Committing Authenticated Encryption
• What is it? How is it different than regular AE?
• Why is it important? What goes wrong without it?
• Why is an RFC needed? What should the RFC say?

Committing authenticated encryption (cAE):
AE where it’s hard to find a ciphertext with multiple correct decryptions.
cAE => ciphertexts are binding commitments
Many people think of authenticated encryption as a lock box. Fine intuition, if keys are random and hidden:
- The box hides what’s inside it (confidentiality)
- Can’t change already-locked box (integrity)

This intuition fails badly if keys can be adversarial.
“Physical” Encryption

This intuition fails badly if keys can be adversarial. AE has no security with attacker control of keys: ciphertexts can have multiple correct decryptions.

cAE = AE + binds attacker to a single decryption
• What is it? How is it different than regular AE?
• Why is it important? What goes wrong without it?
• Why is an RFC needed? What should the RFC say?

**Committing authenticated encryption (cAE):**
AE where it’s hard to find a ciphertext with multiple correct decryptions.
cAE => ciphertexts are *binding commitments*
CTR mode is not committing

Any ciphertext can be decrypted under any key.

Adding a MAC does not help:

*Galois/Counter Mode (GCM) is not committing.*
CTR mode is not committing

Any ciphertext can be decrypted under any key.

Adding a MAC does not help:

*Galois/Counter Mode (GCM) is not committing.*

<table>
<thead>
<tr>
<th>Scheme</th>
<th>cAE?</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-GCM</td>
<td>No</td>
</tr>
<tr>
<td>ChaCha20/Poly1305</td>
<td>No</td>
</tr>
<tr>
<td>OCB</td>
<td>No</td>
</tr>
<tr>
<td>Encrypt-then-HMAC (distinct keys)</td>
<td>No</td>
</tr>
<tr>
<td>Encrypt-then-HMAC (derived keys)</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Places where cAE is needed

• Message franking (abuse reporting for E2EE Messenger)
  • Lack of cAE led to [DGRW] “invisible salamander” bypass: GCM ciphertext with two image decryptions
  • Similar issues found elsewhere after [DGRW]
• OPAQUE (possible aPAKE standard) fragile without cAE
  • Active MitM of login could learn password from size-N dictionary via log(N) interactions with client (unpublished)
  • Needed in other protocols? (ongoing work)
• cAE ensures transcript consistency in group messaging (MLS may use)
• Widely used in research that may be deployed in the future
• What is it? How is it different than regular AE?
• Why is it important? What goes wrong without it?
• Why is an RFC needed? What should the RFC say?

**Committing authenticated encryption (cAE):**

AE where it’s hard to find a ciphertext with multiple correct decryptions.

cAE => ciphertexts are *binding commitments*
RFC means fewer mistakes, better designs

• Anecdotally, misunderstanding is widespread
  • Practitioners and researchers alike make mistakes
• cAE can be tricky to build, many pitfalls
  • Checking multiple values in decryption
    => distinguishable failures possible!
• An RFC can dispel confusion and mandate good schemes

Some cAE constructions from research literature,
but many “knobs” to tweak w/r/t concrete choices

Need guidance on threat models, requirements, use cases!