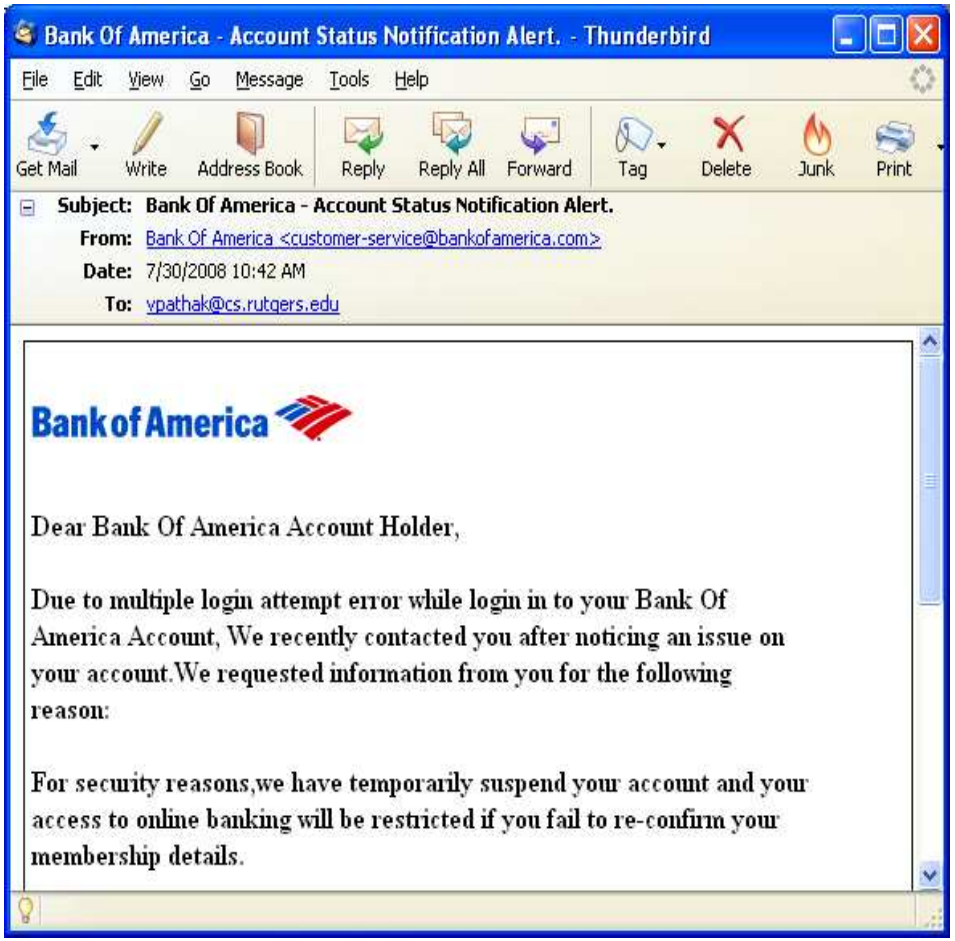
Improving Email Trustworthiness through Social-Group Key Authentication

Vivek Pathak, Danfeng Yao, Liviu Iftode
Department of Computer Science
Rutgers University

Fifth Conference on Email and Anti-Spam
CEAS 2008

Email Trustworthiness

- Should I trust the email?
  - Provide login details
- How is the email trust problem solved
  - Sender authentication
  - Content integrity
- Multi-billion dollar spam industry
  - Message Labs report
  - Phishing emails 78.9%



Traditional Sender Authentication Approaches

- S/MIME
  - Secure email with digital signature
    - Can not forge sender identity
  - PKI deployment
    - How to use across organizational trust boundaries
- PGP
  - Sender identification without trust infrastructure
  - Need user sophistication
    - Maintain key rings

Spam Control Approaches

- DKIM - Domain Keys Identified Mail
  - Infrastructure compatible
  - Authenticate sender domain
    - Need administrative control and upgrade
    - DNS attacks
- Content classification
  - Identify spam features and keywords
  - False positives of spam filters

Threats

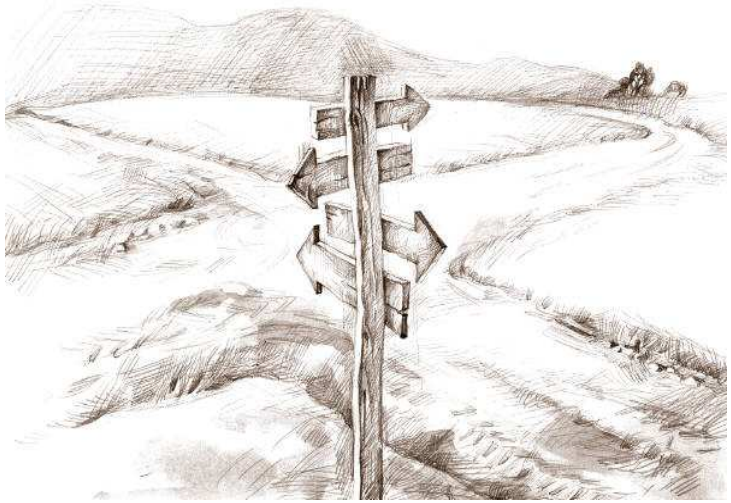
- Network attacks
  - Impersonation
  - Message delivery
    - Add arbitrary messages
    - Delete messages
    - Modify content en-route
      - Man in the middle attack
- Misbehaving participants
  - Stopping failures
  - Byzantine faults
    - Malicious participants may appear to work fine and mislead participants

Social-Group Key Authentication

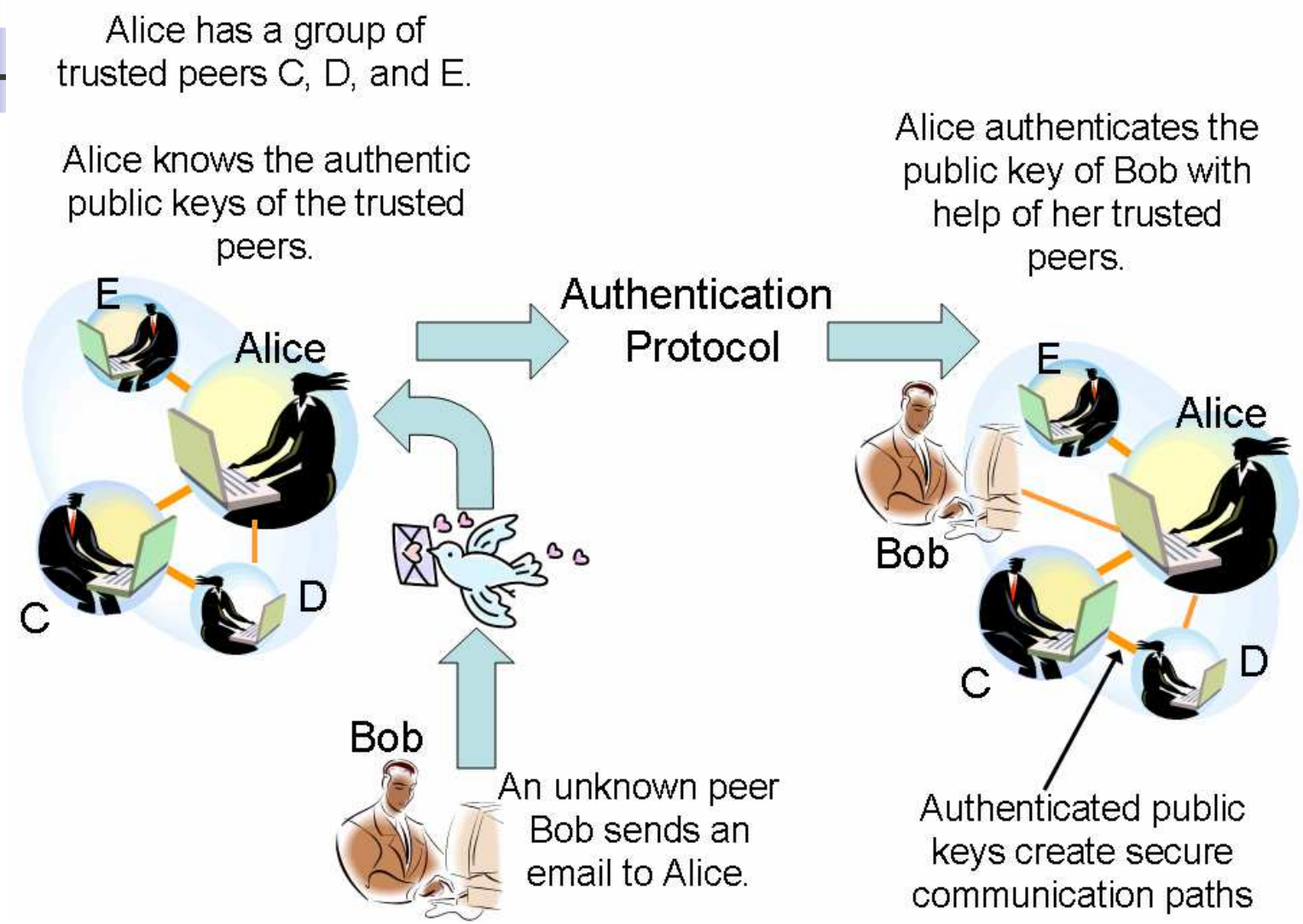
- Complementary approach for sender-authentication
- Use email user base characteristics
  - Pre-existing social groups of collaborating users
  - Unsophisticated but many
- Address issues with current solutions
  - Infrastructure free end-to-end solution
  - Automatic operation to help unsophisticated users
  - Communication across organizational boundaries
  - Robustness
    - Byzantine faults
    - Man in the middle

