Untagging Tor:

A Tale of Onions, Raccoons, and Security Definitions

Jean Paul Degabriele

TECHNISCHE UNIVERSITÄT DARMSTADT

Martijn Stam

University of BRISTOL

Outline of this talk

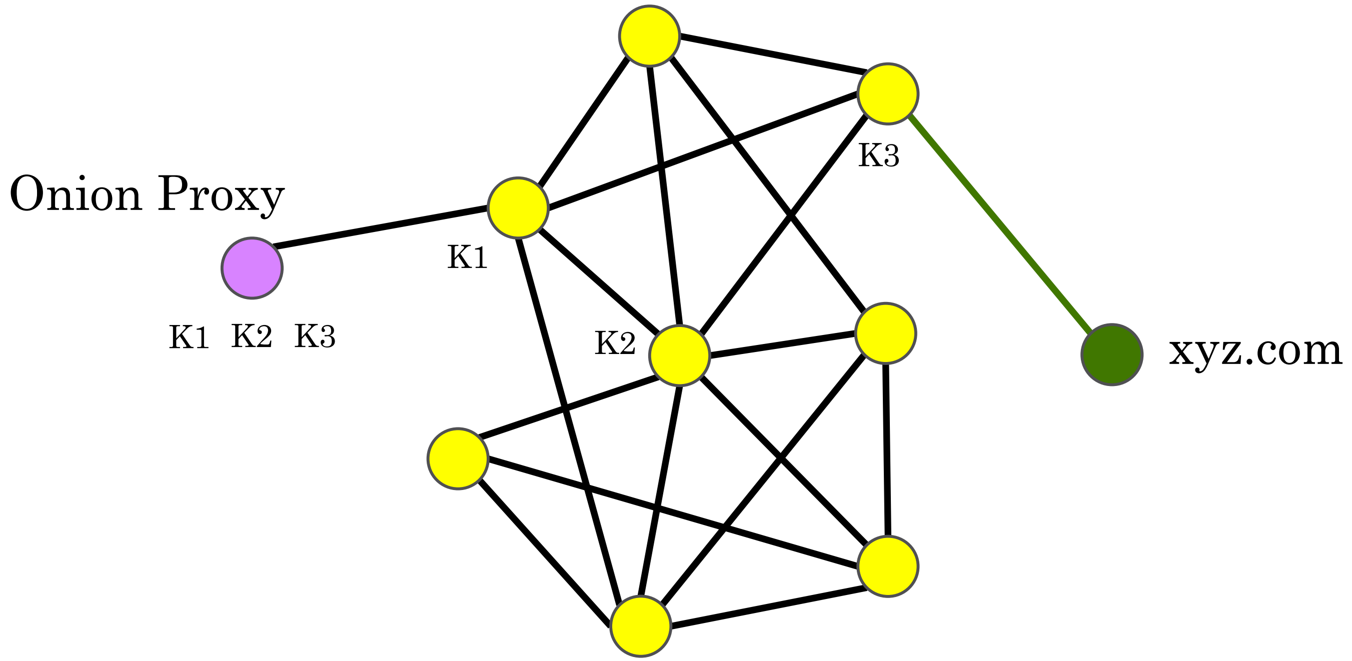
- Overview of Tor
- Tagging Attacks and Their Severity
- Tor Proposal 261
- Security Definitions and Analysis

Overview of Tor

Tor Overview

Four components:

