

# Onion Routing with Replies

Christiane Kuhn

Karlsruhe Institute of Technology

Dennis Hofheinz

ETH Zuerich

Andy Rupp

Université du Luxembourg

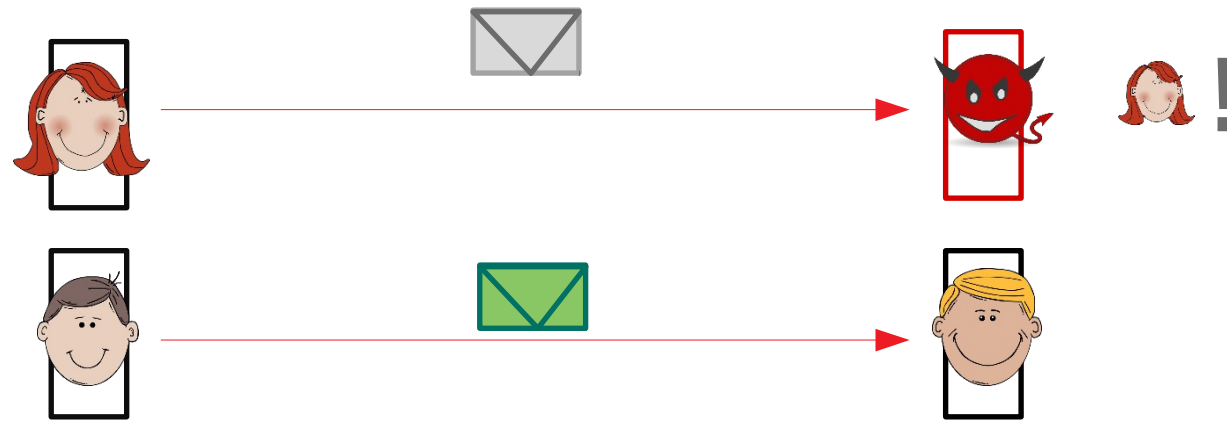
Thorsten Strufe

Karlsruhe Institute of Technology

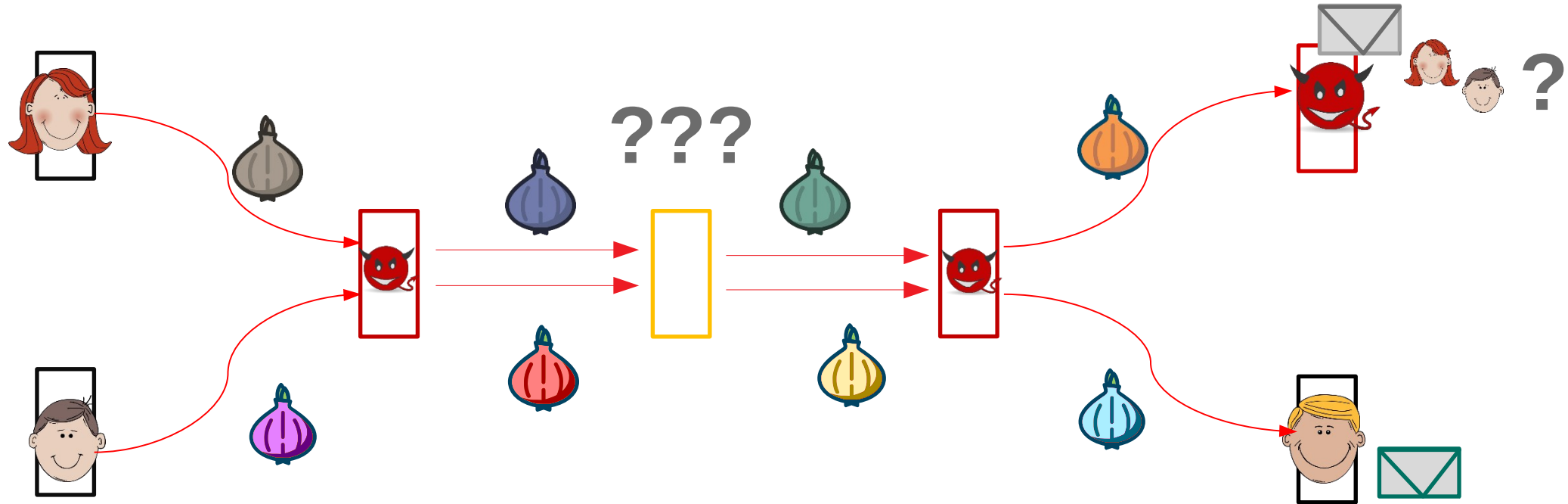