Outline

- Introduction
- **Social-group key authentication protocol**
- Implementation
- Experimental evaluation
- Conclusion

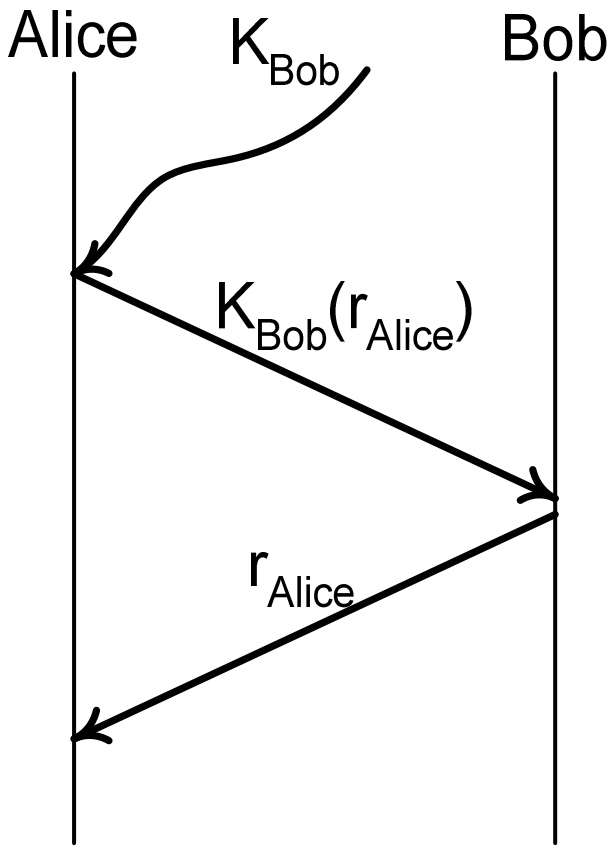


Solution Approach



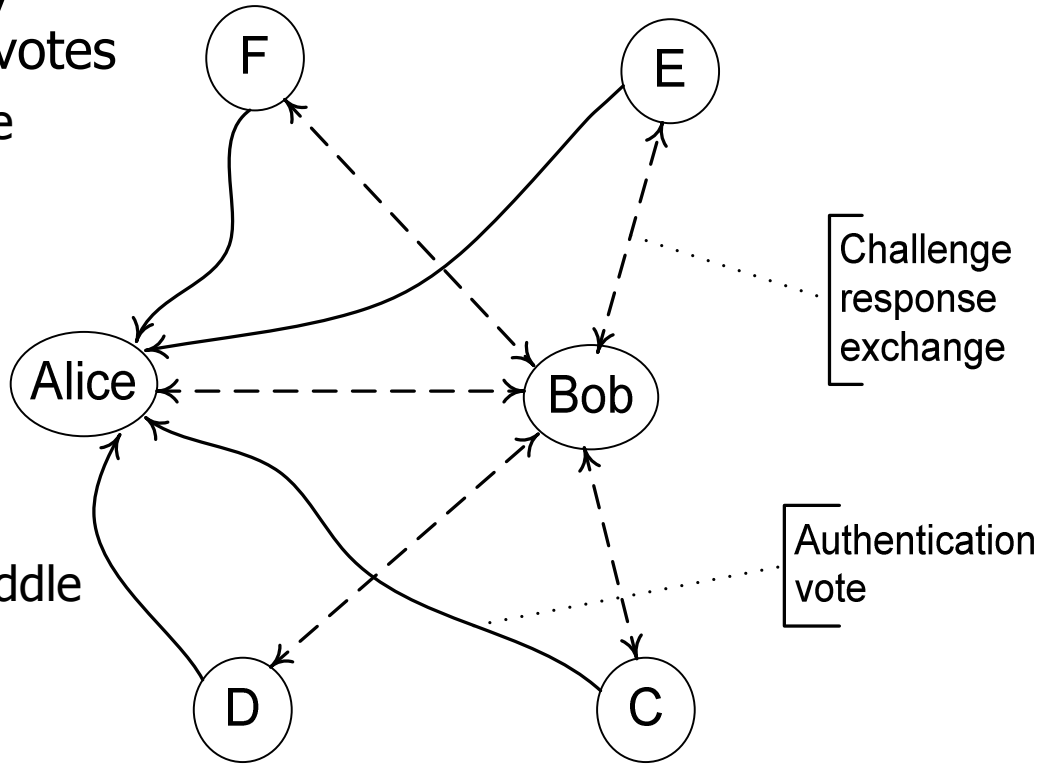
Underlying Email Trust Environment

- Hard to subvert message delivery
- Easy to spoof messages
- Challenge-response protocol
  - Authenticate self-generated public keys of peers
  - Susceptible to man-in-the-middle attack



Authentication Protocol

- Public key authentication by majority on authentication votes
  - Votes are calculated by the social group
  - Vote value depends on challenge response result
- Votes will agree
  - All participants are honest
  - There is no man-in-the-middle attack



Achieving Robustness

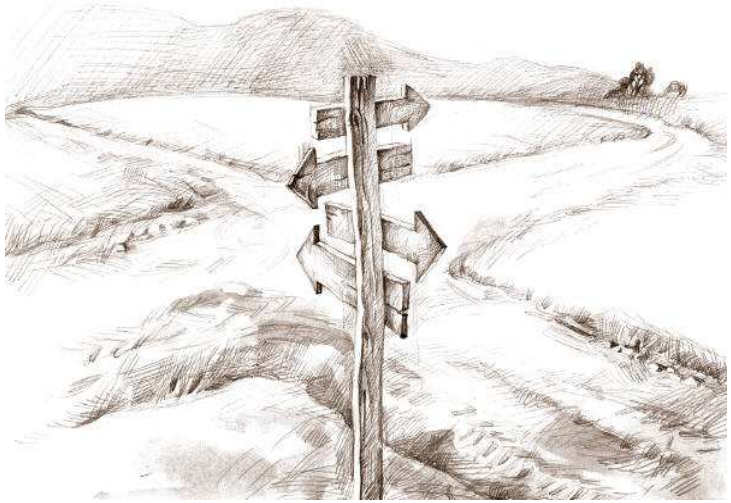
- On observing disagreement in authentication votes
  - Inform all peers
  - Each peer shares the votes it received with its peers
- Identify and remove peers causing the disagreement
  - Man in the middle attacks
  - Malicious or faulty participants
- Correct and robust against Byzantine faults if more than 2/3 of peers are honest
  - V. Pathak and L Iftode. Byzantine fault tolerant public key authentication in peer-to-peer systems *Computer Networks* 50(4) 2006