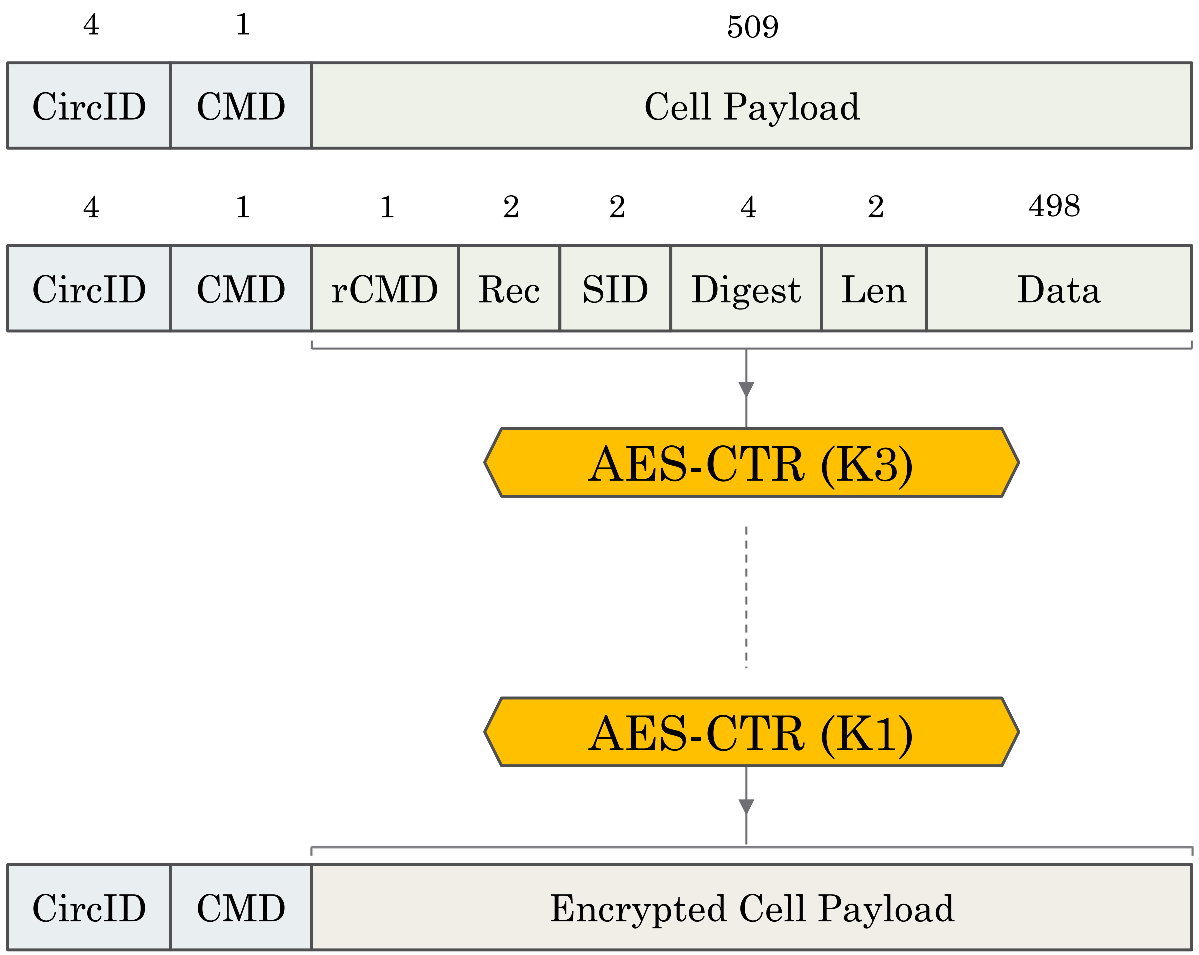
- Link protocol (TLS)
- Circuit Extend protocol
- Relay protocol
- Stream protocol



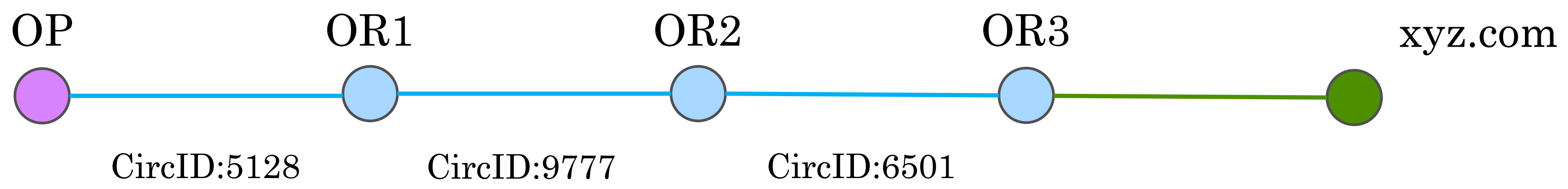
Tor Network
composed of Onion Routers

Relay Cell Format and Processing

- Cells are 514 bytes (v4+)
- **CircID**: Circuit Identifier
- **CMD**: Cell type - RELAY (3) or RELAY_EARLY (9)
- **Rec**: Recognised field (0x0000)
- **Digest**: seeded running hash (truncated SHA-1)



