Authentication in Stealth Distributed Hash Tables

Andrew MacQuire, Andrew Brampton, Idris A. Rai,
Nicholas J. P. Race and Laurent Mathy
{macquire,brampton,rai,race,laurent}@comp.lancs.ac.uk
Computing Department
Lancaster University, UK

Wednesday 30th August
Euromicro 2006

LANCASTER UNIVERSITY

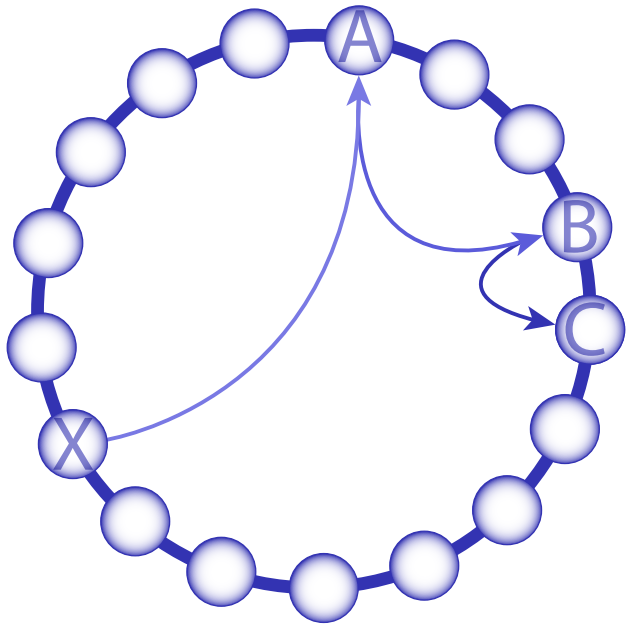
Outline

LANCASTER UNIVERSITY

What is a Distributed Hash Table?

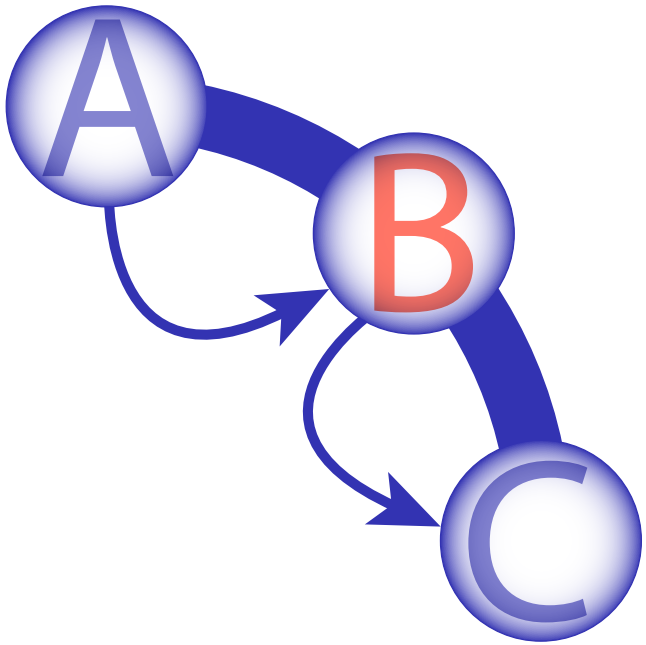
Common Characteristics

- DHTs allow data sharing amongst **numerous** machines
- Nodes and data are **consistently identified** via hash functions
- Distributed routing algorithms allow **any node** to be located
- Peer-to-peer system: typically **all nodes are treated equally**



LANCASTER UNIVERSITY

What security problems exist?