Overlay Authentication Protocol on Email Messages

- SMTP extension headers X-Bft-Auth-*
  - Public key of the sender
  - Digital Signature
  - Cryptographic data
    - Challenge-response nonce and authentication vote
- Protocol Operations
  - *Email_Peer*: Request authentication
  - *Email_Response*: Return authentication vote
  - *Infer_Trust*: Decide authenticity by majority
- Lazy and eager modes of operation
  - Additional email messages

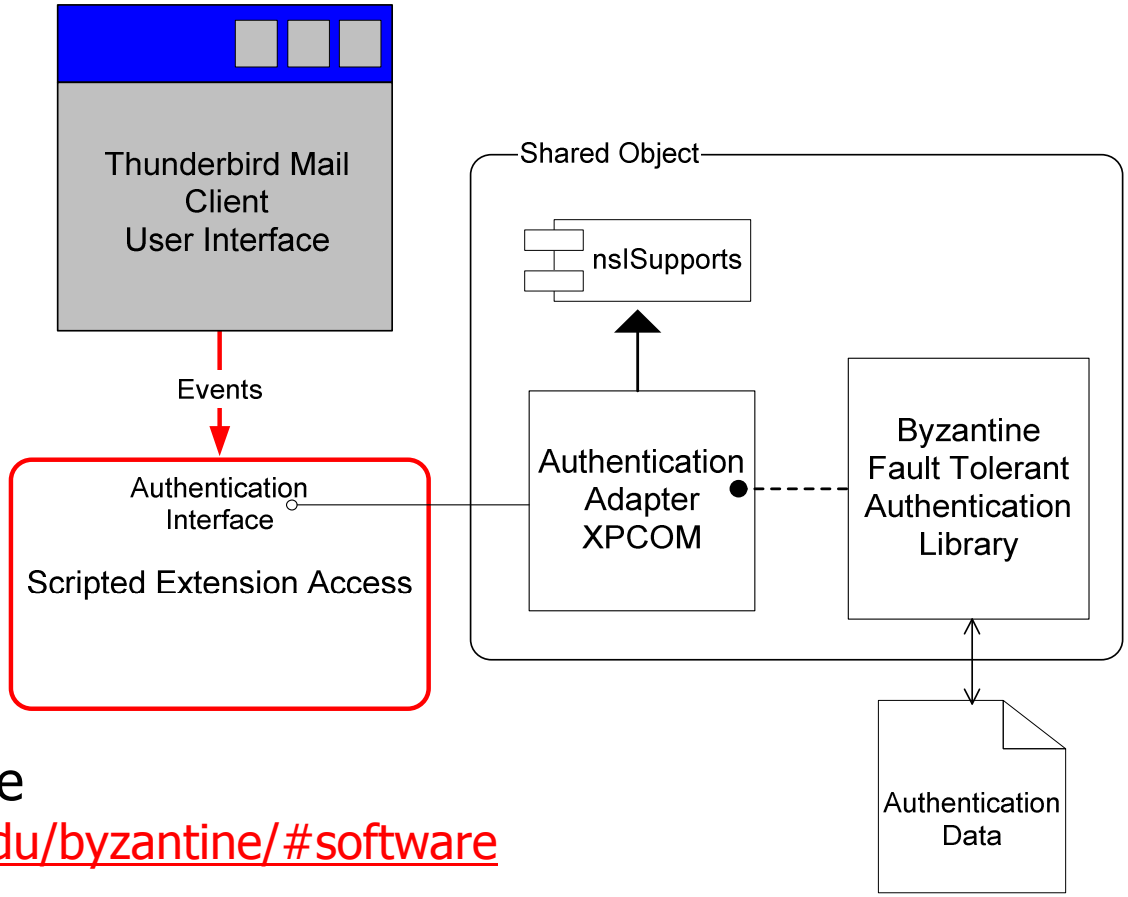
Outline

- Introduction
- Social-group key authentication protocol
- **Implementation**
- Experimental evaluation
- Conclusion



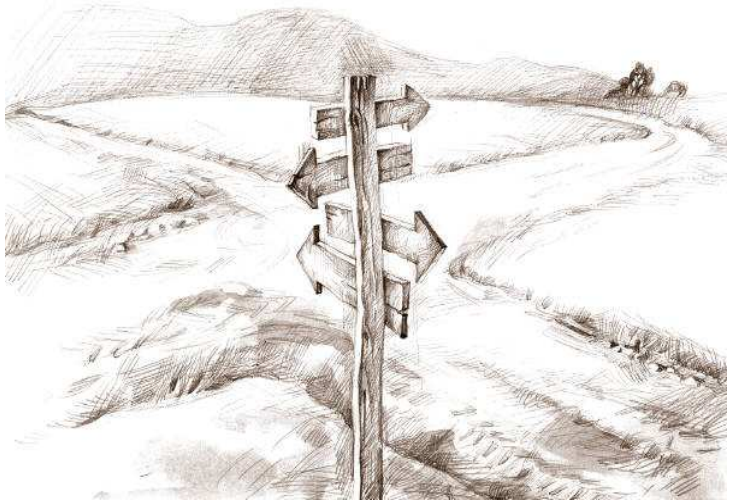
Authentication Plug-in for Thunderbird Email Client

- Implement authentication in the client
  - C++ Library
  - Javascript plugin
- Piggy-back on emails
- Able to cache protocol messages
- Downloadable software
  http://discolab.rutgers.edu/byzantine/#software



Outline

- Introduction
- Social-group key authentication protocol
- Implementation
- **Experimental evaluation**
- Conclusion