Relay Cell Forwarding

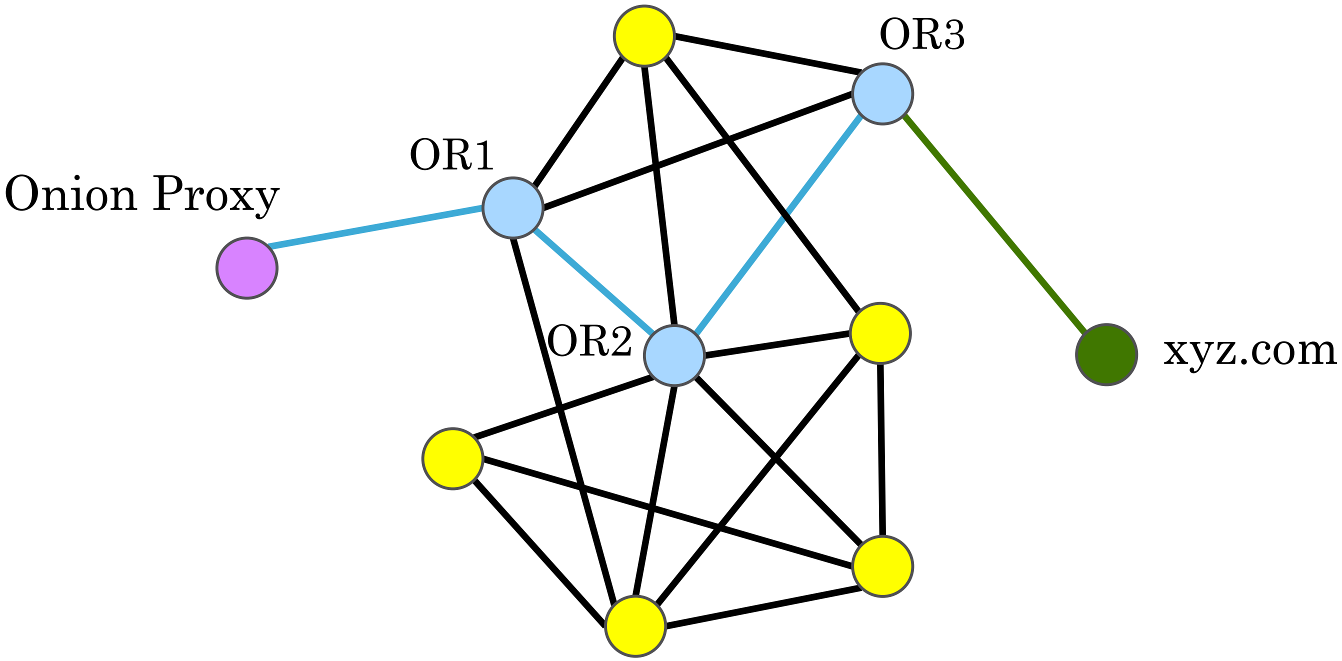


- Note that the same circuit is identified by a *different* **CircID** on each of its edges.
- Upon receiving a cell an OR performs the following:
  - Retrieves the state and key matching the cell's **CircID**.
  - Strips off one layer of encryption.
  - Checks if **Rec** = 0x0000 and the **Digest** verifies: if yes, the cell is recognised as being intended for that OR.
  - Otherwise it replaces the cell's **CircID** and forwards it to the next OR.

Tagging Attacks and Their Severity

Tagging Attacks

- Assume the adversary controls some onion routers.
- OR1 flips a bit in a cell and forwards it over.
- OR3 flips that bit back and tests if decryption succeeds.
- If yes, the adversary has confirmed that the two edges (CircIDs) belong to the same circuit.



- Note the similarity with **traffic correlation attacks**, where roughly the same effect is achieved by matching **traffic patterns** between input and output edges.

The Perceived Severity of Tagging Attacks Over The Years

- 2004 • Tagging attacks were known to the Tor designers, but protecting against them was deemed pointless since traffic correlation attacks would be possible anyway.
- 2008 • **The23rd Raccoon**: *How I Learned to Stop Ph34ring NSA and Love the Base Rate Fallacy*.
- 2009 • Tagging attacks rediscovered by Fu and Ling and presented at Black Hat 2009 – Tor project's response: *Nothing new here!*
- 2012 • **The23rd Raccoon**: *Analysis of the Relative Severity of Tagging Attacks*.
  • Tor project decides to revise the relay protocol and protect against tagging attacks.

The23rd Raccoon's Observations

- Consider a network with 10,000 concurrent circuits, and a TC adversary controlling 30% of the entry/exit nodes.
- Due to noise, correlation detectors inevitably exhibit false positives. Let us assume a false positive rate of 0.5%.
- The probability that a pair of edges truly belong to the same circuit when a match is detected is ~2% (*base rate fallacy*).
- This effect becomes more pronounced as the number of circuits increases, but **tagging attacks are immune** to this.
- The 2012 post describes an **amplification effect** and argues that tagging attacks require less resources.

