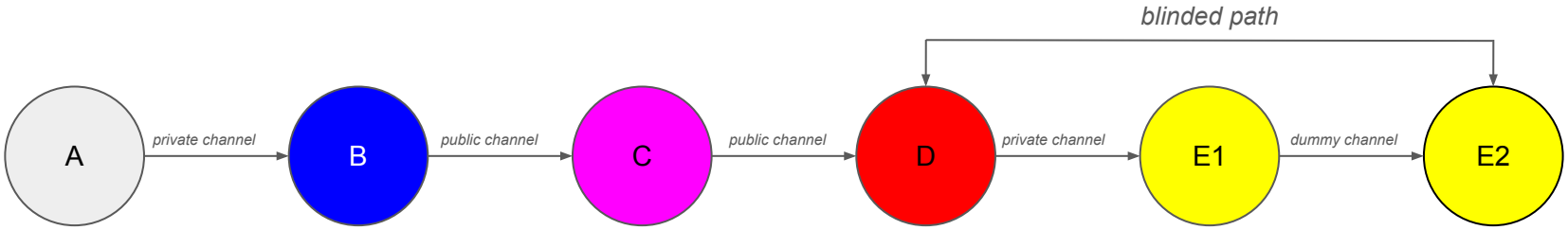
Trampoline ❤️ Bolt 12

Sample payment route

A is a wallet user connected to its LSP B.
E is a wallet user connected to its LSP D.
C is a standard routing node.
E advertises a two-hops blinded path starting at D with a dummy hop.
Note that a real two-hops blinded path would work the same for the mechanisms described in this document.



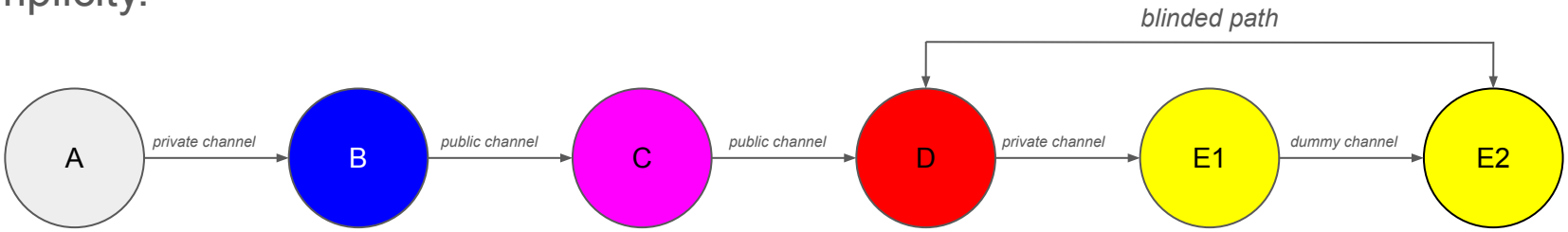
Sample payment route

We use colors to highlight who can decrypt which piece of data and the 😎 emoji when data is encrypted for the *blinded* node, for example:

payload_B: [...] → can be decrypted with B's private key

encrypted_data_D: [...] 😎 → can be decrypted with D's private key **and** a blinding path key

We use a flat fee of 1000 sat and expiry delta of 25 blocks at every hop for simplicity.