KASTEL Research Laboratories



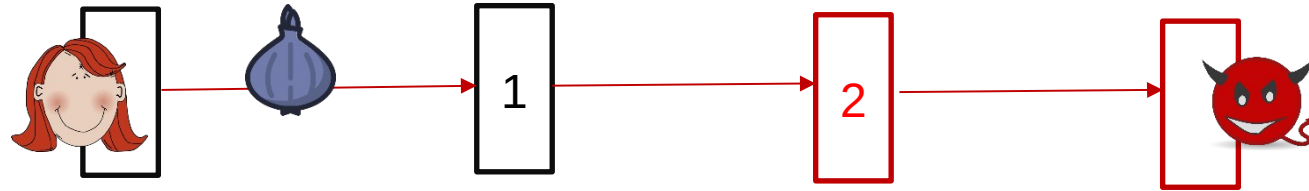
# Goal: Relationship Privacy



# Onion Routing

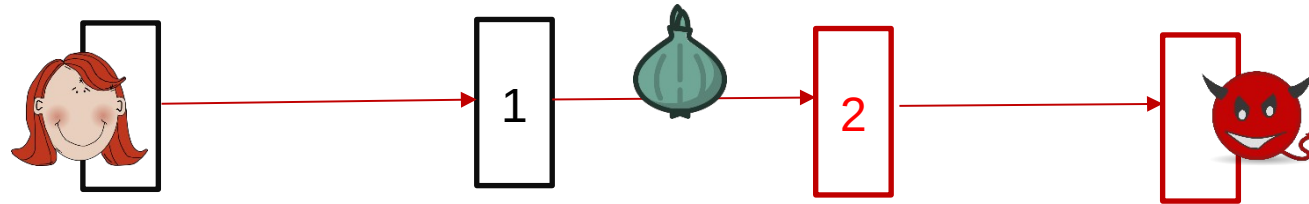


# Header and Payload



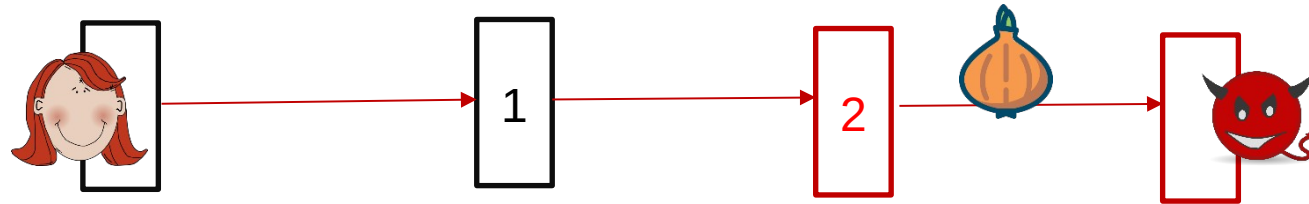
G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In IEEE S&P, 2009

