Nested encryption as used for onion routing

The symmetric, low-latency counterpart of mixnets

\[ C_0 = \mathcal{E}_{K_1}(\mathcal{E}_{K_2}(\mathcal{E}_{K_3}(M))) \]

\[ M = B \ || \ M' \]

\[ C_1 = \mathcal{E}_{K_2}(\mathcal{E}_{K_3}(M)) \]

\[ C_2 = \mathcal{E}_{K_3}(M) \]

\[ C_3 = M \]
What **problem** does nested encryption supposedly solve?

\[ C_0 = E_{K_1}(E_{K_2}(E_{K_3}(M))) \]

\[ M = B \ || \ M' \]

\[ C_1 = E_{K_2}(E_{K_3}(M)) \]

\[ C_2 = E_{K_3}(M) \]

\[ C_3 = M \]
What **problem** does nested encryption supposedly solve?

Concrete, self-contained, understandable. Not building on UC [Canetti], [Camenisch, Lysyanskaya 2005]

A provable-security treatment of it

- Provide **syntax** and a **definition**
- Analyze **constructions**
  - Tor’s relay protocol: doesn’t satisfy our definition
  - LBE: does satisfy our definition

If the underlying blockcipher is a tweakable wideblock PRP
Indistinguishability

\[
\text{Adv}_S(A) = \Pr[A^{\text{Real}} \rightarrow 1] - \Pr[A^{\text{Ideal}} \rightarrow 1]
\]
Seeing our problem as a type of Authenticated Encryption (AE)

“Onion-AE”

Symmetric encryption that aims to achieve both privacy and authenticity
Seeing our problem as a type of Authenticated Encryption (AE)

“Onion-AE” Symmetric encryption that aims to achieve both privacy and authenticity

Lots of flavors of AE already:

- Probabilistic AE [Bellare, Rogaway 2000], [Katz, Yung 2000]
- Nonce-based AE [Rogaway, Bellare, Black, Krovetz 2001]
- Nonce-based AE with associated data (AEAD) [Rogaway 2002]
- Stateful AE [Bellare, Kohno, Namprempre 2004] ← Most closely related
- Misuse-Resistant AE [Rogaway, Shrimpton 2006]
- Release of Unverified Plaintext [Andreeva, Bogdanov, Luykx, Mennink, Mouha, Yasuda 2014]
- Robust AE [Hoang, Krovetz, Rogaway 2015]
- Online-AE [Hoang, Reyhanitabar, Rogaway, Vizár 2015]
A 3-tuple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where

- $\mathcal{K} : \mathbb{N} \rightarrow \mathcal{K}^*$ maps $n$ to $n+1$ strings
- $\mathcal{E} : \mathcal{K} \times \mathcal{M} \times \mathcal{U} \rightarrow \mathcal{C} \times \mathcal{U}$
- $\mathcal{D} : \mathcal{K} \times \mathcal{C} \times \mathcal{S} \rightarrow (\mathcal{M} \cup \mathcal{C} \cup \{\perp\}) \times \mathcal{S}$
(∀ n) (K₀, K₁, ..., Kₙ) ← ℋ(n); (K₀, K₁, ..., Kₙ) ← ℋ(n)
(∀ t) (M₁, ..., Mₜ) ← K; S₀, S₁, ..., Sₜ ← ε

for i ← 1 to t do
  (C₀, S₀) ← ℋ (Kᵢ, Mᵢ, S₀)

  for j ← 1 to n do (Cᵢ, Sᵢ) ← ℋ (Kᵢ, Cᵢ₋₁, Sᵢ)

assert Cₙ = Mᵢ

Correctness
Formalizing security

Oracle silencing:
behave like the utopian game shown unless the response you are about to give is fixed in every correct protocol.
In that case, answer ♦.

Idea explored in CRYPTO 2018 paper.
Silence an oracle response if, for the real game, given the transcript $t$ so far, the answer is fully determined by $\Pi \in \mathcal{C}$.
\textbf{Key}(n')

211 \textbf{if} \ n \neq \bot \ \textbf{then return} \ Err \\
212 \ n \leftarrow n' \\
213 \ (k_0, \ldots, k_n) \leftarrow K(n) \\
\textbf{Enc}(m) \\
221 \textbf{if} \ n = \bot \ \textbf{then return} \ Err \\
222 \ (c, u) \leftarrow E(k_0, m, u) \\
223 \textbf{return} \ c \\
\textbf{Dec}(c, i) \\
231 \textbf{if} \ n = \bot \ \textbf{then return} \ Err \\
232 \ (d, s_i) \leftarrow D(k_i, c, s_i) \\
233 \textbf{return} \ d

\textbf{Key}(n')

311 \textbf{if} \ n \neq \bot \ \textbf{then return} \ Err \\
312 \ n \leftarrow n' \\
\textbf{Enc}(m) \\
321 \textbf{if} \ n = \bot \ \textbf{then return} \ Err \\
322 \ c \leftarrow C \\
323 \textbf{return} \ c \\
\textbf{Dec}(c, i) \\
331 \textbf{if} \ n = \bot \ \textbf{then return} \ Err \\
332 \textbf{if} \ i = n \ \textbf{then} \ d \leftarrow \bot \\
333 \textbf{else} \ d \leftarrow C \\
334 \textbf{return} \ d
Without oracle silencing

Concurrent work

[Degabriele, Stam 2018]

Untagging Tor: A Formal Treatment of Onion Encryption
Without oracle silencing

Concurrent work
[Degabriele, Stam 2018]
Untagging Tor: A Formal Treatment of Onion Encryption
Limitations on this treatment of onion-AE

- Only attended to **outbound** messages
- No **corrupted** routers
- Fixed sequence of hops: **no “leaky pipe”**
- Authenticity checked only at **time of exit.**
  
  "Lazy authenticity"

Alternative: “**Eager authenticity**” might be preferred.
**Tagging attacks**

**Confirmation attacks** that a particular flow into an entry node leaves at some particular exit node, based on the **malleability** of the encryption

[Goldschlag, Reed, Syverson 1996]
[Dingledine, Mathewson, Syverson 2004]
[Fu, Ling 2009]  [Racoon23 2012]

\[ \mathcal{A} \] exploits malleability of encryption scheme to *tag* a ciphertext, e.g., xor’ing it with some constant \( \Delta \)

\[ \mathcal{A} \] detects the mauled ciphertext, confirming the originator of this flow.

Excluded because \[ \mathcal{AE} \implies \text{nonmalleability} \implies \text{no tagging attacks} \]
LBE is onion-AE secure

\approx Mathewson’s Proposal 202 (Design 1, Large Block Encryption), 2012.
Proposal 261 is 202 with AEZ

\[ C_0 = \mathbb{E}^{c_1\text{-hist}} \left( \mathbb{E}^{c_2\text{-hist}} \left( \mathbb{E}^{c_3\text{-hist}} (M || 0) \right) \right) \]

\[ \mathbb{E} \text{ a wideblock TBC, eg AEZ, EME2, Farfalle, HHFHFH} \]

**Theorem** [informal]: From an adversary \( \mathcal{A} \) that attacks LBE[\( \mathbb{E} \)] we construct an adversary \( \mathcal{B} \) that breaks \( \mathbb{E} \) as a PRP with comparable resources and advantage.
Final remarks

Two major definitional variants for onion-AE, **eager** and **lazy** authenticity. Both can be defined with oracle silencing. Which notion is desired?


Does any of this matter for Tor? I don’t know. But it’s best when we build our protocols out of primitives that achieve strong, formalized security definitions.