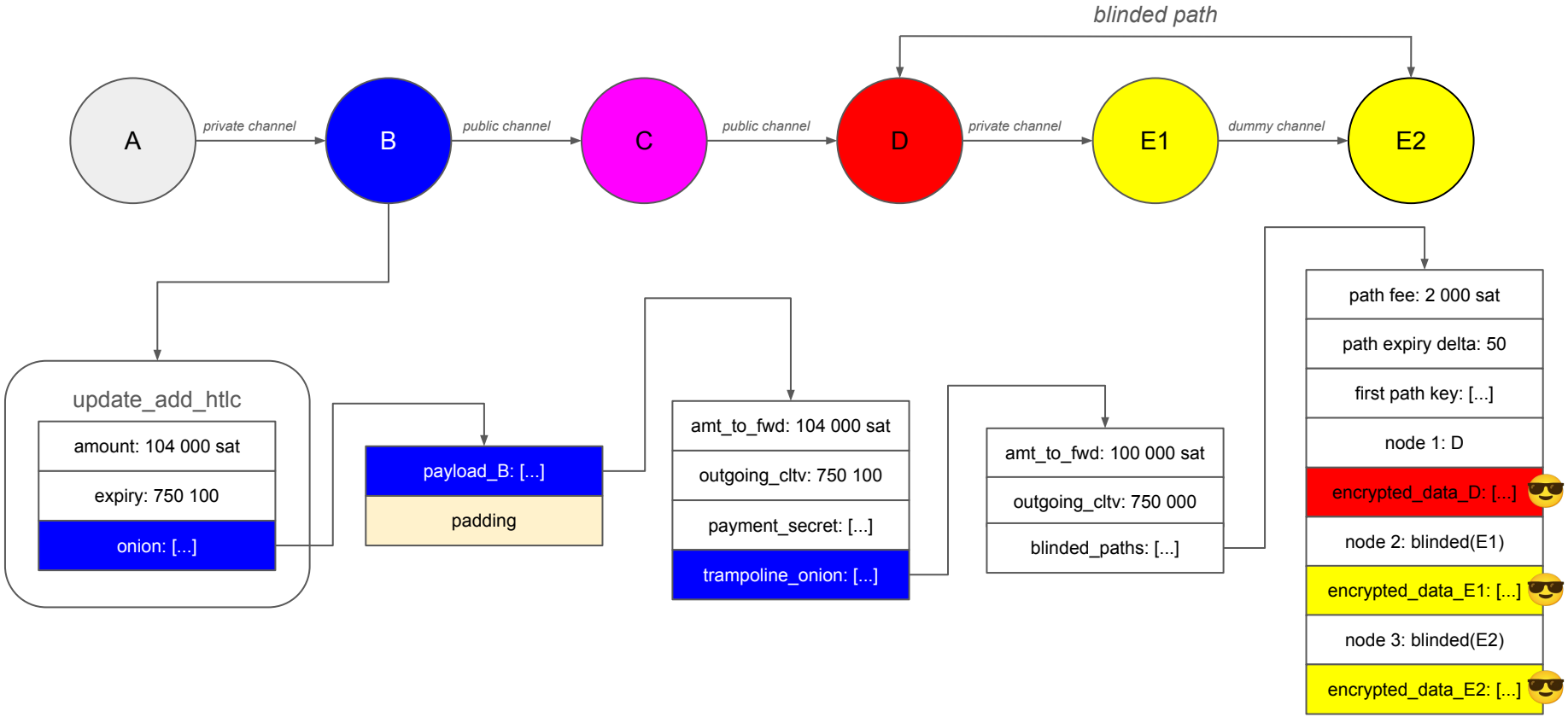


Scenario #1: recipient does *not* support trampoline

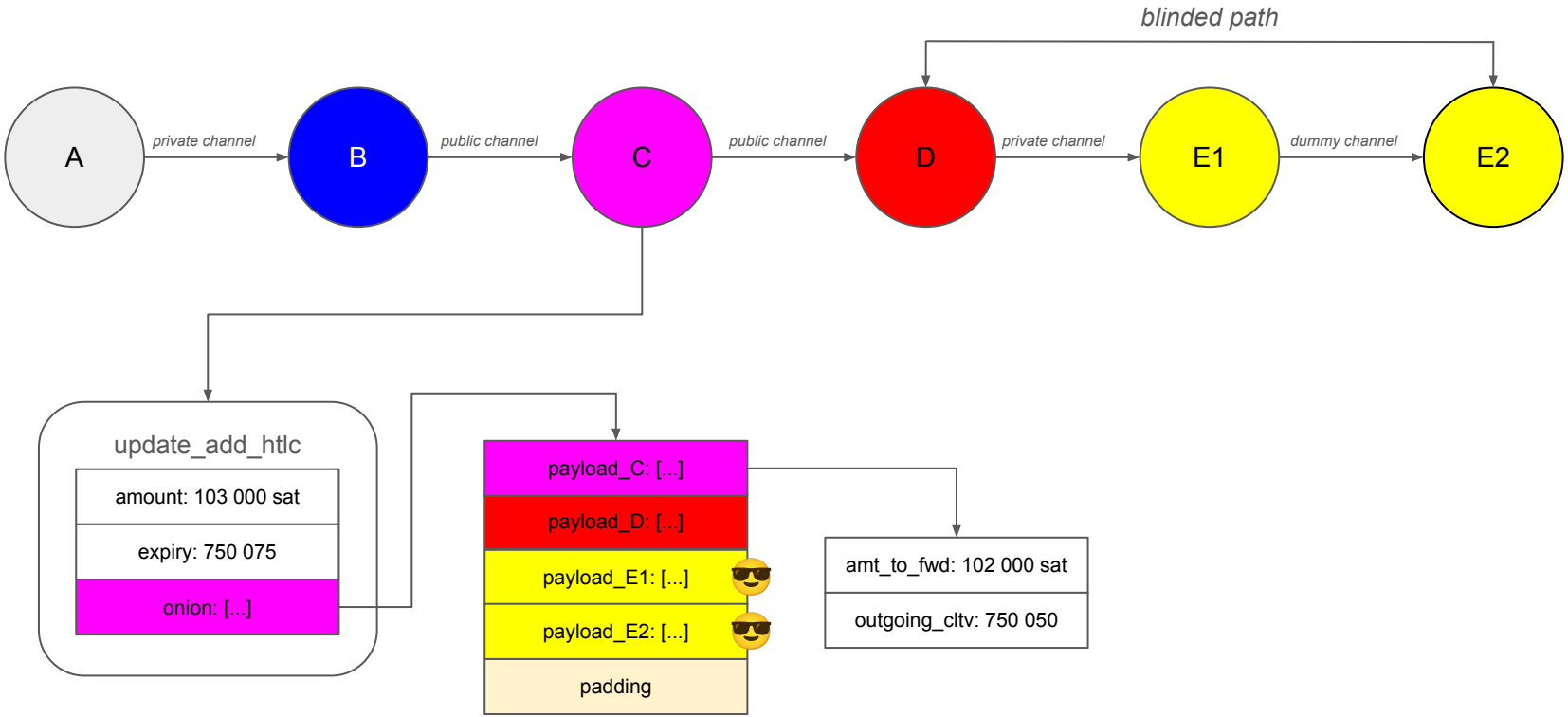
High-level design:

- We include the blinded path in the trampoline onion for the last trampoline node
- This reveals the introduction node to the last trampoline node, which isn’t great if the recipient doesn’t try to protect its identity
- But the sender can use multiple trampoline hops to ensure that the first trampoline node doesn’t learn anything about the recipient
- In the example below, we use a single trampoline hop, but it’s trivial to add another one

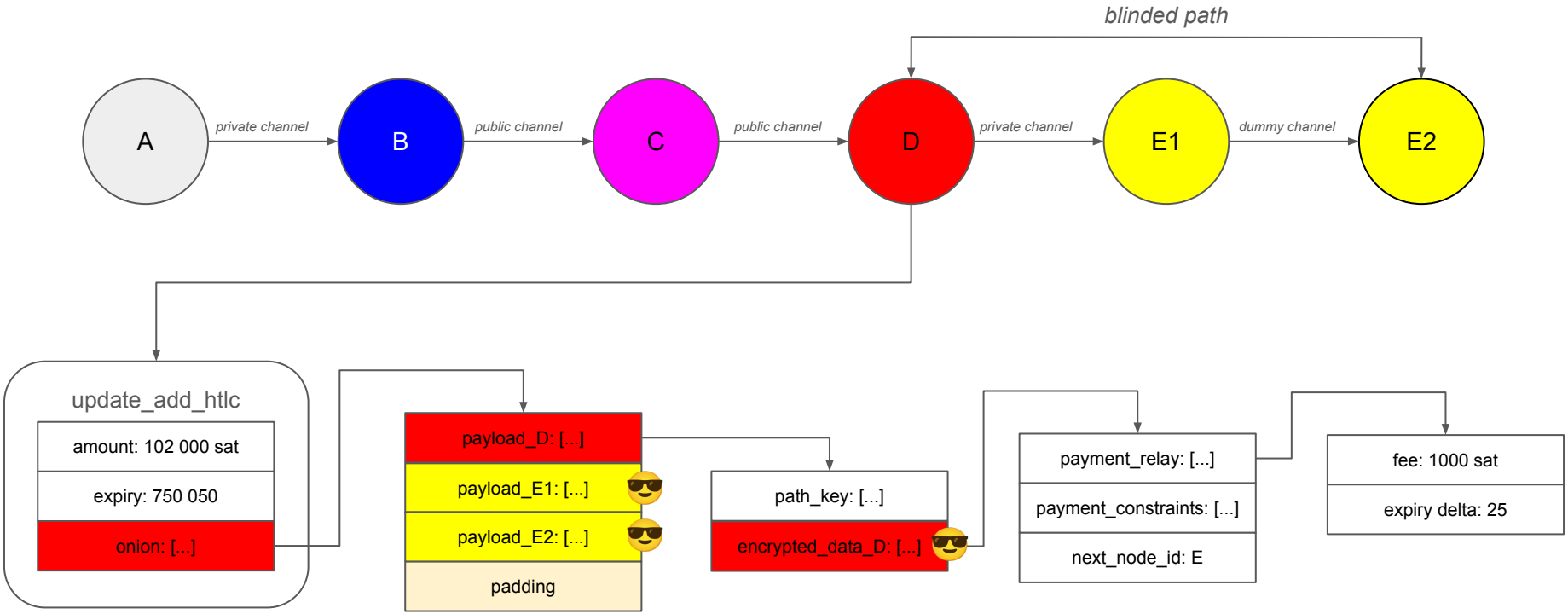
Scenario #1: HTLC received by B (relay to blinded path)