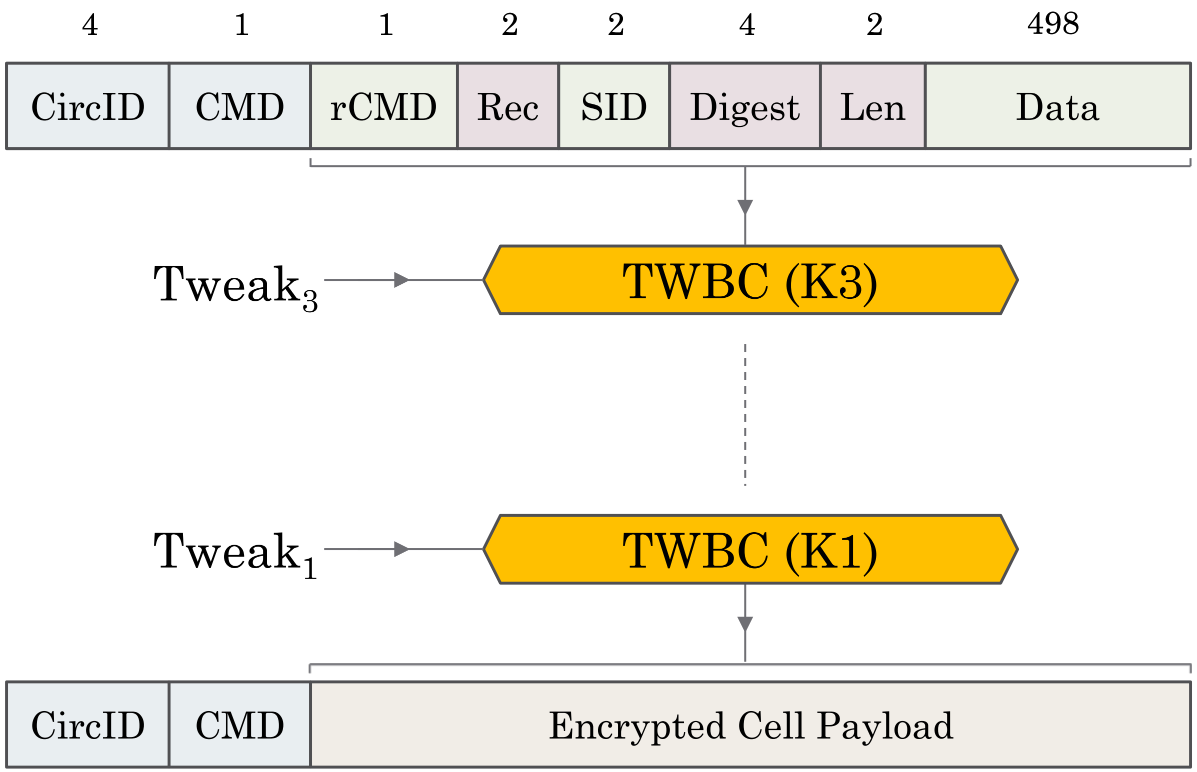
Tor Proposal 261

Thwarting Tagging Attacks

- Tagging attacks are enabled by the **malleability** of **counter mode** encryption employed in Tor.
- A naïve fix would be to append a MAC tag at each layer of encryption, but **this leaks information**!
- This leakage can be prevented with appropriate **padding** to ensure the **cell size is constant** throughout.
- An alternative approach, resulting in a higher throughput, is to use a **tweakable wide-block cipher**.
- Possible instantiations include AEZ, HHFHFH, and Farfalle.

Relay Cell Processing in Prop 261

- **Digest:** now set to 0x00000000.
- AES-CTR replaced by TWBC.
- Each layer maintains a separate tweak, updated with each cell.
- **CMD** is included in each tweak (RELAY or RELAY_EARLY).
- End-to-end integrity via **encode-then-encipher**.
- Verify zeros in **Rec**, **Digest**, and **Len** (7 msb) – total 55 bits.



Security Definitions and Analysis

Prior Works on Onion Encryption

- **[CL05]** Introduced a UC security definition for onion encryption.
- However, their notion is tailored for the **mix-net** setting where: cells are *routed individually* (no circuits), onion routers are *stateless*, and the onion encryption is *public-key*.
- **[BGKM12]** Introduced a UC security definition intended for Tor's use case, covering both circuit establishment and onion encryption.
- Their definition has a number of shortcomings, but the most prominent is that it **does not protect against tagging attacks**.
- Indeed this vulnerability was turned into a feature – referred therein as **predictable malleability**.

What Does Onion Encryption Contribute?

