Onion Routing with Replies

Christiane Kuhn
Karlsruhe Institute of Technology

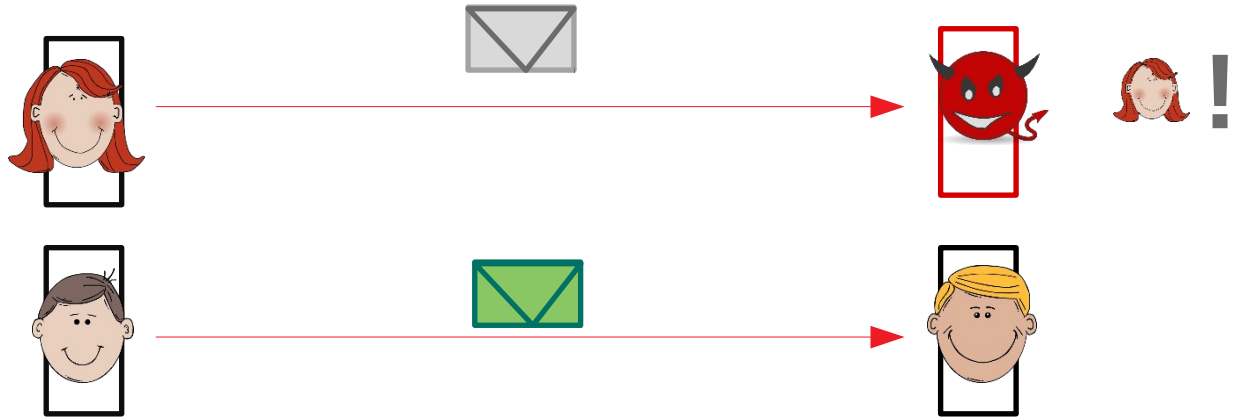
Dennis Hofheinz
ETH Zuerich

Andy Rupp
Université du Luxembourg

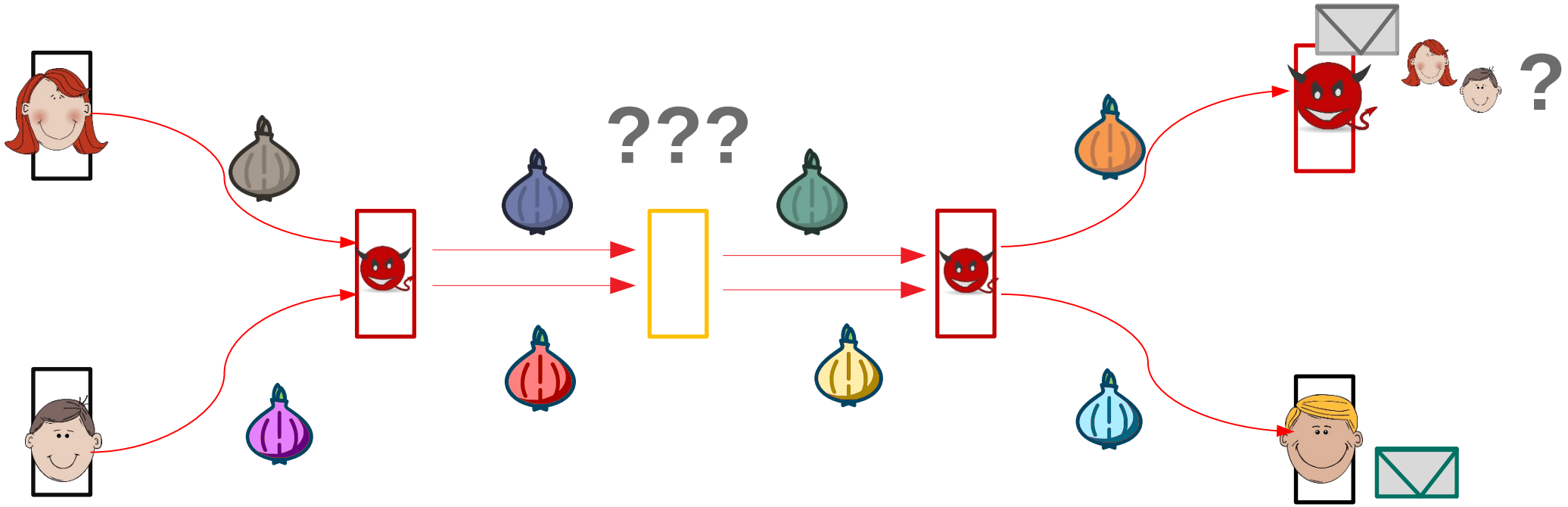
Thorsten Strufe
Karlsruhe Institute of Technology