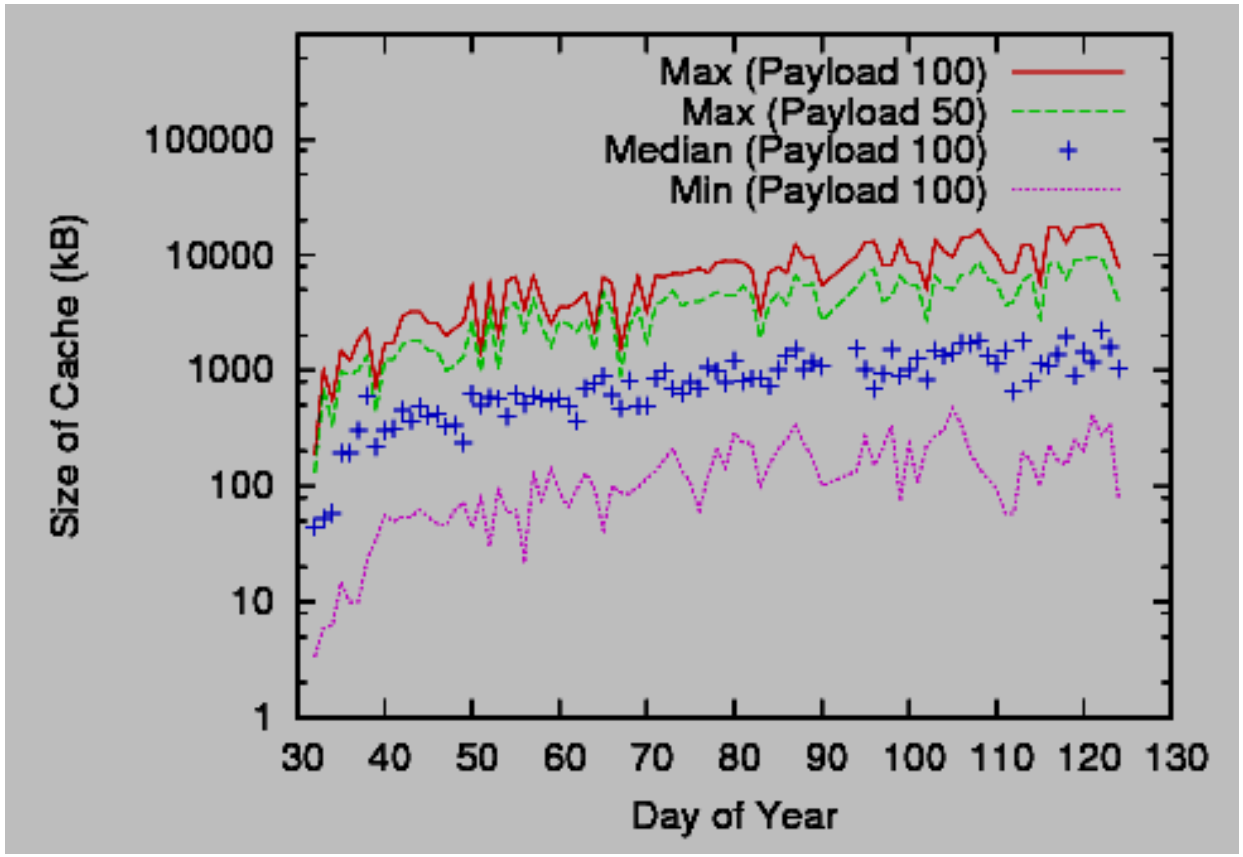


Experimental Methodology

- Protocol execution on email trace
  - Collect anonymous trace from email server log
    - cs.rutgers.edu 1.19 million emails in 92 days
    - ask.com 2.54 million emails in 56 days
  - Select peers from email social group
- Experimental parameters
  - Message payload
  - Lazy and eager modes of operation
- Measurements
  - Fraction of peers authenticated in 92 days
  - Overhead introduced

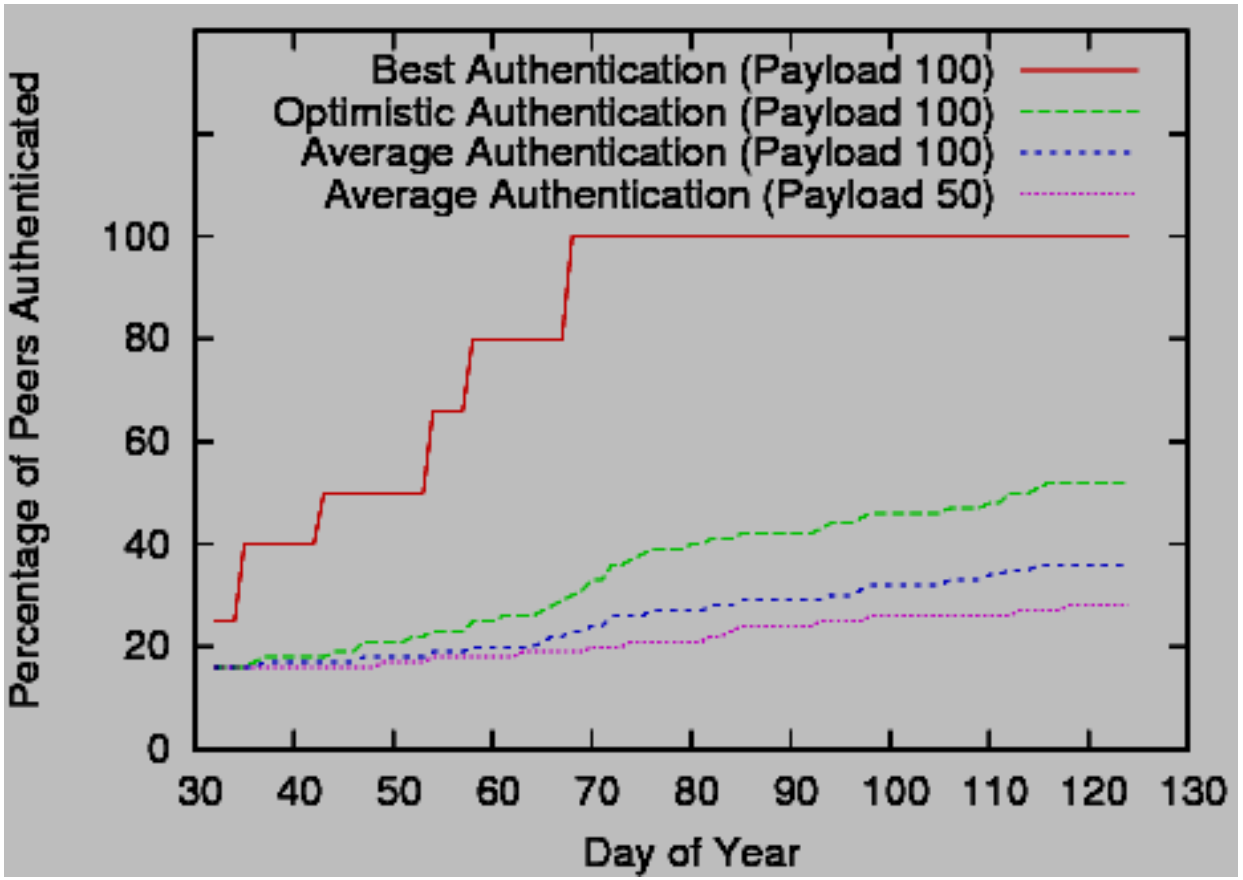
Cost of Lazy Mode Authentication

- No additional email messages
- Use local cache to store and forward protocol messages



