

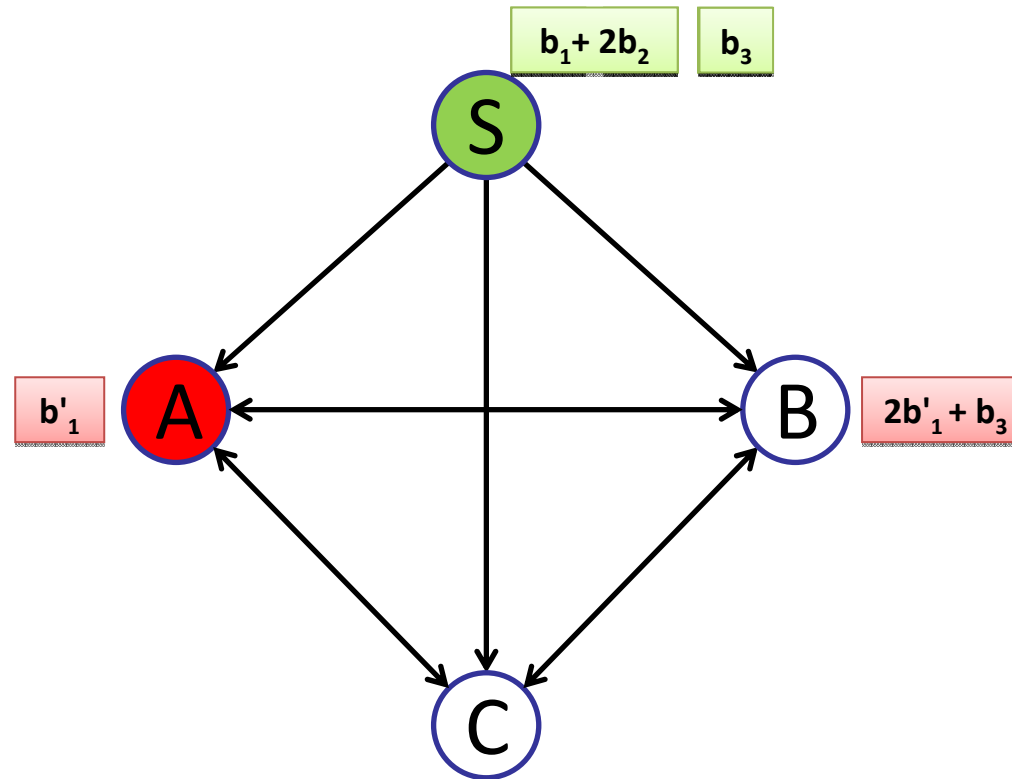
TESLA-Based Defense Against Pollution Attacks in P2P Systems with Network Coding

Anh Le, Athina Markopoulou

University of California, Irvine

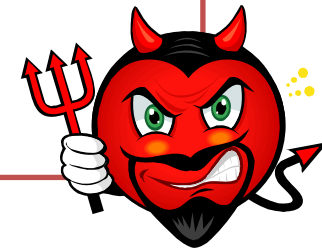


Pollution Attacks in P2P Systems with NC



Pollution Attacks in P2P Systems with NC

- Large number of corrupted packets
- Waste network resources
- Prevent decoding



Prior Pollution Defense Mechanisms

1. Homomorphic Signatures and Hash Functions

- Large verification time
[Boneh09] [Gkantsidis06]

2. Homomorphic MACs (better)

- Only c -collusion resistant, small c
[Agrawal09] [Zhang11]
- Only work on directed acyclic graphs
[Li10]
- No elimination of attackers



Prior Pollution Defense Mechanisms

3. Our prior work: **SpaceMac**

- Provide in-network detection by parent-child cooperation
 - In-network detection does not work when there is colluding adversaries
- Used with a probabilistic non-repudiation protocol to support attacker identification
 - Higher communication overhead per security



Our Proposal

A Complete Defense Mechanism

- In-network detection
- Precise identification
- Arbitrary collusion resistance
- Low overhead
- Require time synchronization