KASTEL Research Laboratories

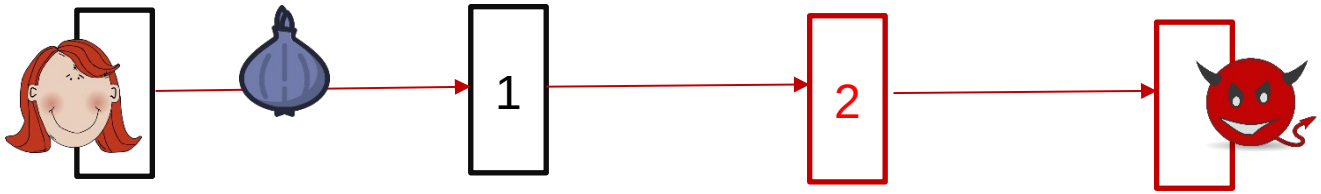
Goal: Relationship Privacy

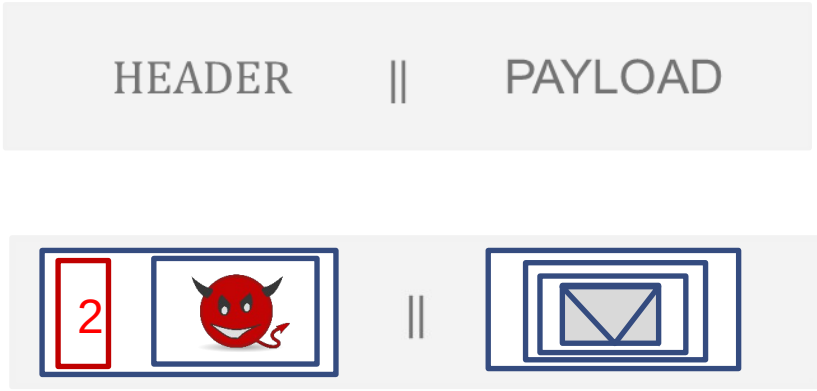


Onion Routing



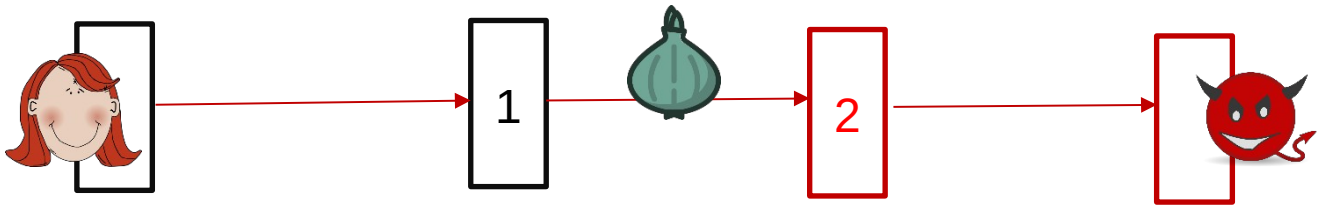
Header and Payload

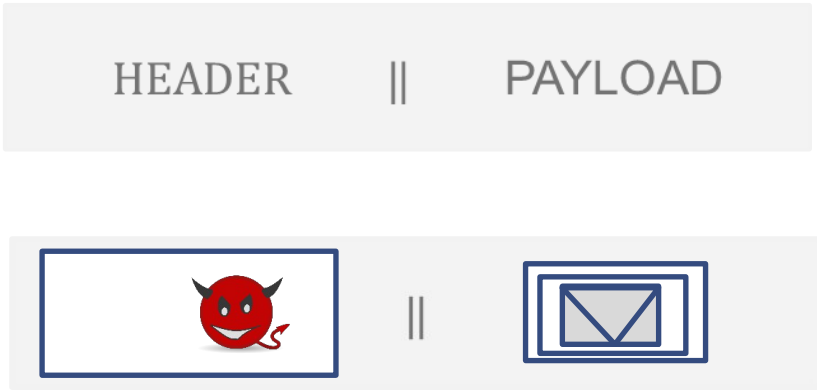




G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In IEEE S&P, 2009