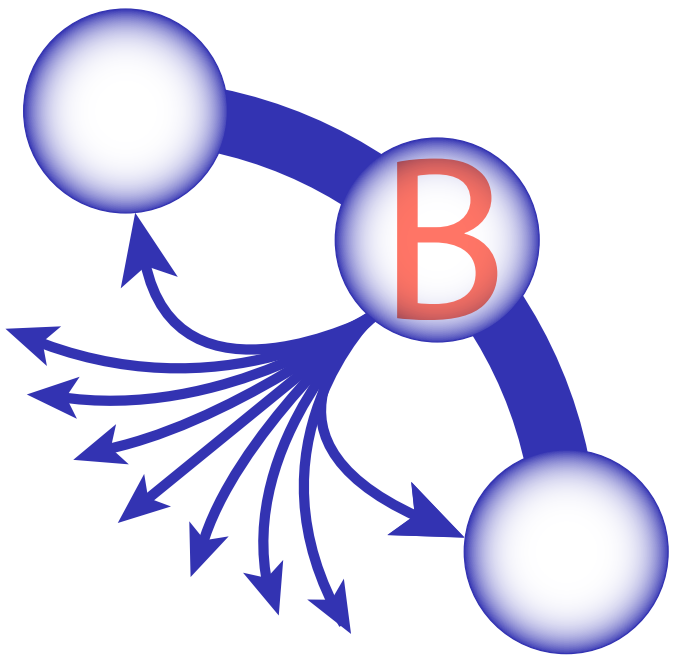


Common Attacks

- Messages are **routed through** intermediate nodes
  - *Sniffing*
  - *Man-in-the-Middle*
  - *Denial of Service*
- Nodes can inject **unwanted** messages
  - *Spoofing*
  - *Pollution*

LANCASTER UNIVERSITY

What security problems exist?



Common Attacks

- Messages are **routed through** intermediate nodes
  - *Sniffing*
  - *Man-in-the-Middle*
  - *Denial of Service*
- Nodes can inject **unwanted** messages
  - *Spoofing*
  - *Pollution*

LANCASTER UNIVERSITY

What can be done?

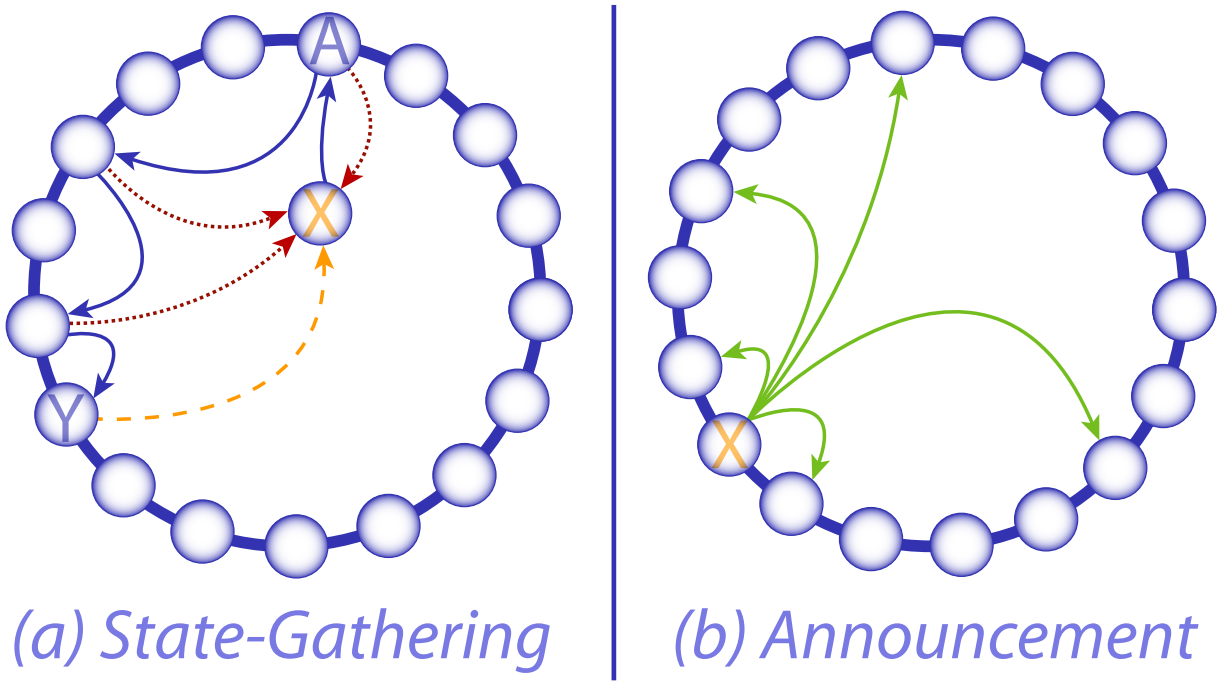
- Could simply **deny access** to untrusted nodes...
  - ...might **exclude** well-behaved nodes unnecessarily
- Better to **restrict** the operations untrusted nodes can perform
  - ...but how can nodes be **separated** into trusted and untrusted?