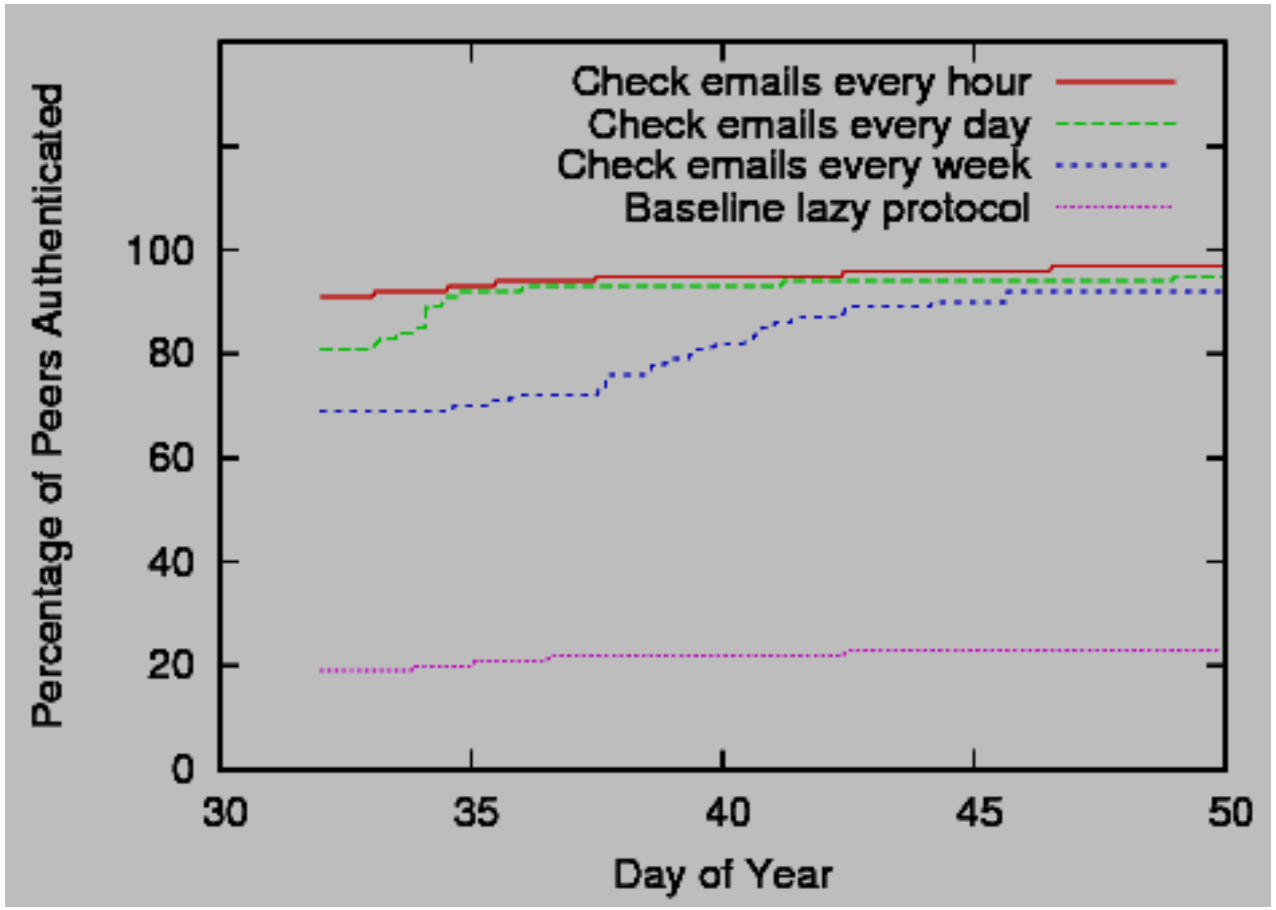
Authentication Performance in Lazy Mode

- Full backward compatibility
  - Too slow
- Protocol messages piggy-backed on real emails
  - Avoid spam filters
- Optimistic authentication
  - Partial completion



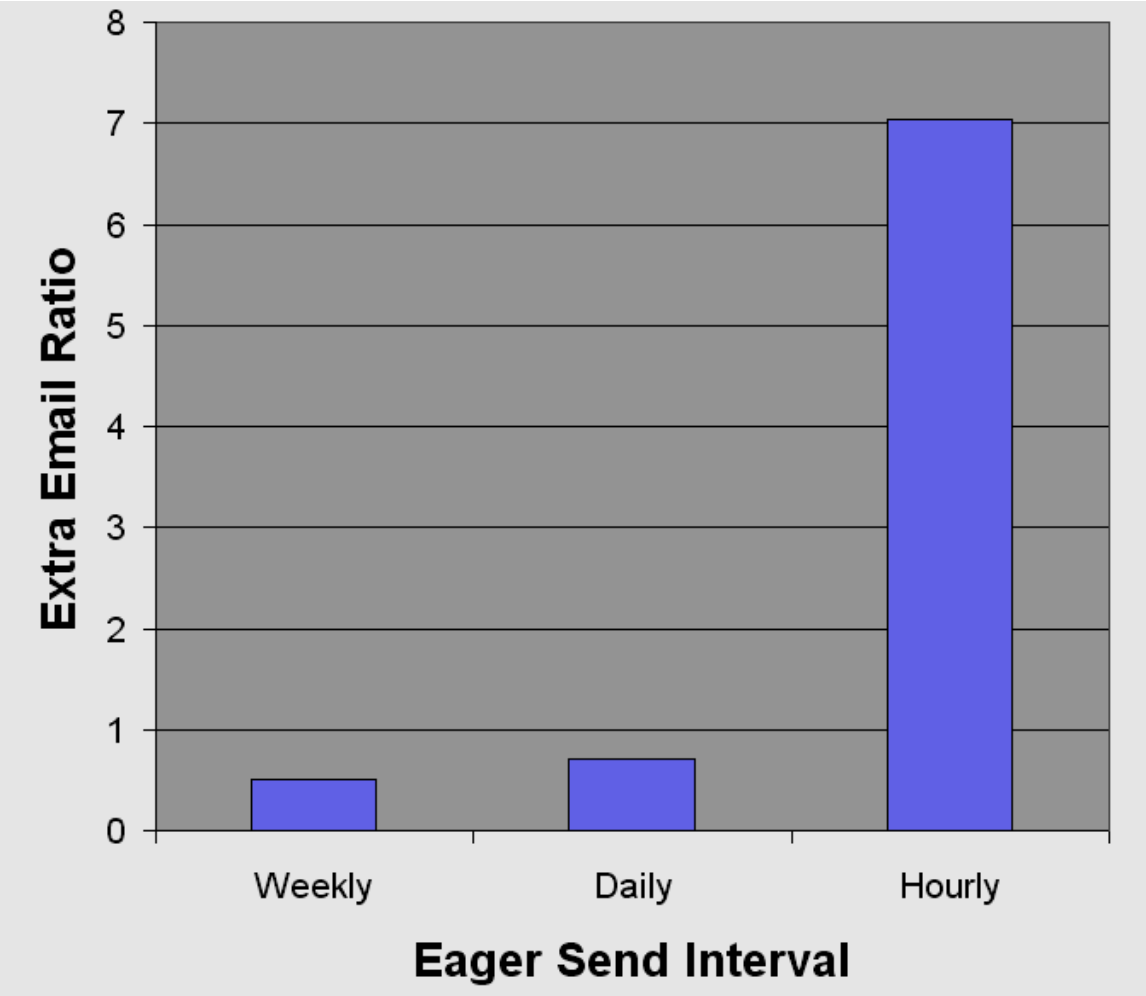
Authentication Performance in Eager Mode

- Rapid authentication
- Need to inject emails periodically
- Authenticate 80% of peers in two periods



Email Overhead Introduced in Eager Mode

- Moderate overhead of new emails
- Overhead increases with greater eagerness

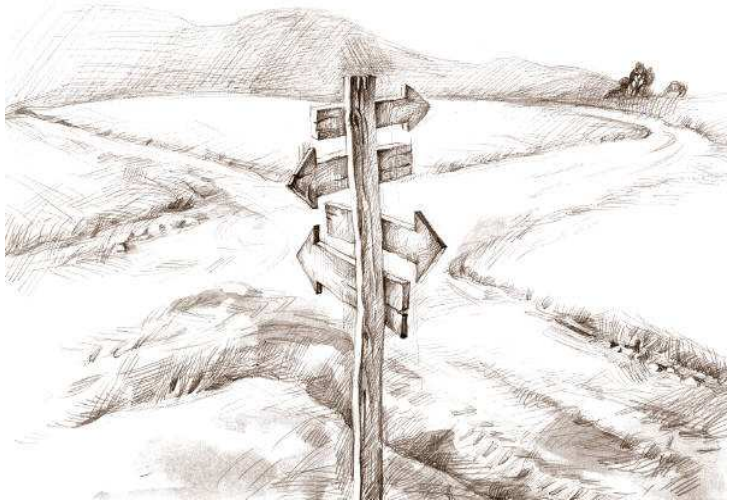


Usability Discussion

- Lazy mode
  - Dependent on natural email communication pattern
  - Emails have different urgency levels
    - No way to get quicker conclusion
  - Users need to send and receive emails for protocol to progress
- Eager mode
  - Risk of getting caught in spam filters
  - Email client needs to be online to participate in the authentication protocol

Outline

- Introduction
- Social-group key authentication protocol
- Implementation
- Experimental evaluation
- **Conclusion**



Summary

- Social-group key authentication
  - Use social-groups of email users
  - Tolerate Byzantine faults
  - Automatic sender authentication
  - Compatible with existing infrastructure
- Experimental results
  - Modest overhead
  - Performance issues handled through eager mode

Conclusions

- Authenticate public keys through Social-group key authentication protocol
- Traditional digital signatures
  - Sender authentication
  - Content integrity
- Improve Email Trustworthiness

