Secure Anonymous Acknowledgments in a Delay-Tolerant Network

Edoardo Biagioni
University of Hawai’i at Mānoa

ICNC 2023
Honolulu, Hawai'i

esb@hawaii.edu

Outline

- Background
  - Acknowledgements
  - Delay-Tolerant Networks
  - AllNet peer-to-peer network
- Secure Acknowledgements
  - Anonymous Acknowledgements
- Applications
  - Selective Acknowledgements

TCP Acknowledgements

- Every TCP transmission must be confirmed by the receiver
- the sender marks each transmission with a sequence number
- the receiver sends back a message confirming it has received every byte of that transmission: an **acknowledgement**
  - TCP acks are cumulative: an ack for a later transmission acks all prior transmissions
- TCP acks are sent in the clear and are not secure: anyone that knows the sequence number can send a spoofed TCP ack

Delay Tolerant Networks

- an ad-hoc network supports transmission directly from one user device to a directly-connected user device
  - even without any network infrastructure
  - and supports multiple hops across user devices
- a device that moves between two groups can deliver messages
  - it may cache any messages received
  - and later deliver those messages to devices that haven't seen them yet
  - this is a **data mule** or **sneakernet**
- because data delivery may be slow,

  upper-layer protocols must be designed for the DTN

AllNet Peer-to-Peer Networking

- AllNet is designed to work well over ad-hoc and delay-tolerant networks
  - and to use the Internet when available
- main application: user-to-user chat, which is naturally delay-tolerant
- on ad-hoc, broadcast to everyone, forwarding cached messages
  - a message has a 16-byte message ID
  - messages are only forwarded to hosts that have not yet received them
- all data messages are encrypted
  - but the message ID is sent in the clear part of the message

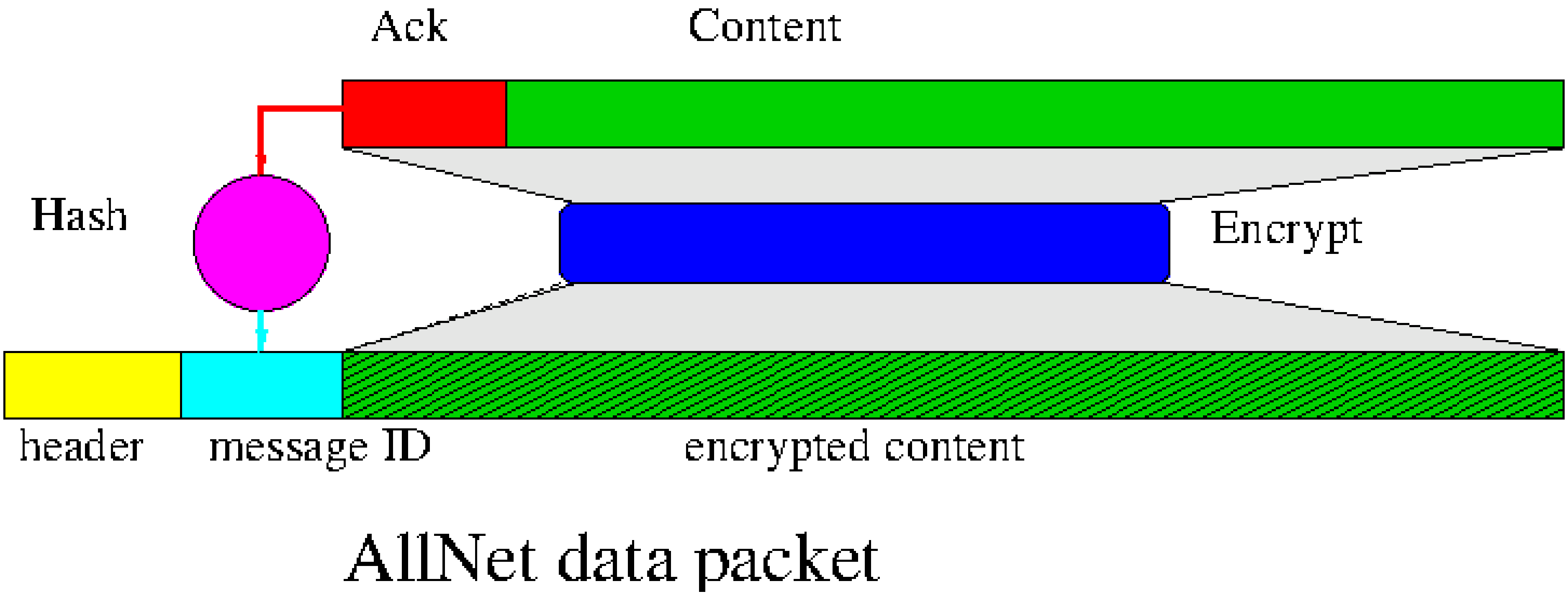
http://alnt.org

AllNet Caching and Secure Acknowledgements

- cache size is finite, so messages should be evicted from the cache once they are received by the final device
- before originating a message, the sender randomly selects a 16-byte message ack
  - the hash of the ack is the message ID
- a device that receives an ack can hash it to see if it matches any of its cached messages
  - and evict those messages from its cache

only the intended receiver can issue valid acks

AllNet Secure Acknowledgments



Secure and Anonymous Acknowledgements

- Since the ack is encrypted in the data message,

  only the intended receiver can issue valid acks
- Secure: attackers would need to break either the encryption or the hash function to generate a spoofed ack
- Anonymous: an AllNet ack is a random number, and the message ID is the hash of a random number, so neither identifies sender or receiver (AllNet addresses are optional)

Acknowledging Partial Transmission

- larger messages are sent as a number of fragments
- each message has an ack and a message ID
- likewise, each fragment has a fragment ack and fragment ID
  - a fragment ack removes the corresponding fragment from caches
- receivers can send a message ack even before receiving all fragments
  - the sender then stops sending the remaining fragments
  - intermediate devices stop caching all fragments of the message
- if a message contains an image in different resolutions, the receiver may ack it after receiving only the smaller-resolution image

Summary and Questions

- randomly generated acks hash to cleartext message IDs
- encrypted ack sent in the message
- receiver that can decrypt the message can issue valid ack
- any device that sees the ack can compare it to the message IDs of its cached messages
- information about AllNet at http://alnt.org/

Questions?