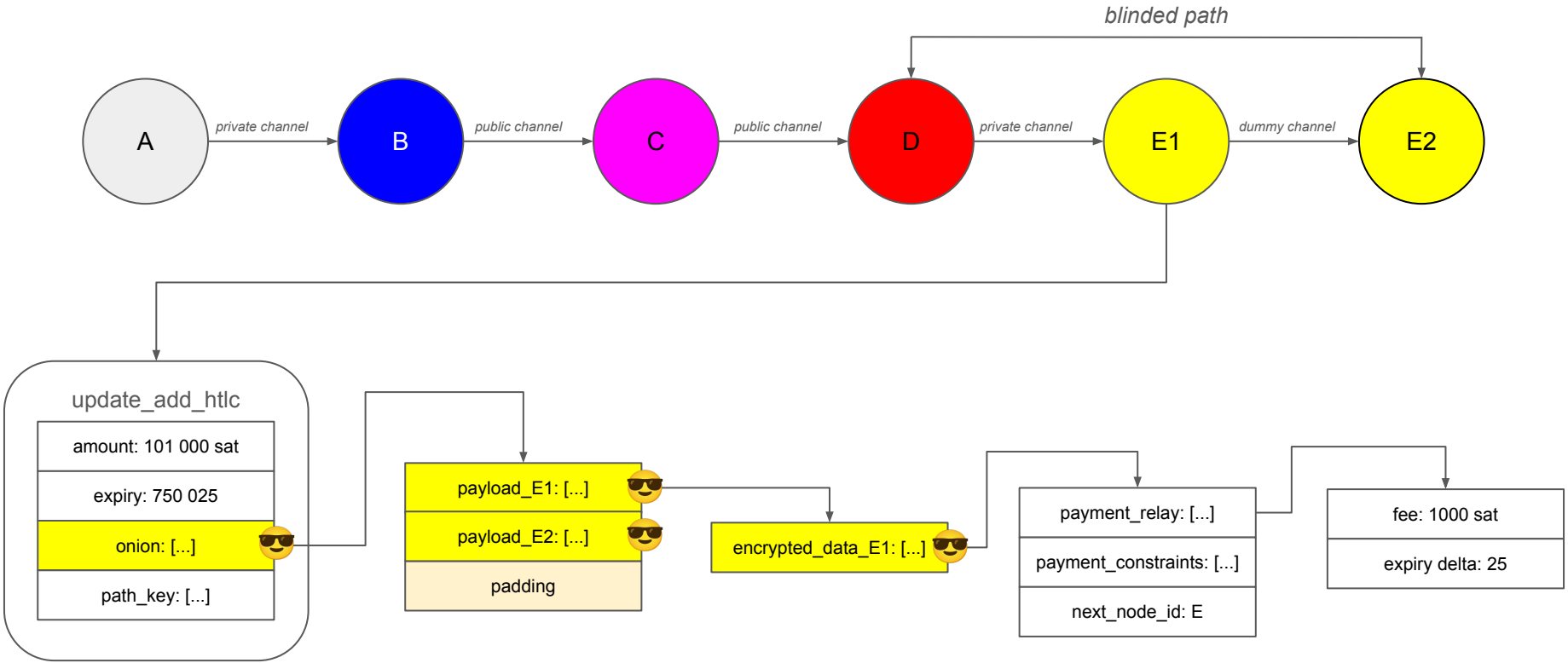
Scenario #1: HTLC received by C (standard channel relay)