# Header and Payload



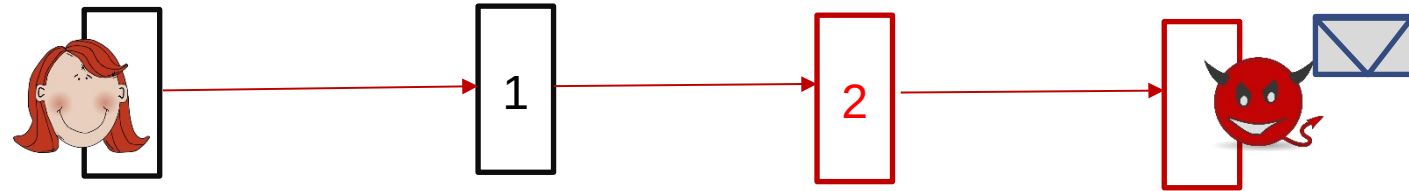
G. Danezis and I. Goldberg. Sphinx: A compact and provably secure mix format. In IEEE S&P, 2009

# Header and Payload



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



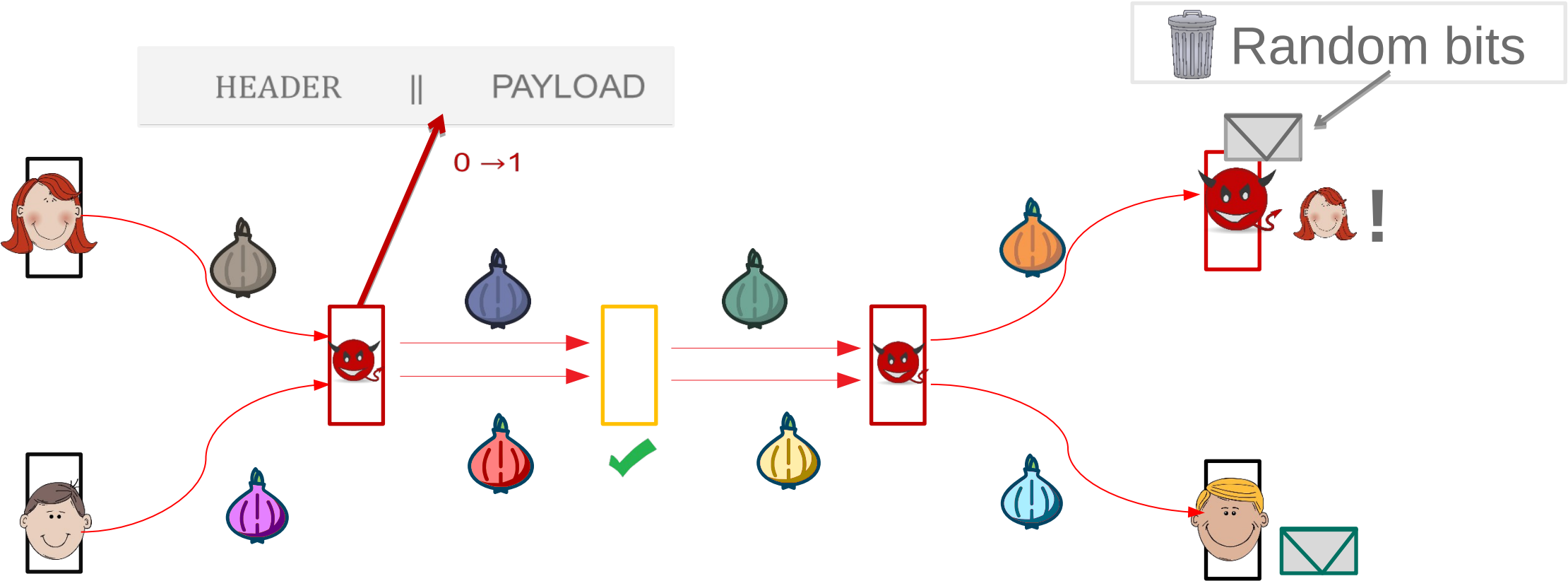
## Integrity



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

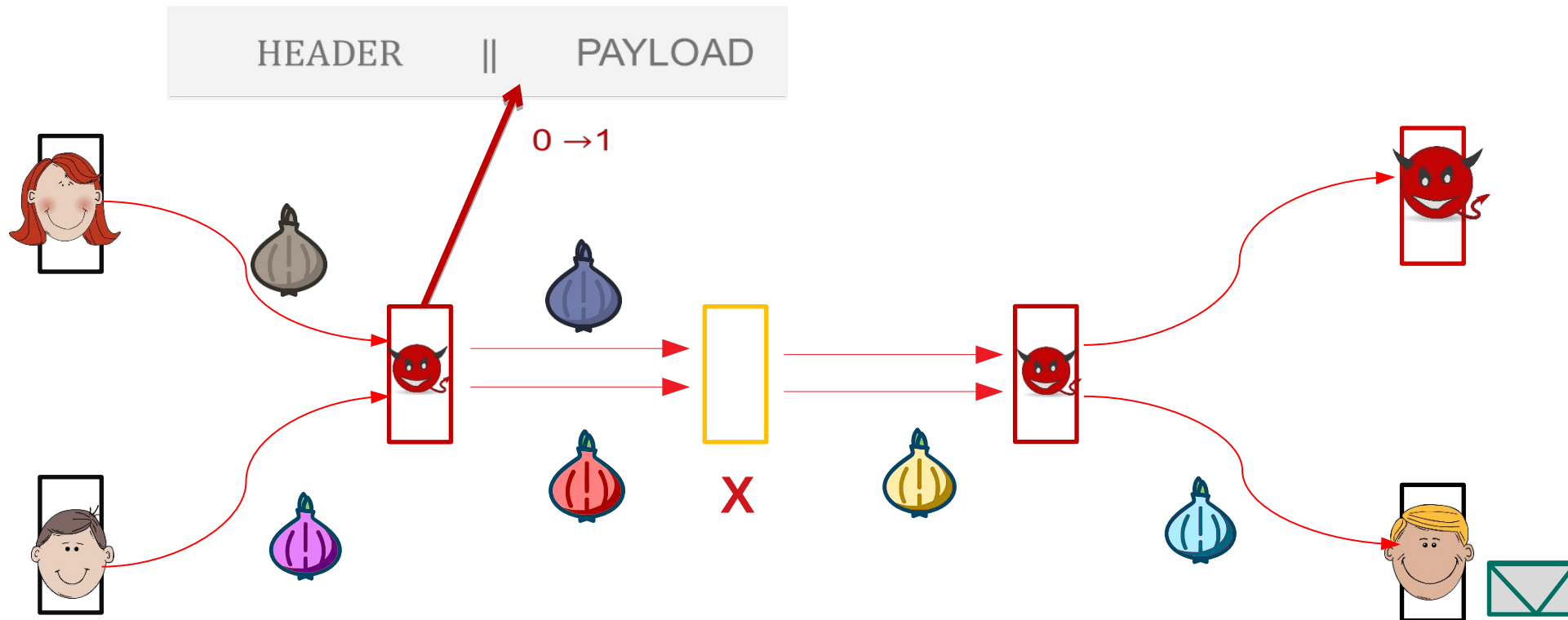


# Malleability Attack



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