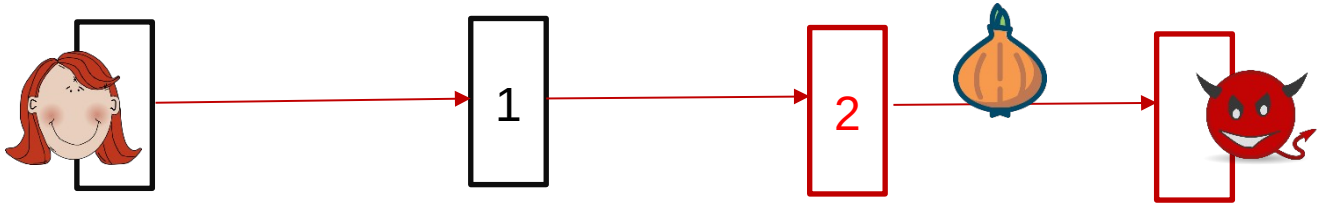
Header and Payload





G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In IEEE S&P, 2009

Header and Payload

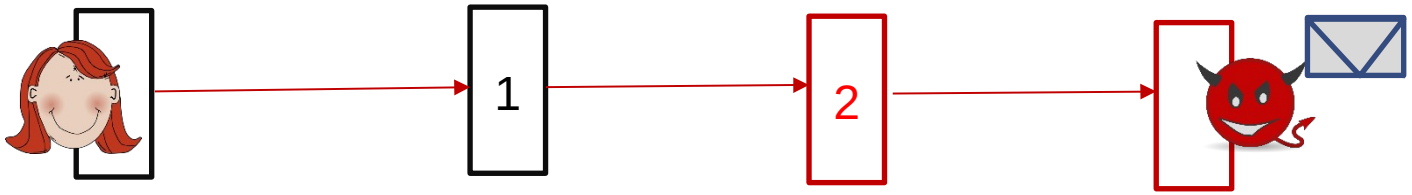


HEADER || PAYLOAD

|| 

G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In IEEE S&P, 2009

Header and Payload

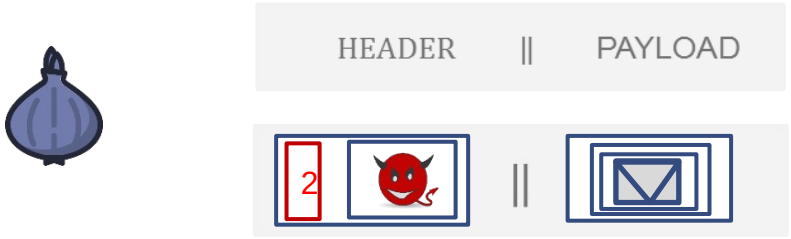