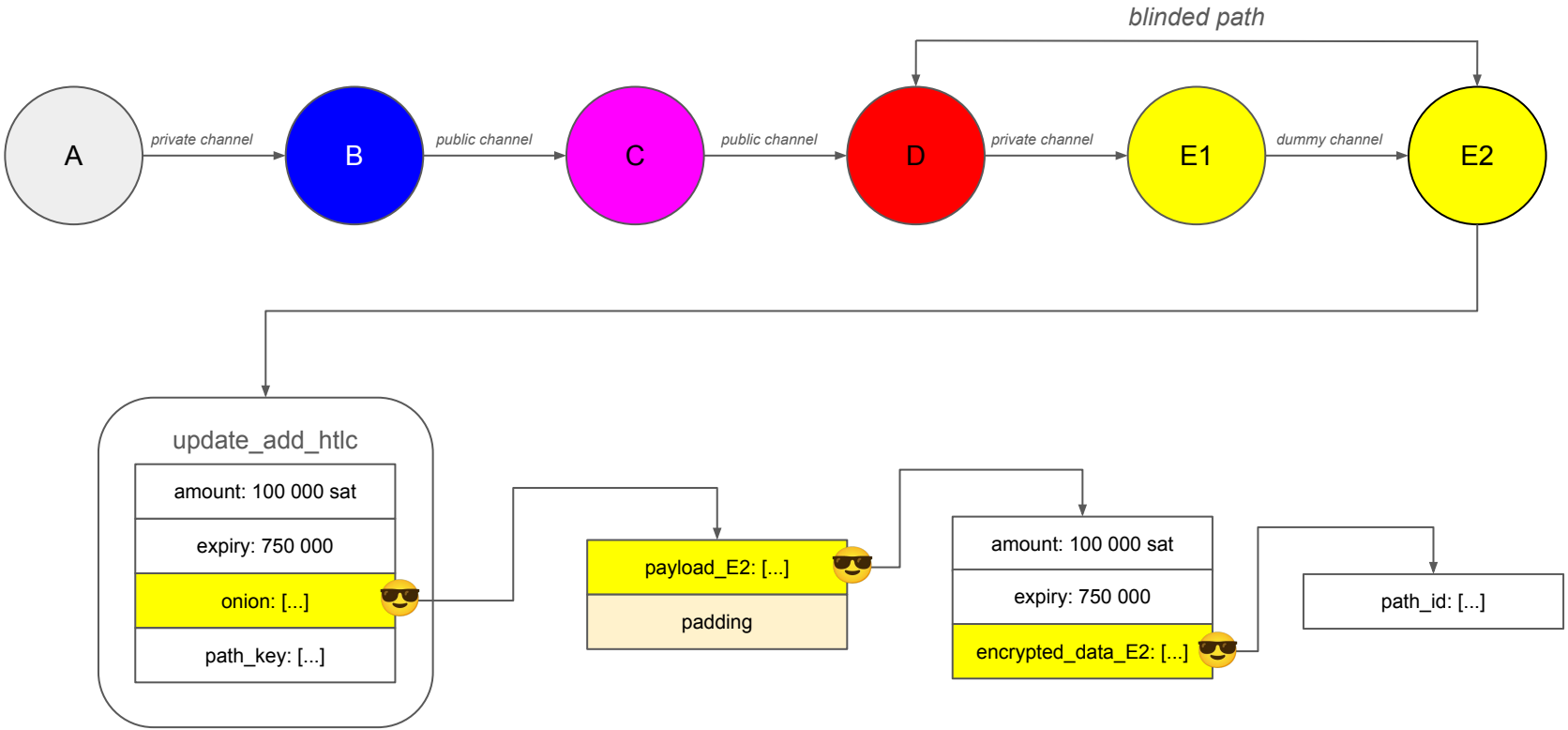
Scenario #1: HTLC received by D (introduction node)



Scenario #1: HTLC received by E1 (inside blinded path)



Scenario #1: HTLC received by E2 (inside blinded path)