What can be done?

- Could simply **deny access** to untrusted nodes...
  - ...might **exclude** well-behaved nodes unnecessarily
- Better to **restrict** the operations untrusted nodes can perform
  - ...but how can nodes be **separated** into trusted and untrusted?

How can a Stealth DHT help?

- Stealth DHTs separate nodes into **Service** and **Stealth** classes
  - An altered join procedure (without phase ‘b’) **stops** Stealth nodes from being routing **intermediaries**



(a) State-Gathering *(b) Announcement*

LANCASTER UNIVERSITY

How can nodes be authenticated?

- Service node status is given to **trusted** nodes only, all others become Stealth nodes
  - ...however, a system is required to **enforce** this separation

- **Public Key Infrastructures**
  - ...allow multiple users to validate each other's identities and messages (as well as encrypt traffic) with **no prior exchange** of data

Typical PKI Components

- **Registration Authority**
- **Certification Authority**
- **Certificate Repository**
- *Revocation List?*

LANCASTER UNIVERSITY

How can nodes be authenticated?

- Service node status is given to **trusted** nodes only, all others become Stealth nodes
  - ...however, a system is required to **enforce** this separation

- **Public Key Infrastructures**
  - ...allow multiple users to validate each other's identities and messages (as well as encrypt traffic) with **no prior exchange** of data

Typical PKI Components

- **Registration Authority**
- **Certification Authority**
- **Certificate Repository**
- *Revocation List?*

LANCASTER UNIVERSITY

How can nodes be authenticated?

- Service node status is given to **trusted** nodes only, all others become Stealth nodes
  - ...however, a system is required to **enforce** this separation

- **Public Key Infrastructures**
  - ...allow multiple users to validate each other's identities and messages (as well as encrypt traffic) with **no prior exchange** of data

Typical PKI Components

- **Registration Authority**
- **Certification Authority**
- **Certificate Repository**
- *Revocation List?*

LANCASTER UNIVERSITY

How does a Stealth DHT PKI work?

- Could be a completely **external** system...
- Could also consist of one or more **internal** service nodes...
  - The *Registration* and *Certification Authorities* **must be highly trusted**
  - The *Certificate Repository* can be **any node(s)**, as certificates are immutable and digitally signed

LANCASTER UNIVERSITY

What implementation considerations exist?

- **Certification Hierarchy**
  - *Single globally trusted key?*
    - **Simple**, but problematic if the key is ever **compromised**
  - *Hierarchy of keys?*
    - More **complex**, but allows for finer-grained **control**
    - **Highest-level** keys can be kept secure and used **rarely**

What implementation considerations exist?

- **Permissions Management**
  - *Permissions within certificates?*
    - **Simple**, but **increases** message size
    - Does **not** require an extra lookup
    - Difficult to alter; certificates are **immutable**
  - *Hold permissions on dedicated service node(s)?*
    - **Complex**, yet **flexible**
    - **Increases** messaging overhead through lookups

LANCASTER UNIVERSITY

What implementation considerations exist?