## Reliable 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

Reply-Request Indistinguishability

Malleability Attack

**Payload Authentication for Replies**

# ... authentication for replies

Sender: cannot precalculate

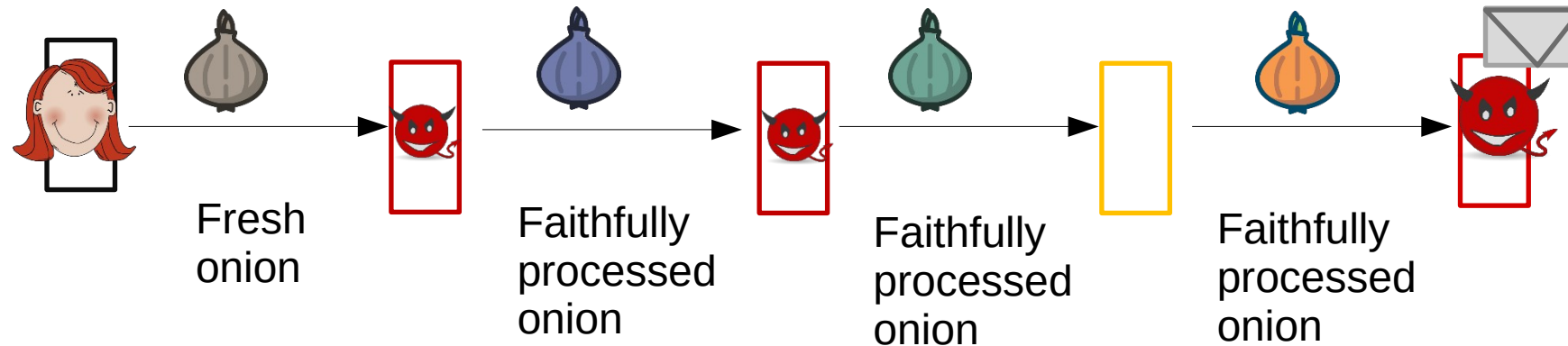
Receiver: cannot be trusted to know parts of other onion layers



# Secure, reliable Onion Routing Protocols

# Protocol 1: SNARGs

Idea: Prove you did it right!