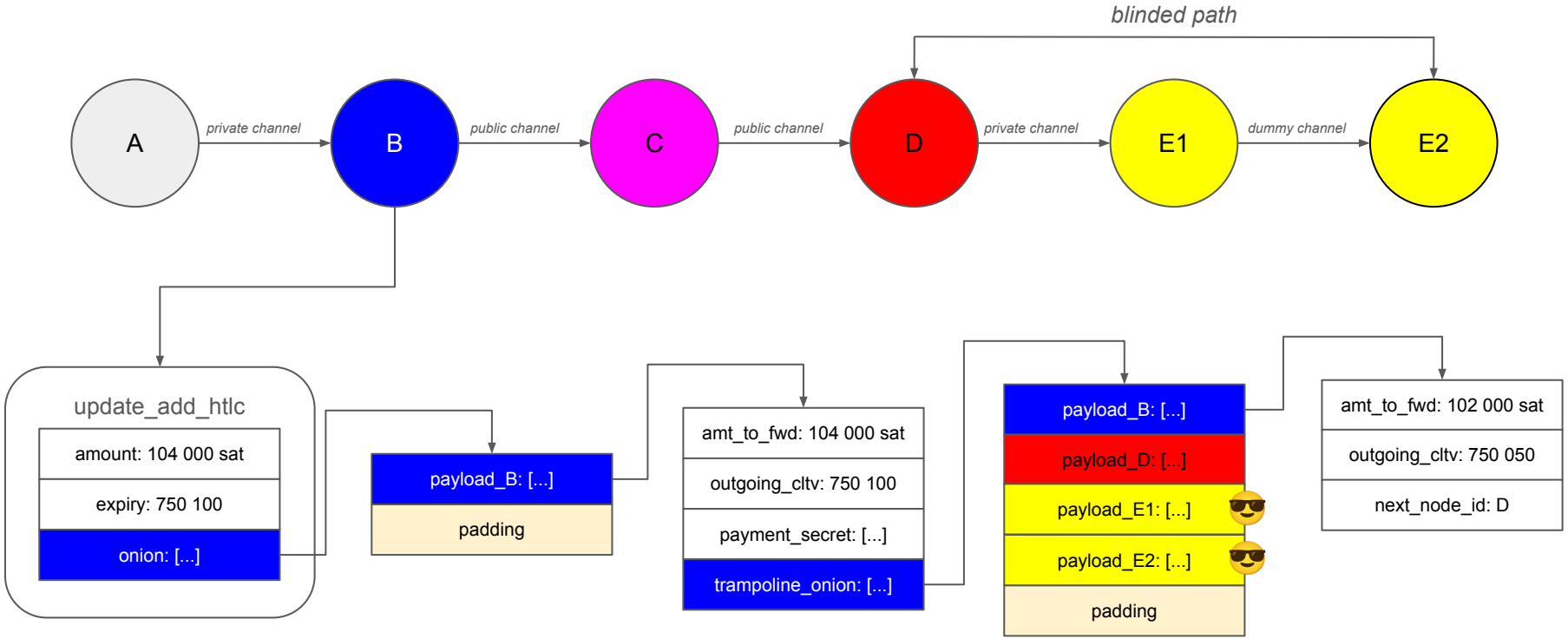


Scenario #2: recipient supports trampoline

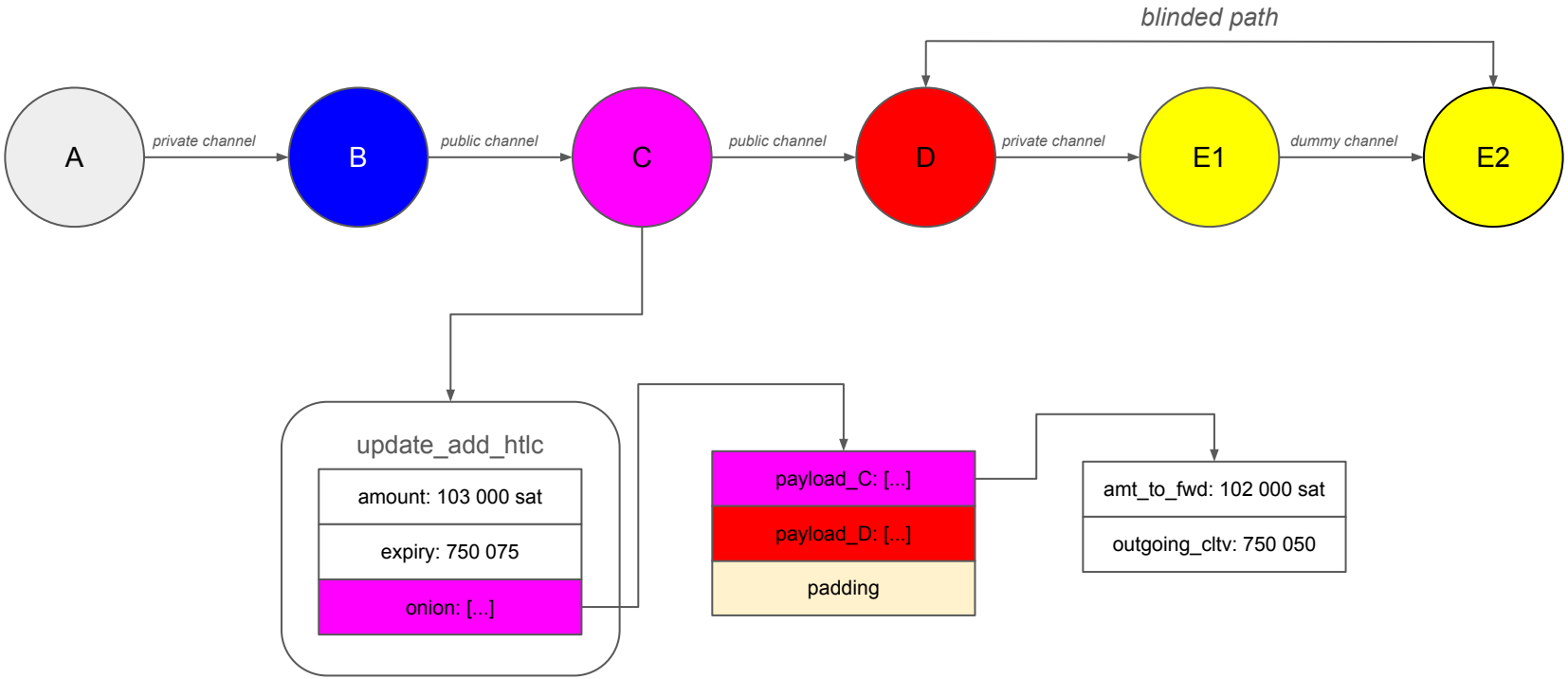
High-level design:

- We assume that all nodes inside the blinded path support trampoline
- We create a trampoline onion that uses each blinded hop as a trampoline hop
- We add one (or more) trampoline hop(s) before the blinded path
- We include the blinded path's encrypted data payloads in the trampoline onion payload for each blinded node
- This lets the sender include additional TLVs in the trampoline onion for the recipient (e.g. keysend)
- The first trampoline node does not learn the blinded path's details

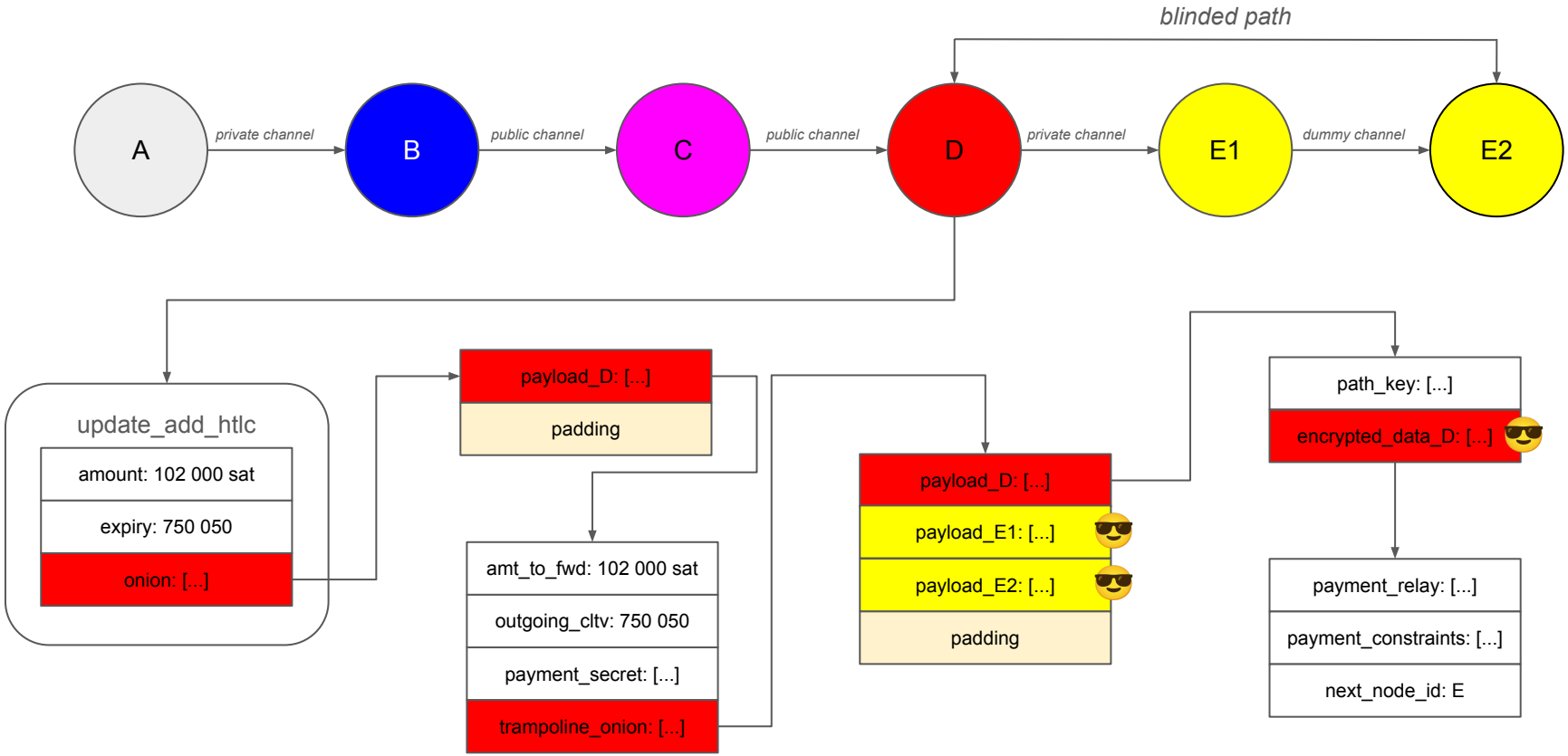
Scenario #2: HTLC received by B (relay to trampoline)