Future Work

- Denial of Service
  - Place limits on eager send mechanism
- Impact of user input and overrides
  - Large scale deployment
  - Usage data collection
- Use authenticated identities for
  - Collaborative spam control
  - Social networks
    - FaceBook, MySpace etc.
- Implement social-group key authentication on online social networks

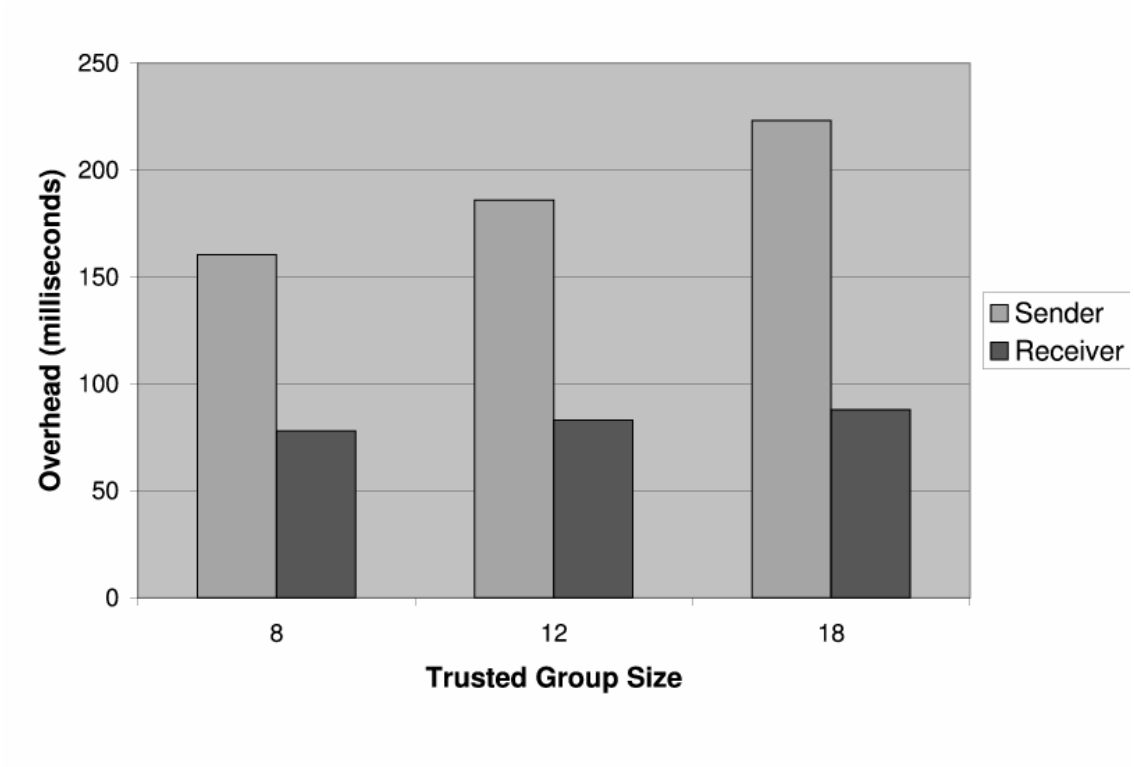
Questions

Thank You

Vivek Pathak
vpathak@cs.rutgers.edu

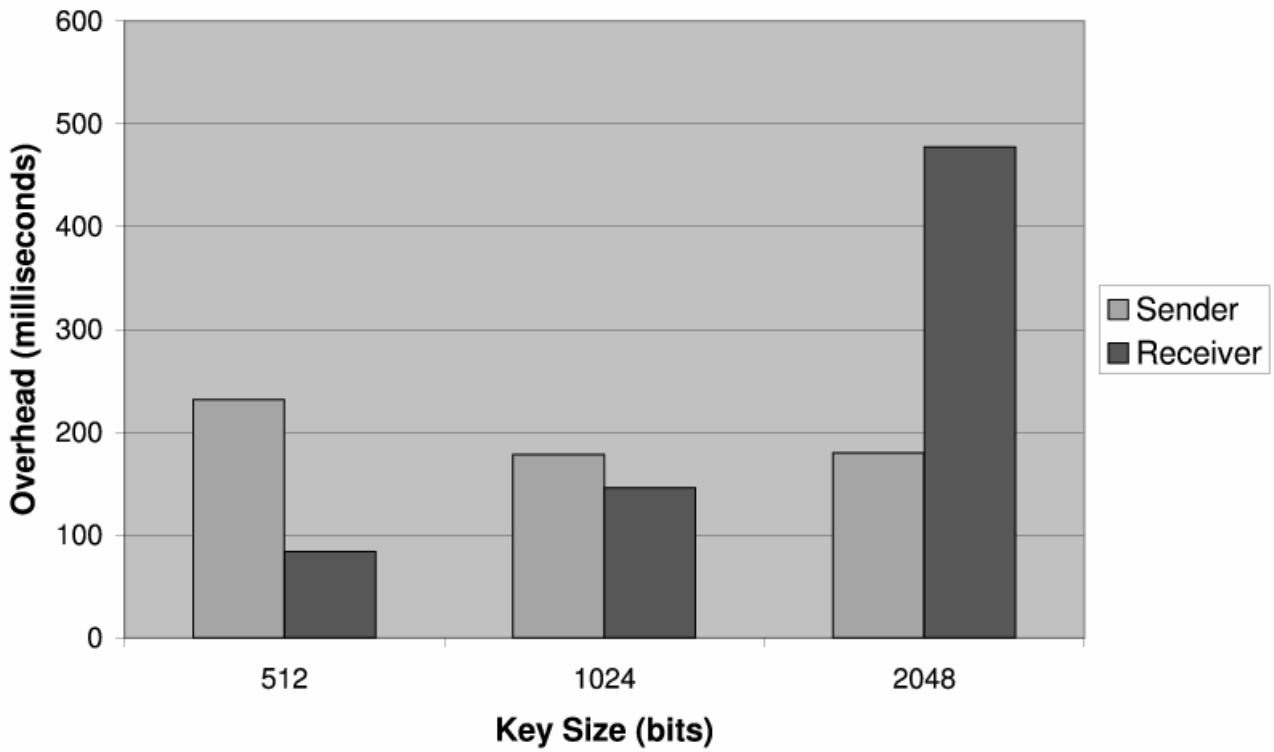
Microbenchmark Trusted Group Size

- Impact of trusted group size
- Sender does more work
- Total overhead is modest



