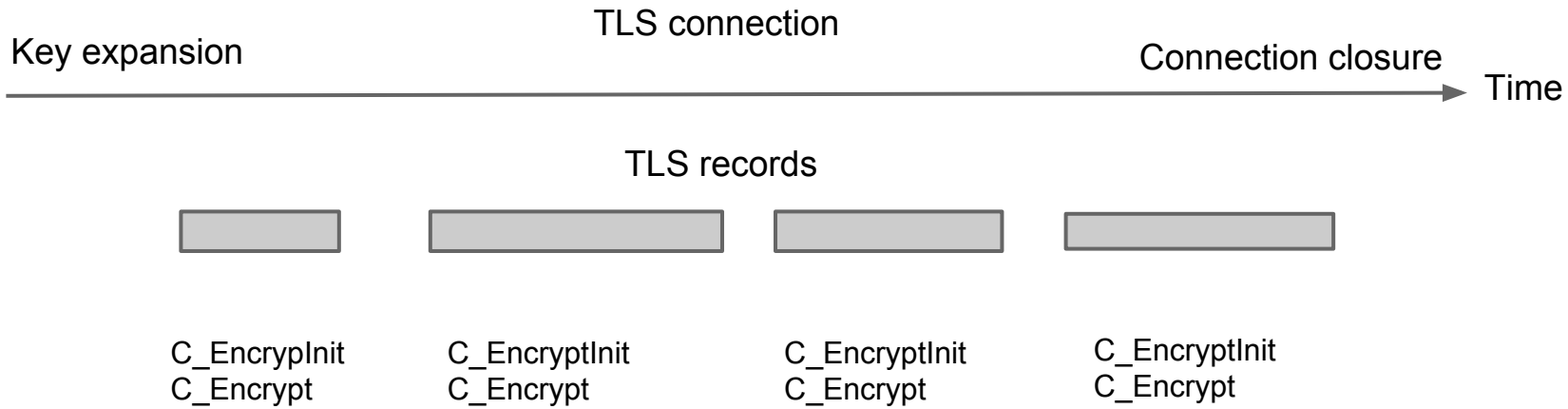
- Use case
- Problems
- Proposed solution

Use case: encrypting TLS records



RC4 stream cipher and CBC block ciphers,
even with per-record explicit IV

Use case: encrypting TLS records



Need a **C_EncrypInit** call for each record to specify the **IV** and **AAD**, even though the **key stays the same**.