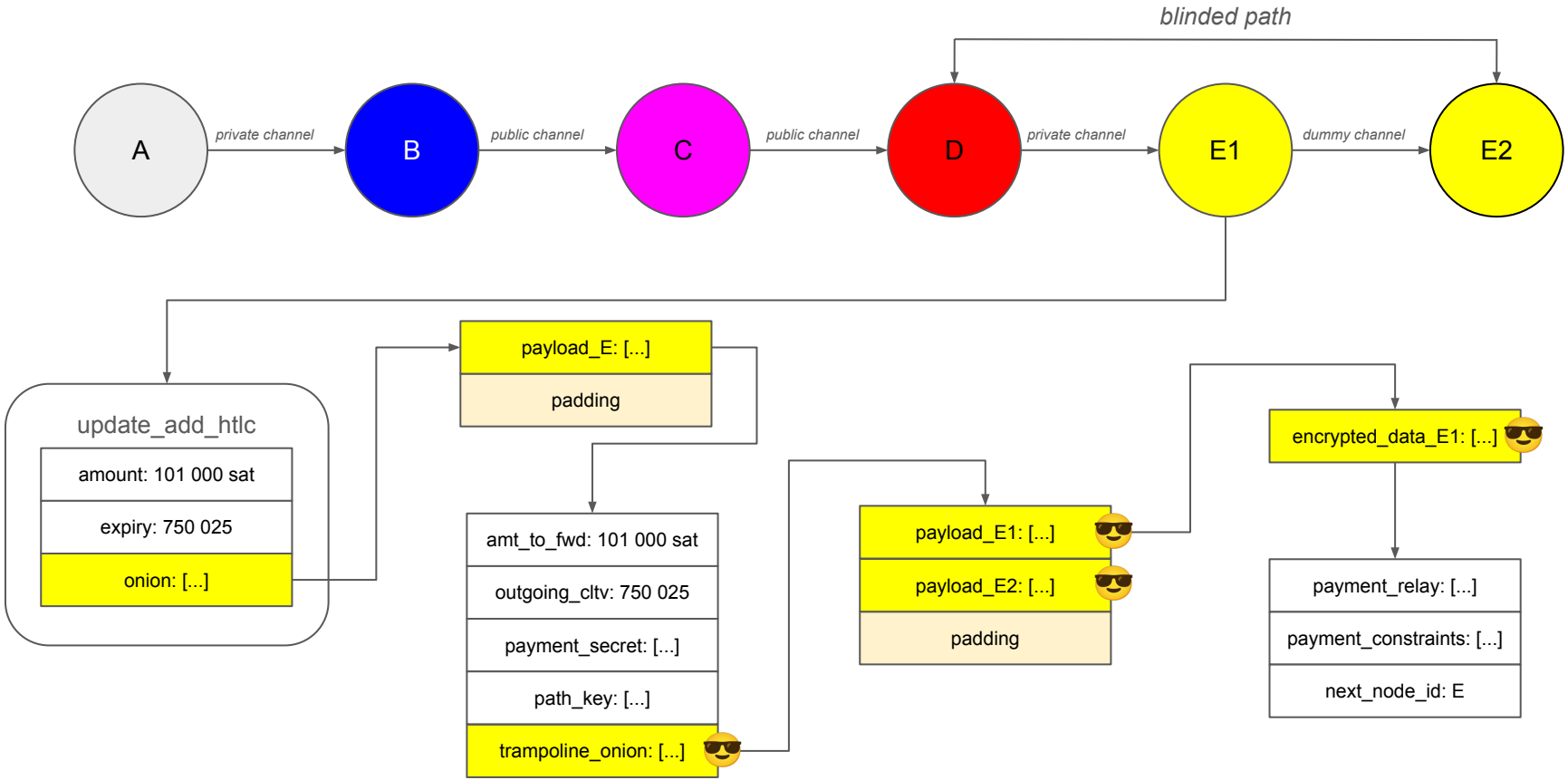
Scenario #2: HTLC received by C (standard channel relay)



Scenario #2: HTLC received by D (blinded trampoline intro)



Scenario #2: HTLC received by E1 (blinded trampoline)



Scenario #2: HTLC received by E2 (blinded trampoline)