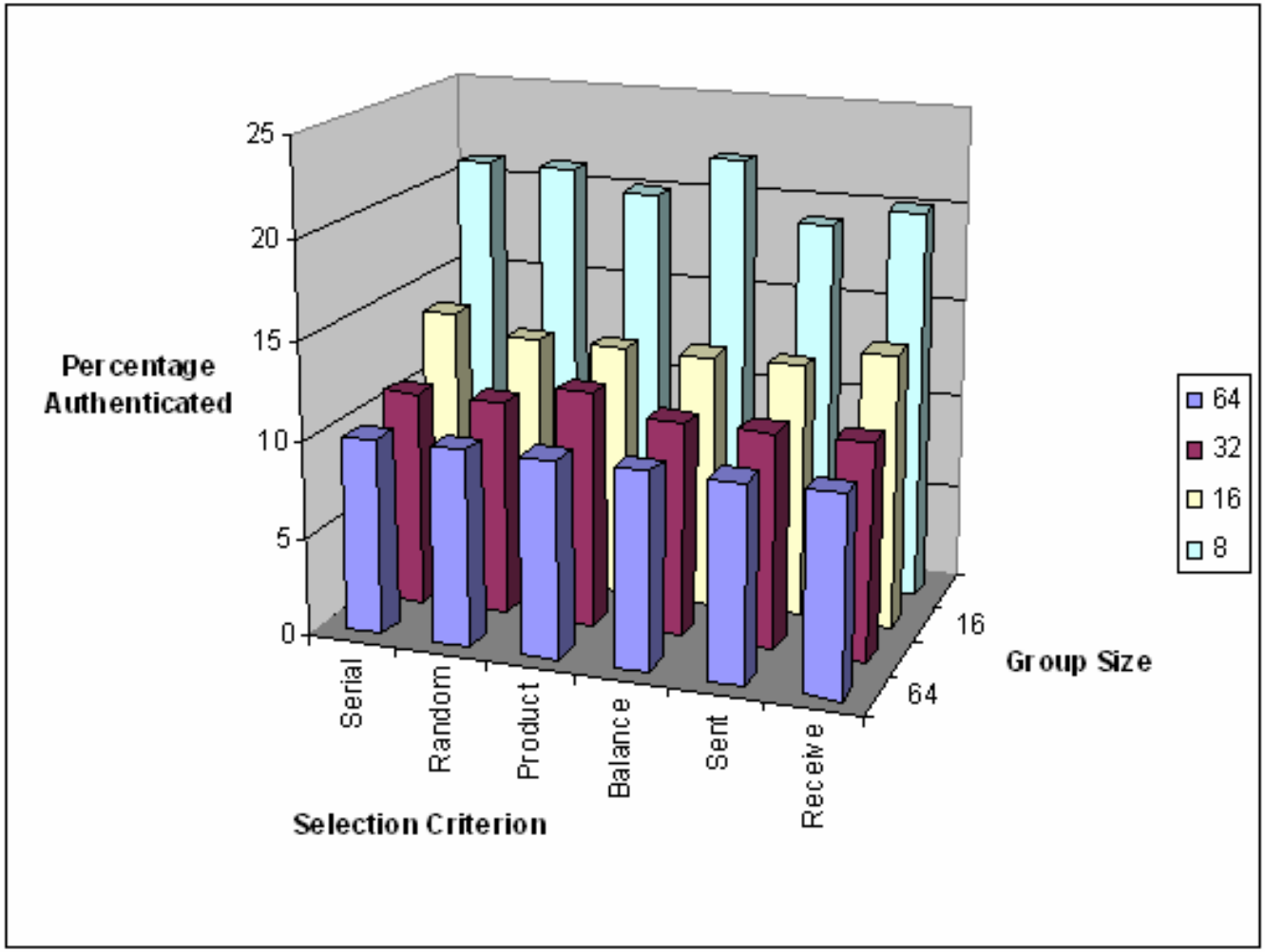
Microbenchmark Public Key Length

- Increased receive overhead
- Validation of challenge response results
- Feasible latency of less than 1 sec.



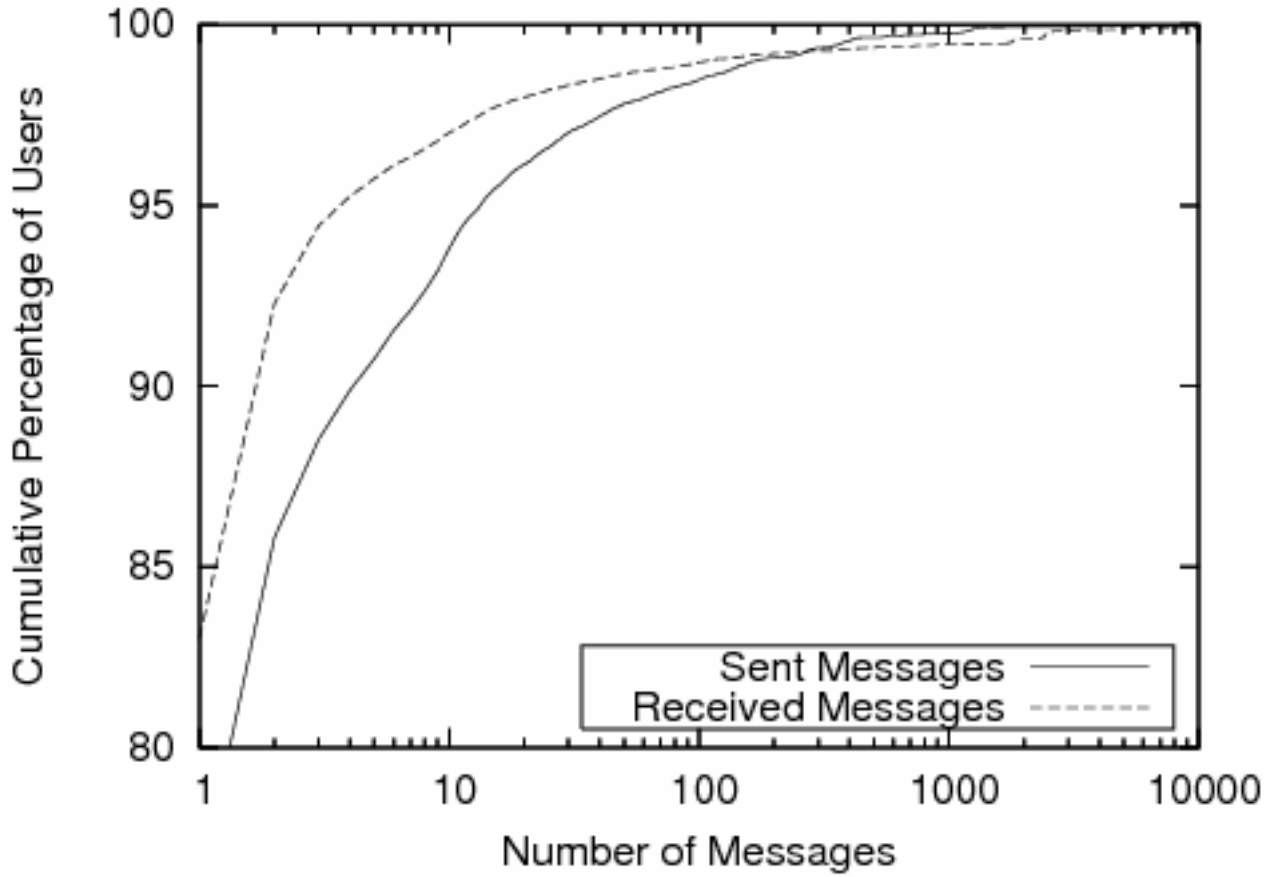
Bootstrapping

- Selection of trusted group affects authentication performance
- Effect is more intense at smaller group sizes



Asymmetric Send-Receive Behavior

- Most email users in sample have low activity



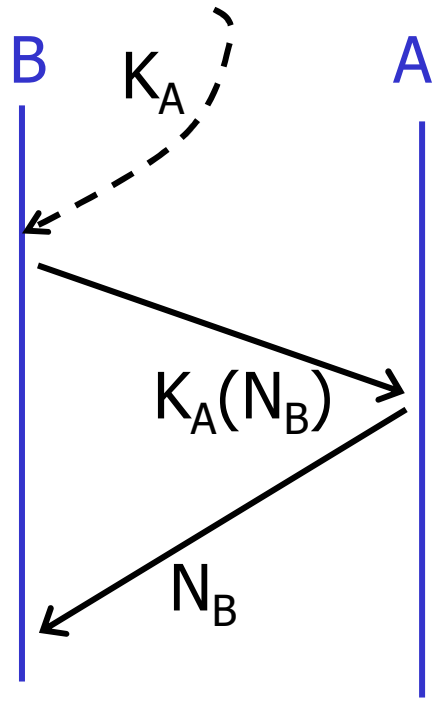
Attack Model

- Malicious peers
  - Honest majority
  - At most t of the n peers are faulty or malicious peers where $t = {}^{1-6Ø}/_{3}\, n$
- Passive adversaries
- Active adversaries
  - Relax network-is-the-adversary model
    - Unlimited spoofing
    - Limited power to prevent message delivery