HEADER || PAYLOAD

|| 

G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In IEEE S&P, 2009

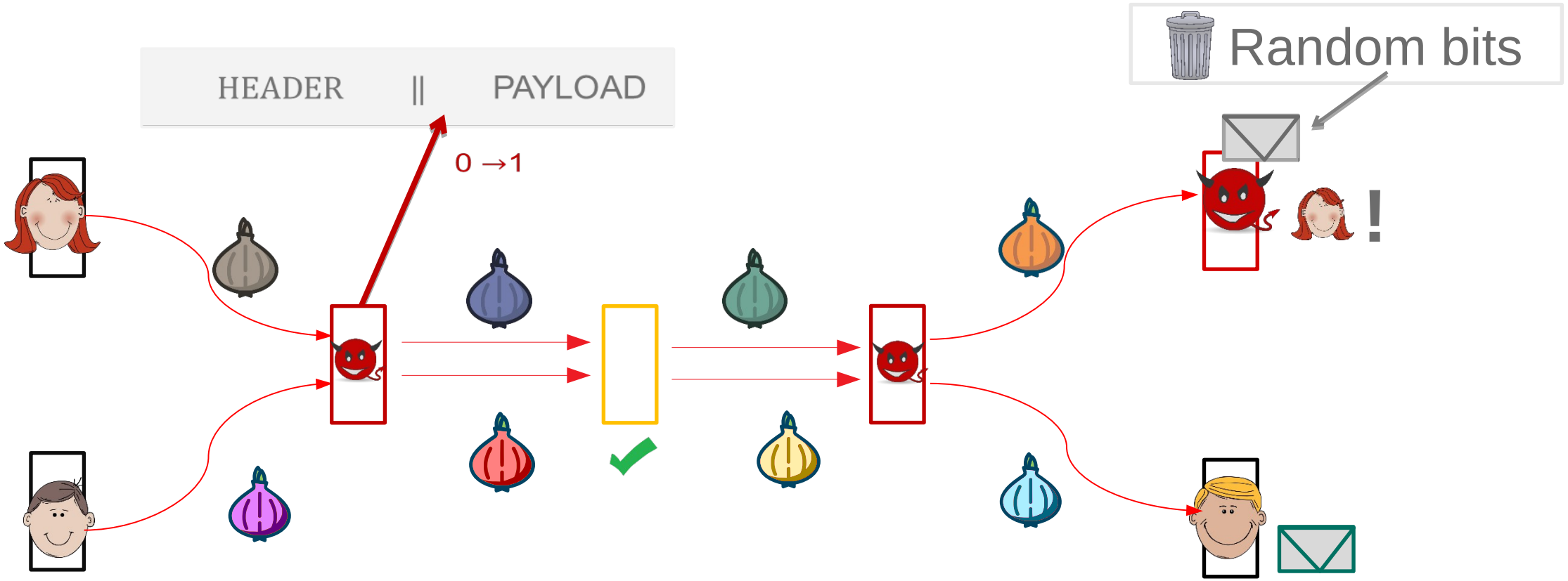
Header and Payload



Integrity

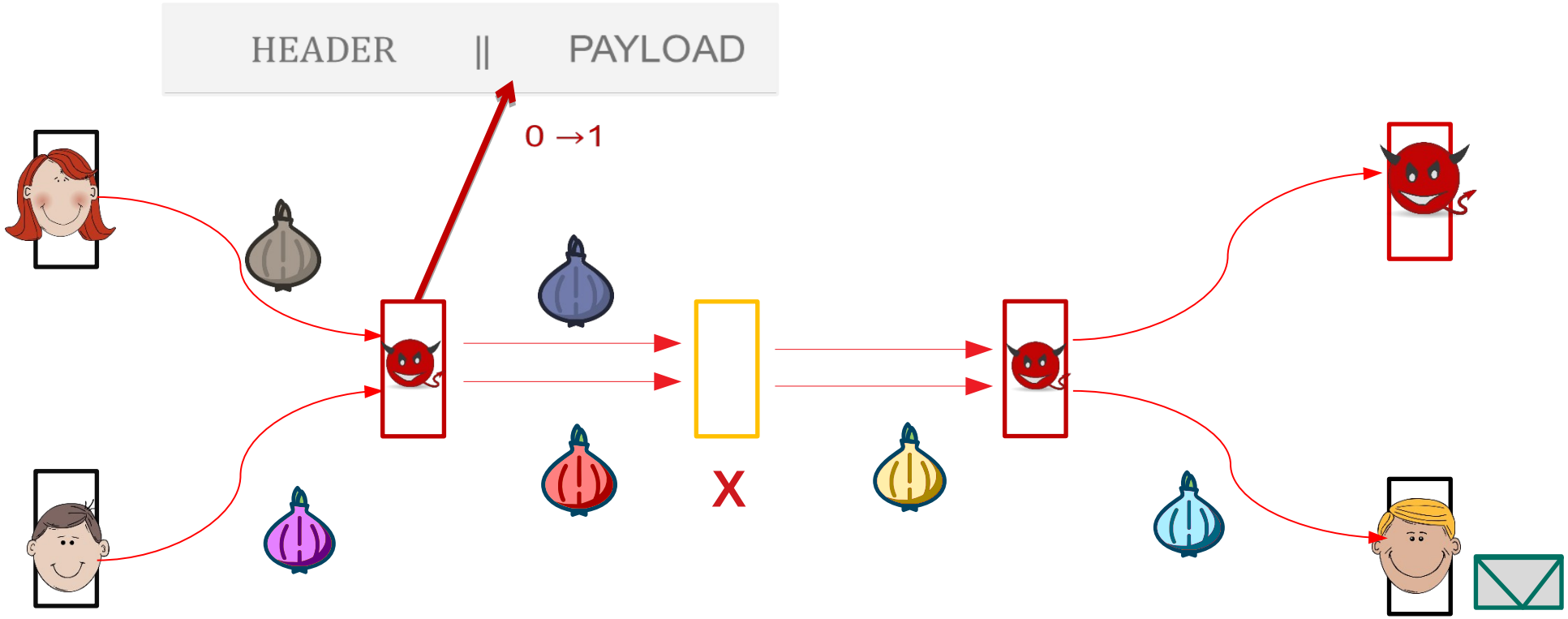
G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In IEEE S&P, 2009

Malleability Attack