- It is natural to expect **confidentiality**, **integrity**, protection against **replay** and **reordering** of cells, etc.
- The main goal of Tor is anonymity, but this is achieved through a combination of **cryptographic mechanisms** and other factors such as **network size** and **traffic load**.
- Our goal is to identify what security can the **cryptographic component** contribute towards anonymity, assuming **other factors to be ideal**.
- We contend that the answer is **Circuit Hiding**.

Intuition Behind Circuit Hiding

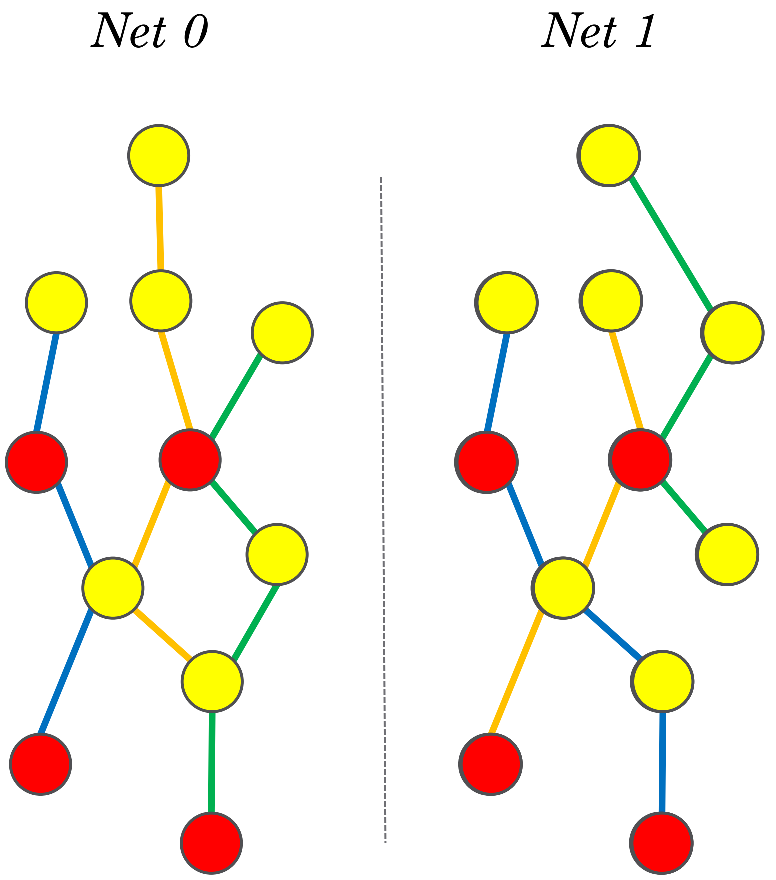
An adversary should not be able to learn any ***new information about the circuits' topology*** *in the network beyond what is* ***inevitably leaked through node corruptions****.*

This should hold even when the adversary can ***choose the messages that get encrypted*** *and is able to* ***reorder****,* ***inject****, and* ***manipulate cells*** *on the network.*

- Note how tagging attacks fit in this broader class of attacks.

Circuit Hiding (Simplified)

- Adversary specifies a **set of nodes** and indicates the **subset that it controls**.
- It specifies **two networks** (sets of circuits).
- The **interface with the corrupted nodes** must be the **same in both networks**.
- A **network** is chosen at **random** and the adversary gets to **interact** with it **via** the **corrupted nodes** and tries to **determine which** network it is.
- This is the main idea, the **actual definition** is significantly **more complex**.



The Security of Proposal 261

- It turns out that Proposal 261 **is not** circuit hiding!
- The reason is that the cell header's **CMD** field can be used to tag cells by switching its value from RELAY to RELAY_EARLY.
- A similar vulnerability was exploited in the **2014 CMU incident** on Tor's Onion Services which took down Silk Road.
- Recall that **CMD** was included in the wide-block cipher's tweak but, while it helps, it does not prevent the attack.

The Security of Proposal 261

- In practice, however, there are a number of factors that limit the exploitability and efficacy of this attack.
- The RELAY_EARLY cell type is needed in Tor's mechanism for limiting the maximum circuit size.
- It may make sense in practice to accept this issue and rely on the other mitigating factors rather than eliminate it completely.
- We prove that **a variant of Prop 261**, where **CMD** is fixed to RELAY, **is circuit hiding**, showing that the **overall design is sound and effective against tagging attacks**.

Concluding Remarks

Concluding Remarks

- For more details, look out on **eprint.iacr.org** for our paper: ***Untagging Tor: A Formal Treatment of Onion Encryption.***

- Plenty more work to be done on the formal analysis of Tor - e.g. **Circuit Extend** protocol.

- More work is needed to better understand **The23rd Raccoon's** observations and validate them empirically.