Authentication Model

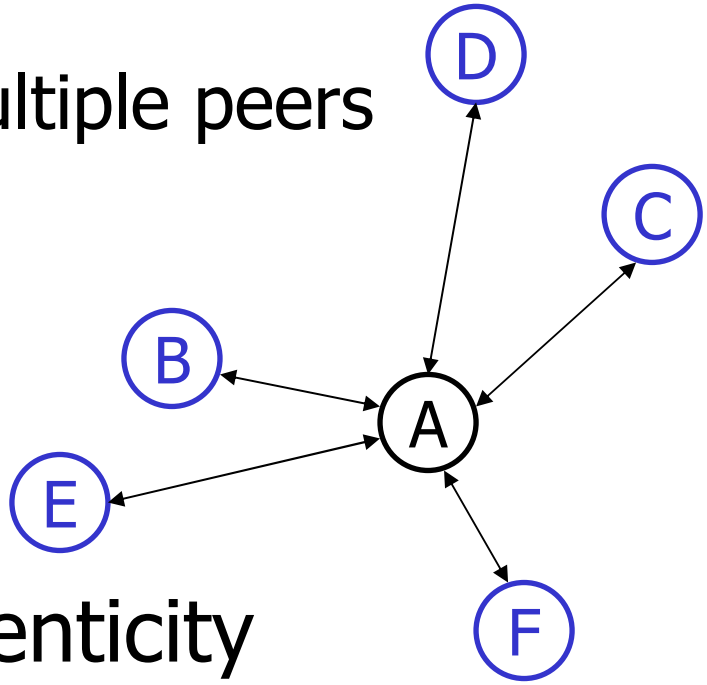
- Challenge-response protocol
  - No active attacks
- Man in the middle attack
  - Limited number of attacks
- Proof of possession of K_A

  $\{B,A,Challenge,K_A(r)\}_B$, $\{A,B,Response,r\}_A$



Authentication Model

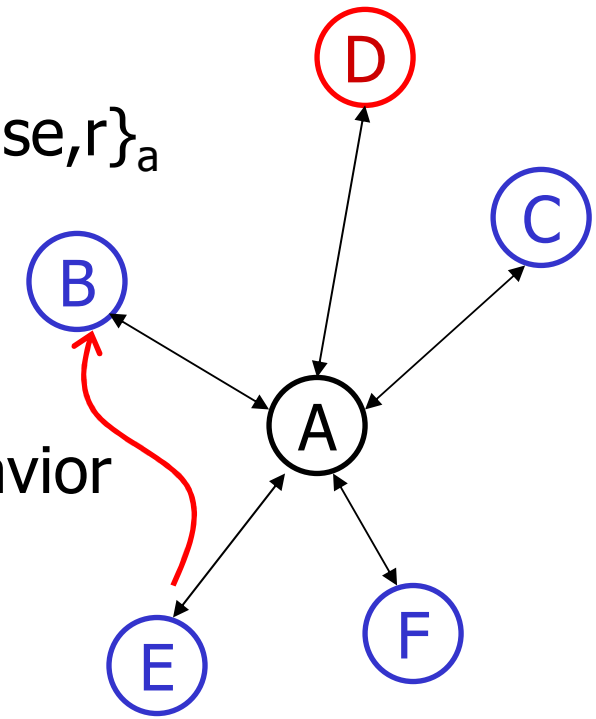
- Distributed Authentication
  - Challenge response from multiple peers
  - Gather proofs of possession



- Lack of consensus on authenticity
  - Malicious peers
  - Man-in-the-middle attack

Authentication Correctness

- Validity of proofs of possession
  - $\{e,a,Challenge,K_a(r)\}_e$, $\{a,e,Response,r\}_a$
- All messages are signed
  - Required for proving malicious behavior
  - Recent proofs stored by the peers



From peers	P_B	P_C	P_D	P_E	P_F
From A	P_B	P_C	P_D	P_E	P_F

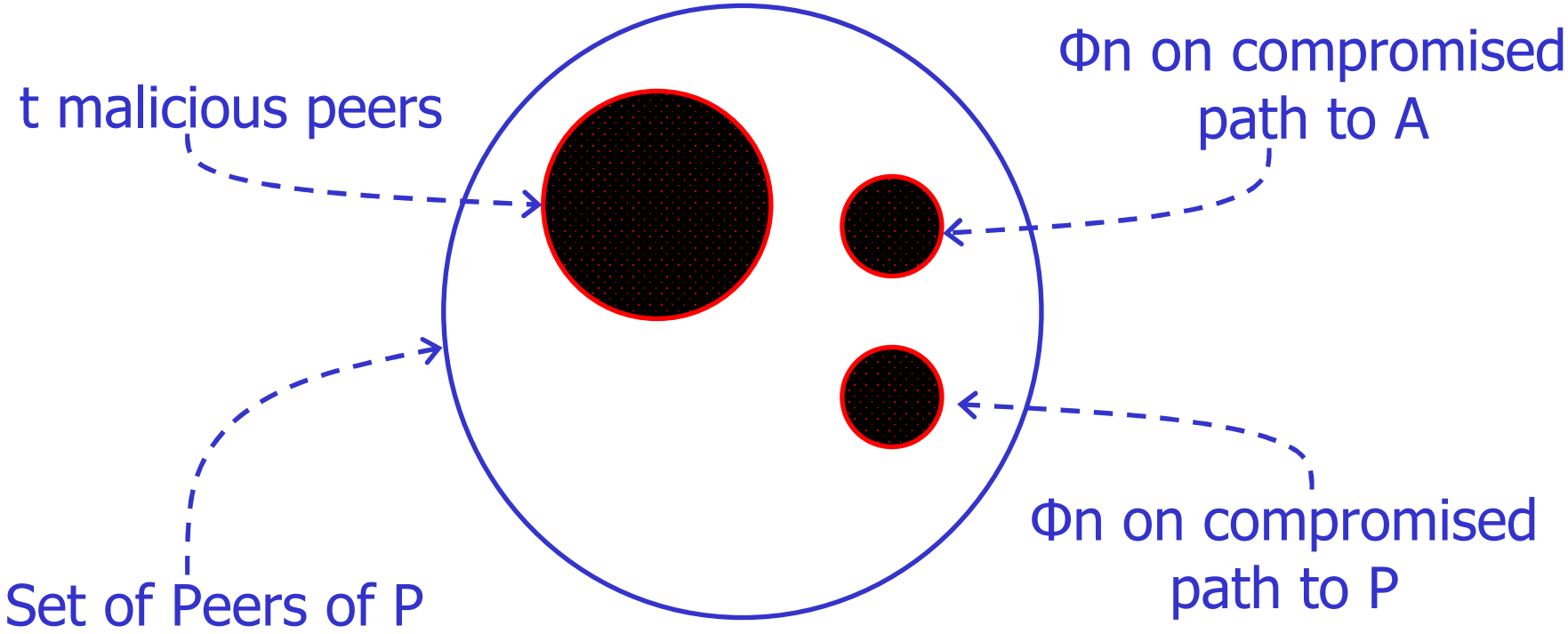
Byzantine Agreement Overview

- Publicize lack of consensus
  - Authenticating peer sends proofs of possession to peers
- Each peer tries to authenticate A
  - Sends its proof-of-possession vector to every peer
  - Byzantine agreement on authenticity of K_A
- Majority decision at every peer
  - Identify malicious peers
  - Complete authentication

From B	1	1	0	1	1
From C	1	1	1	1	1
From D	1	1	1	1	1
From E	1	1	0	1	1
From F	1	1	0	1	1

Byzantine Agreement Correctness Overview

- Consider proofs received at a peer P



Byzantine Agreement Correctness Overview

- $t + 2Ø n$ may not arrive
  - P receives at least $n-t-2Ø n$ proofs
- $t + 2Ø n$ may be faulty
  - P receives at least $n-2t-4Ø n$ correct agreeing proofs
  - P decides correctly by majority if $n-2t-4Ø n > t + 2Ø n$
- Agreement is correct if $t < {}^{1-6Ø}/_{3}\, n$