C. Kuhn, M. Beck, and T. Strufe. Breaking and (partially) fixing provably secure onion routing. In IEEE S&P, 2020

Preventing the Malleability Attack

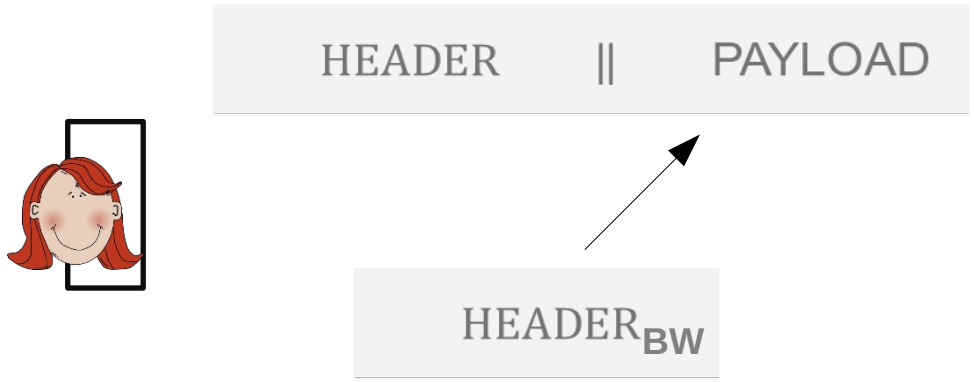


Add explicit payload authentication

C. Kuhn, M. Beck, and T. Strufe. Breaking and (partially) fixing provably secure onion routing. In IEEE S&P, 2020

Reply support

Repliable Onion:



Replying:

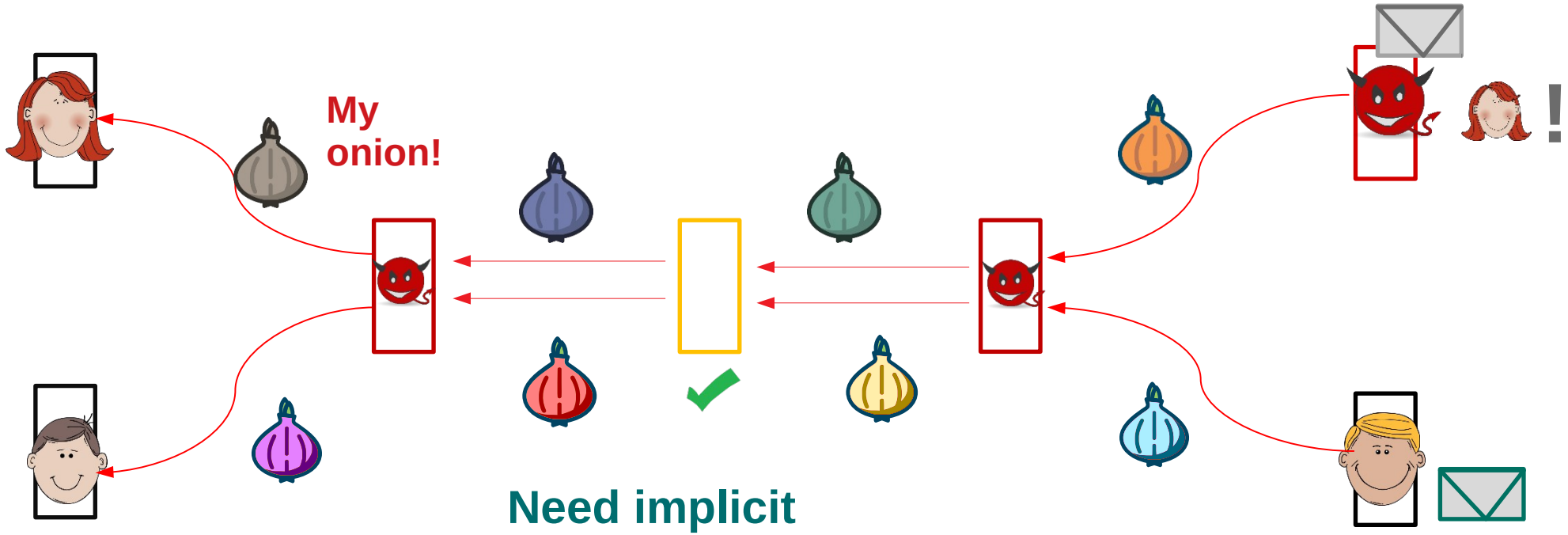
G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In IEEE S&P, 2009