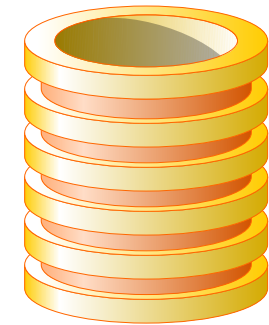
**Authentication chain!**

# Protocol 2: Updatable Encryption

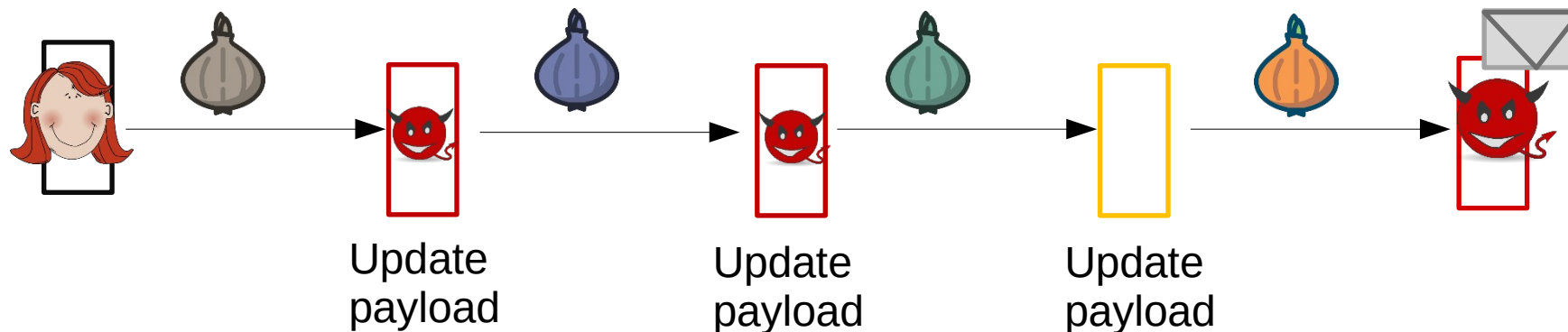
## Idea: Updatable Encryption

Update stored ciphertexts from old to new key  
 With update token

Provides **plaintext integrity**

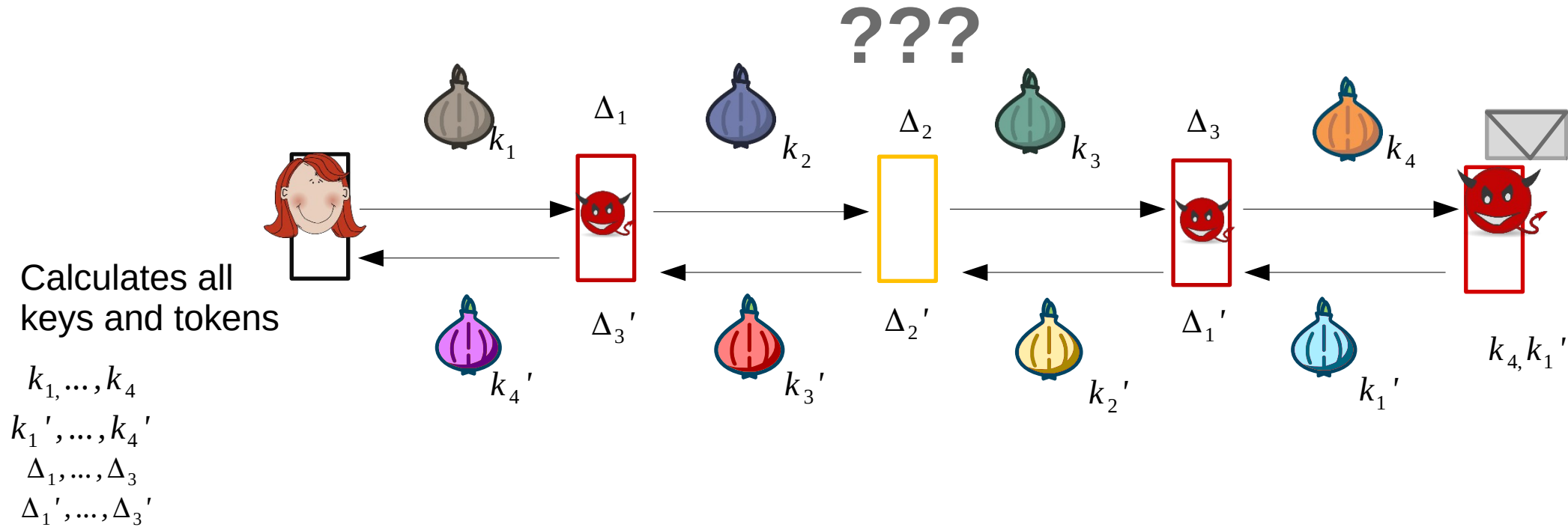


$$k_{old} \xrightarrow{\Delta} k_{new}$$





# Protocol 2: Updatable Encryption



**Modified?** Plaintext Integrity!

# Reusable Security Properties

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

Ideal Functionality  $\longrightarrow$  Game-based Properties

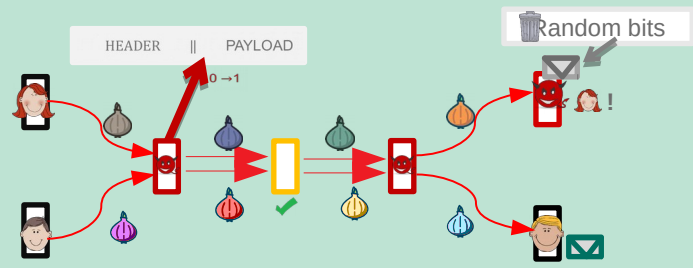
**Major difference:** Protection against the malleability attack **and** reply support!

[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



The diagram illustrates an onion routing network where a message is sent through multiple relays. A packet is shown with a 'HEADER' and 'PAYLOAD'. A red arrow indicates a bit flip from 0 to 1 in the header. A yellow box highlights a relay where the packet is modified. A 'Random bits' box shows a trash can icon, indicating that the original header information is discarded. The message eventually reaches its destination, which is shown with a checkmark and an envelope icon.


**Onion Routing and Malleability attack**



A cartoon onion character with a red devil-like face and horns is shown next to a real onion. The text 'My onion!' is written in red.

**Challenge with Replies**

Prove it!



A stack of gold coins is shown above a diagram of a key transition. The diagram shows a horizontal arrow pointing from  $k_{old}$  to  $k_{new}$ , with a  $\Delta$  symbol above the arrow.

**Implicit Payload Authentication**

# Thank you!

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