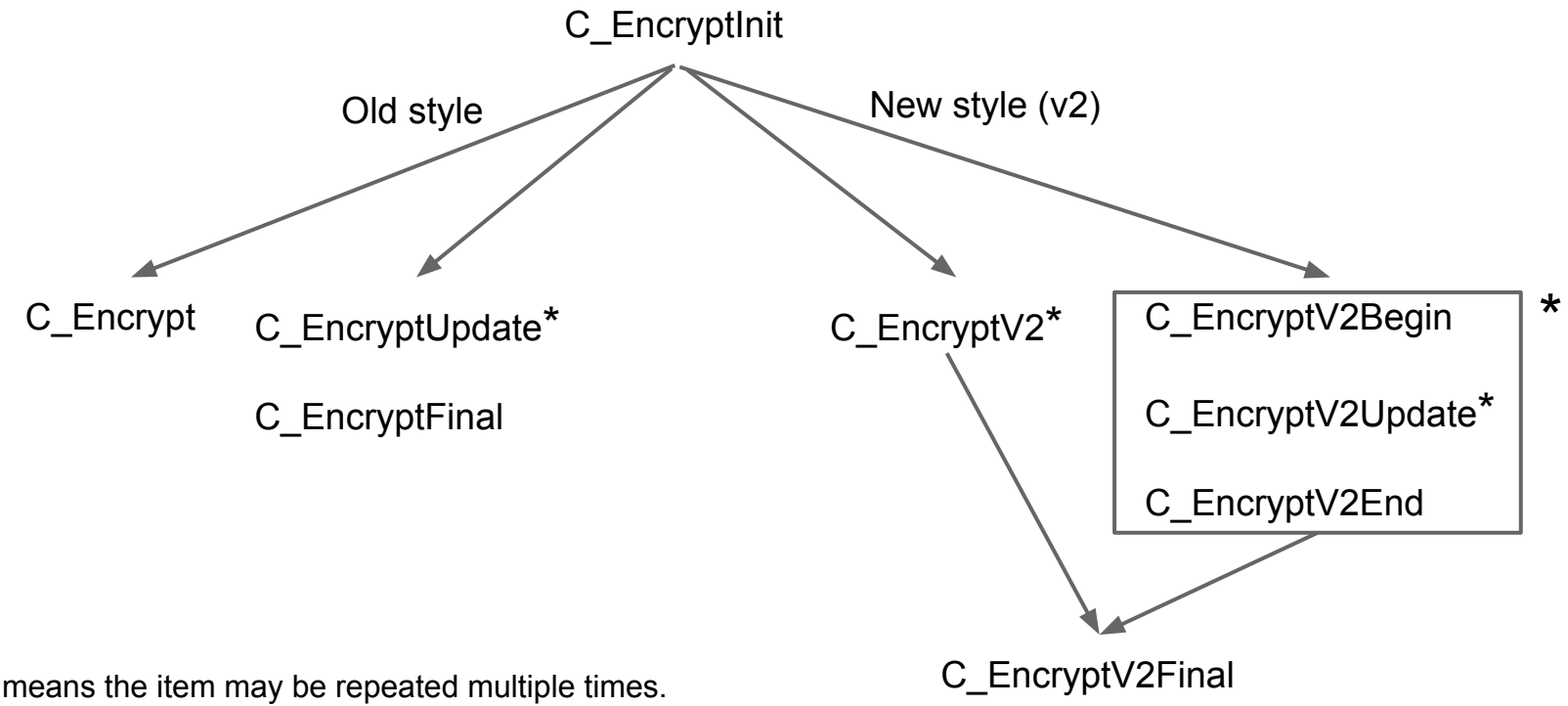
AEAD ciphers such as AES-GCM

Problems

- C_EncryptInit overhead: performs IV-independent initialization repeatedly

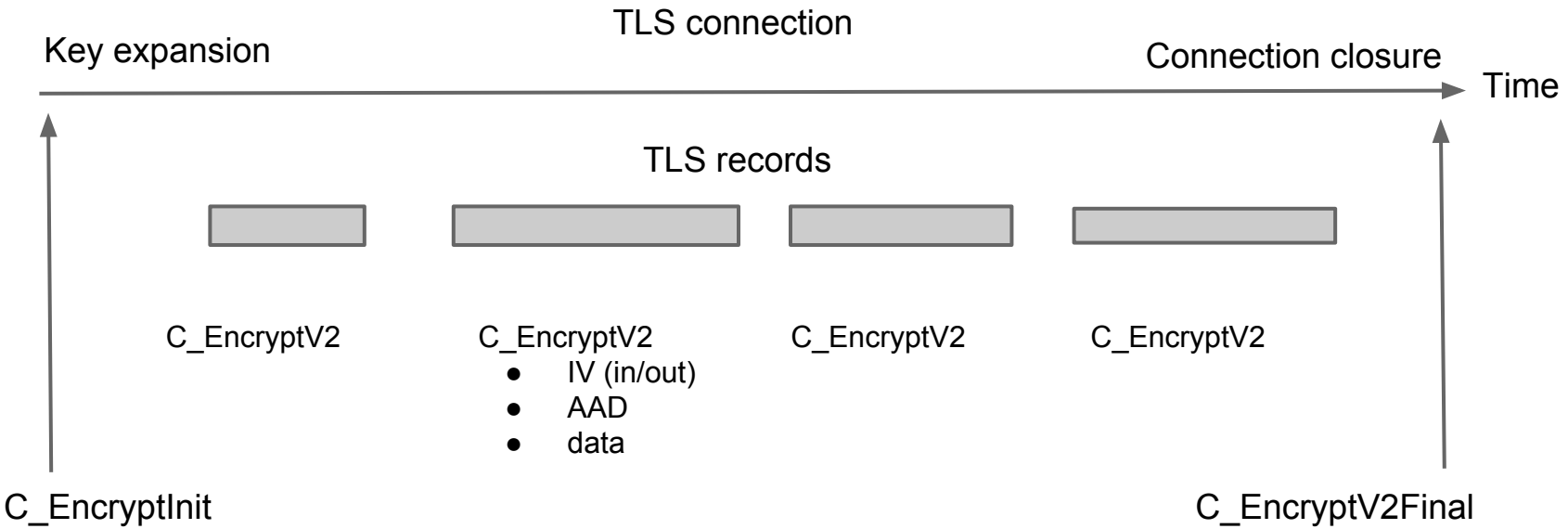
- IV generation: for CTR and GCM, IV must not be repeated

Proposal: v2 encryption



* means the item may be repeated multiple times.

Use case: encrypting TLS records



- IV (in/out)
- AAD
- data

- v2 encryption mechanism, with IV generator
- key

C_EncryptV2 operates in **single-part mode**.

AES GCM and nonce generator

```
#define CKM_AES_GCM_V2 0x00000700

typedef struct CK_GCM_PARAMS_V2 {
  CK_ULONG ulNonceLen;
  CK_ULONG ulTagLen;
  CK_MECHANISM_PTR pNonceMech;
} CK_GCM_PARAMS_V2;


#define CKM_GCM_NONCE_DETERMINISTIC   0x00000750
#define CKM_GCM_NONCE_RBG_BASED       0x00000751

typedef struct CK_GCM_NONCE_DETERMINISTIC_PARAMS {
  CK_BYTE_PTR pFixed;  /* the fixed field */
  CK_ULONG    ulFixedLen;
} CK_GCM_NONCE_DETERMINISTIC_PARAMS;

typedef struct CK_GCM_NONCE_RBG_BASED_PARAMS {
  CK_BYTE_PTR pFree;  /* the free field */
  CK_ULONG    ulFreeLen;
} CK_GCM_NONCE_RBG_BASED_PARAMS;
```