… but secure replies are difficult

… authentication for replies

Sender: cannot precalculate

Receiver: cannot be trusted to know parts of other onion layers



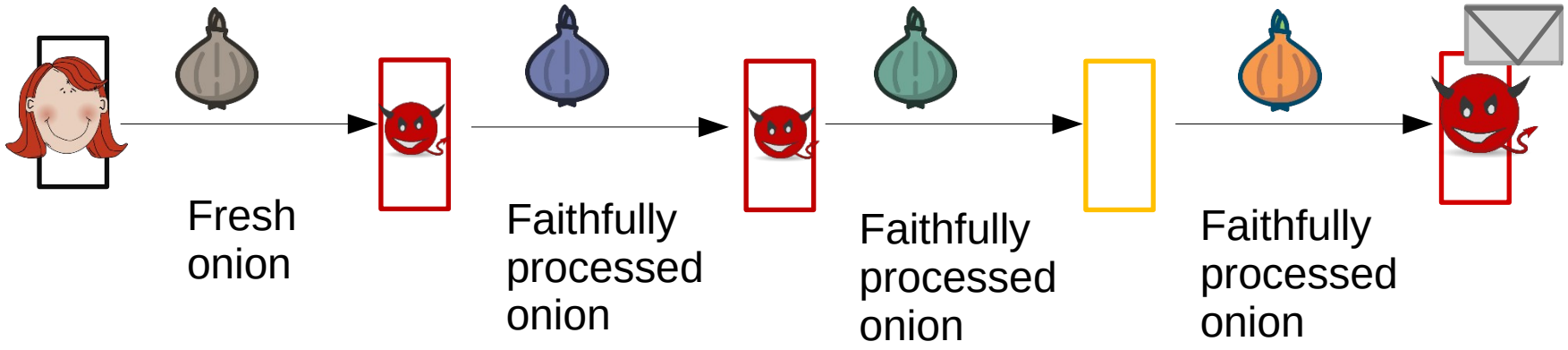
Need implicit payload authentication!

Secure, repliable

Onion Routing Protocols

Protocol 1: SNARGs

Idea: Prove you did it right!

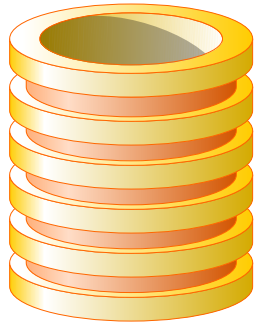


Authentication chain!