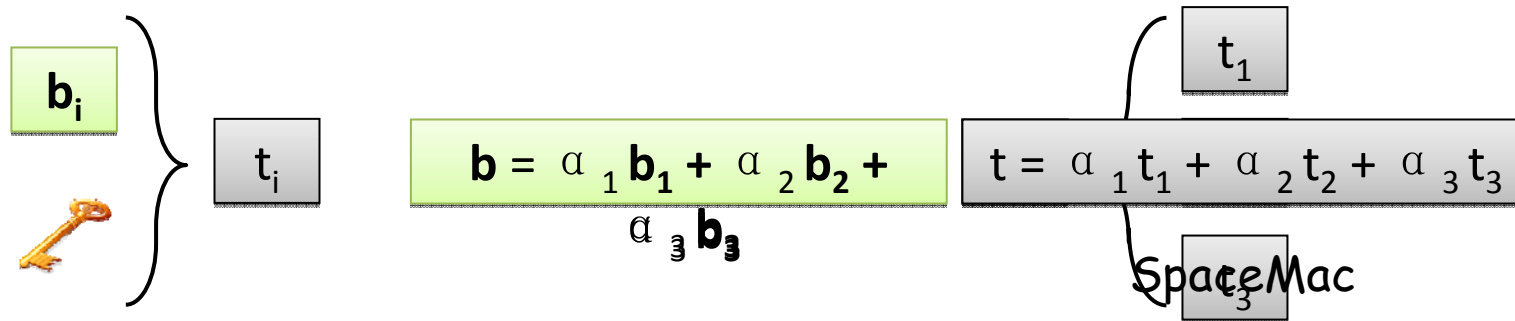
Outline

1. Background and Motivation
 - o Pollution Attacks
 - o Existing Defense
2. Detection Scheme
3. Identification Scheme
4. Performance Evaluation
5. Conclusion



Building Blocks

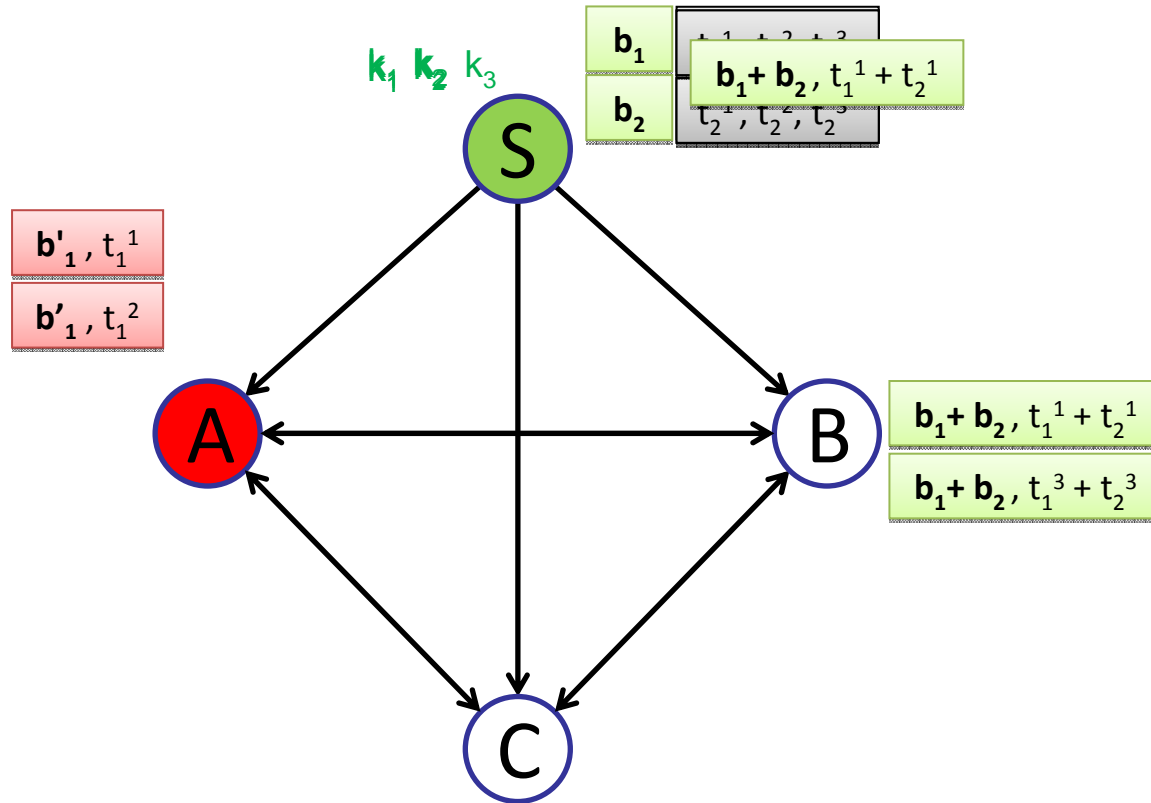
1. Homomorphic message authentication codes (MACs)



2. TESLA broadcast authentication (delayed key disclosure)



TESLA-Based Detection



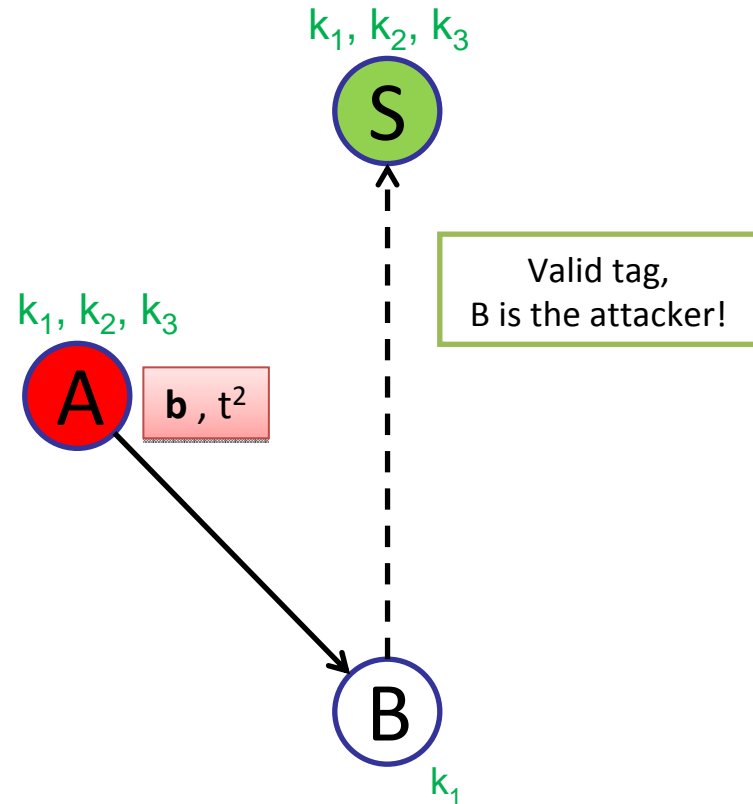
- Key idea:
Pre-distribution of source tags
- All nodes are time-sync'd
- Nodes know key release schedule of **S**
- Nodes only accept "safe" blocks