Protocol 2: Updatable Encryption

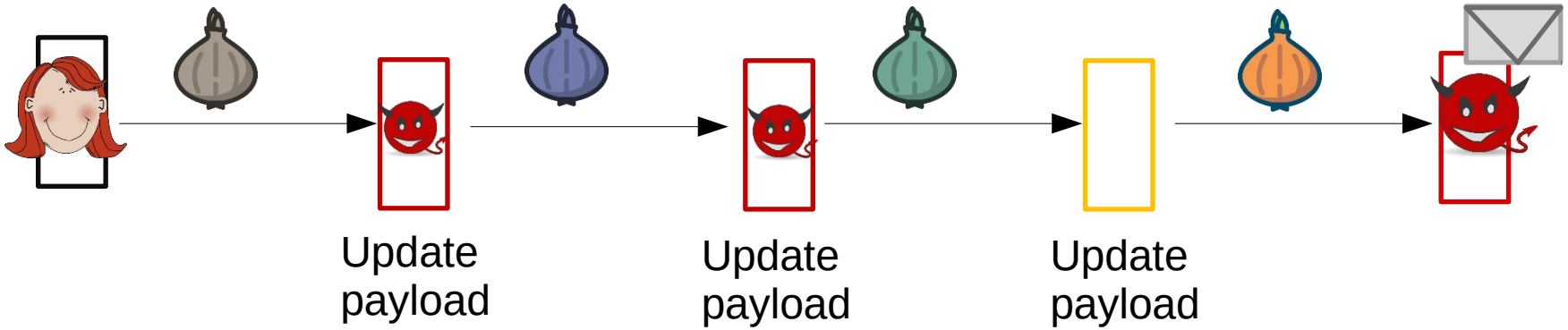
Idea: Updatable Encryption

Update stored ciphertexts from old to new key
With update token

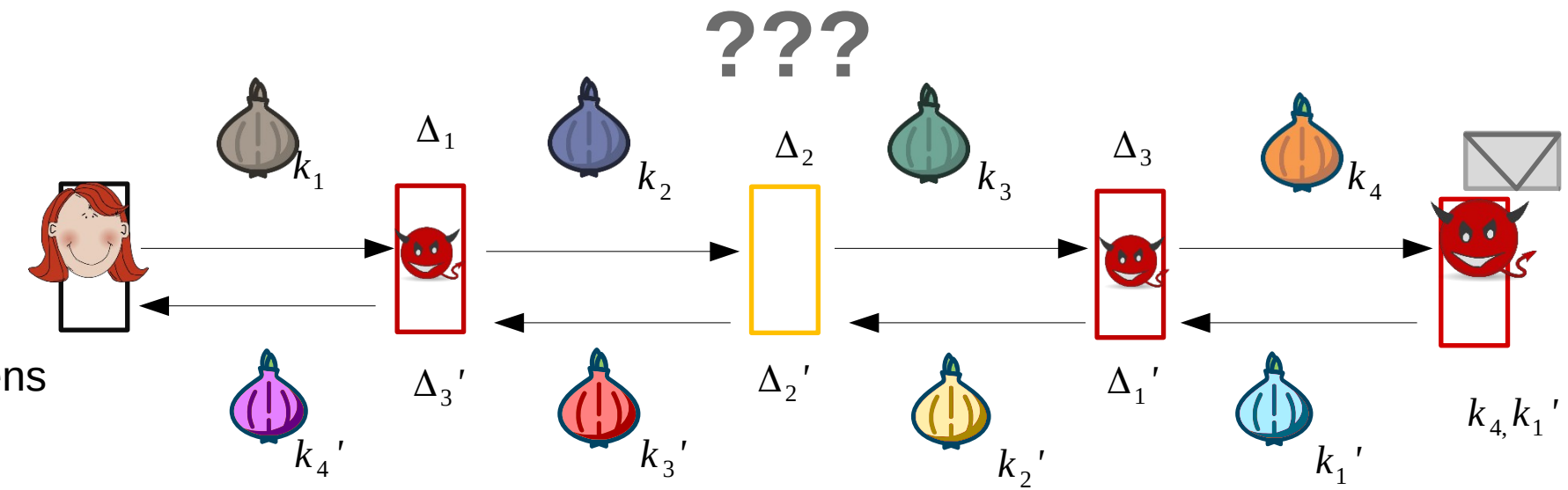


$k_{old} \xrightarrow{\Delta} k_{new}$

Provides **plaintext integrity**



Protocol 2: Updatable Encryption



Calculates all keys and tokens

$k_1, \ldots, k_4$

$k_1', \ldots, k_4'$

$\Delta_1, \ldots, \Delta_3$

$\Delta_1', \ldots, \Delta_3'$

Modified? Plaintext Integrity!

Reusable Security Properties

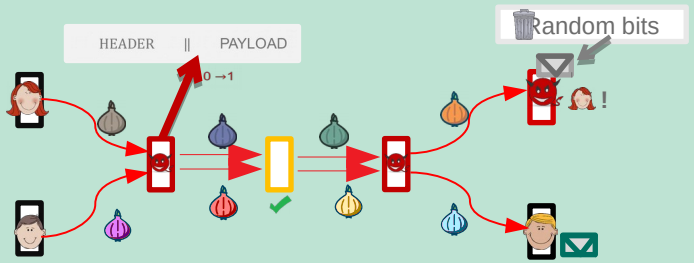
Similar to earlier [1,2] and concurrent work [3]:

Ideal Functionality ⟶ Game-based Properties

Major difference: Protection against the malleability attack **and** reply support!

[1] J. Camenisch and A. Lysyanskaya. A formal treatment of onion routing. In Crypto, 2005.
[2] C. Kuhn, M. Beck, and T. Strufe. Breaking and (partially) fixing provably secure onion routing. In IEEE S&P, 2020
[3] M. Ando and A. Lysyanskaya. Cryptographic shallots: A formal treatment of repliable onion encryption. Eprint

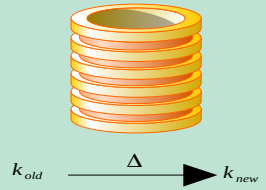
Summary



Onion Routing and Malleability attack

Challenge with Replies

Prove it!



Implicit Payload Authentication

Thank you!

https://eprint.iacr.org/2021/1178