TESLA-Based Identification

o Key idea:

Non-repudiation property of TESLA

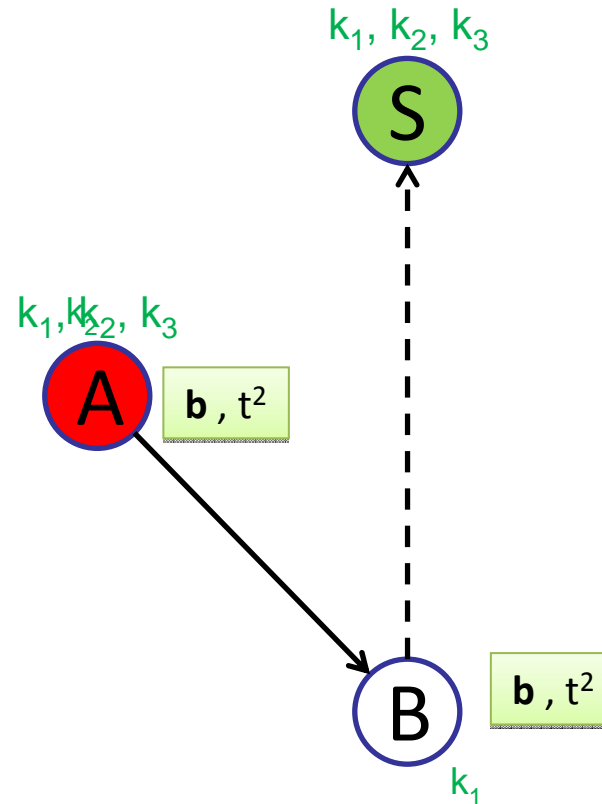
- Controller knows key release schedule of sender
- Sender sends an evidence tag
- Receiver reports evidence tag
- Tag can only be generated by sender by the time the report reaches controller



TESLA-Based Identification (cont.)

To **prevent** the sender from sending **bogus tag**:

- Sender needs to eventually release keys to make receiver accept it blocks

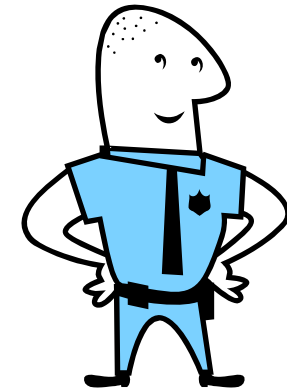


Security Guarantee

- Detection Scheme:

- q : field size
- l_1 : # detection tags
- Prob. of failed detection :

$$\frac{1}{q^{l_1}}$$



- Identification Scheme:

- h : # corrupted blocks uploaded
- l_2 : # identification tags

- Prob. of identification :

$$\sum_{i=2}^h \binom{h}{i} \left(1 - \frac{1}{q^{l_1}}\right) \left(\frac{1}{q^{l_1}}\right)^{h-i}$$

- Prob. of framing a benign sender :

$$\frac{1}{q^{l_2}}$$

Outline

1. Background and Motivation
2. Detection Scheme
3. Identification Scheme
4. Performance Evaluation
5. Conclusion



Performance Evaluation

1. Setting:

- 64 Kbps, $q=2^8$, $n=2048$, $m=128$, $l_1=l_2=3$
- 2.8 Ghz CPU, 32 GB RAM
- SpaceMac implementation in Java and C/C++

Available at <http://www.ics.uci.edu/~anhml/software.html>

2. Bandwidth Efficiency:

- Pre-distribution overhead = 1%
- Online detection overhead = 0.1%
- Online identification overhead = 0.3%

3. Computation Efficiency (C/C++):

- Detection delay = 201 μs
- Identification delay = 402 μs
- Combined delay = 603 μs



Conclusion

- A Complete Defense Mechanism for P2P Systems:

Detection + Identification

- Main building blocks:

Homomorphic MACs + TESLA

- Key properties:

- ✓ In-network detection
- ✓ Precise identification
- ✓ Arbitrary collusion resistance
- ✓ Low overhead
- Require time sync



o Questions