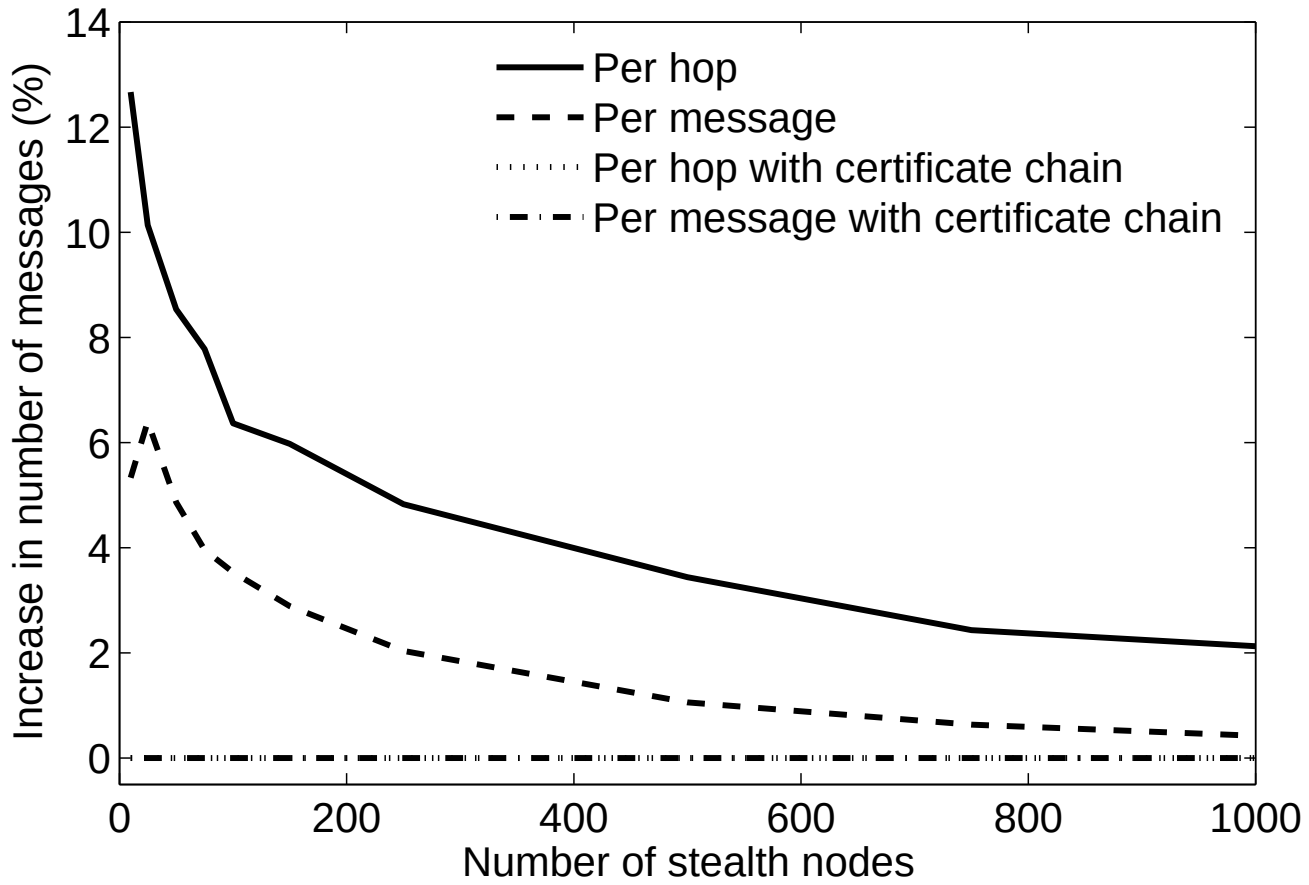
- **Certificate Revocation**
  - *Short expiry times?*
    - Nodes are **re-issued** certificates periodically; unwanted certificates are **not** re-issued
    - **Tradeoff** between certification overhead and response time
  - *Certificate Revocation List?*
    - List stored/replicated across dedicated service node(s)
    - Polled periodically or per-authentication operation; **another tradeoff** between overhead and response time

LANCASTER UNIVERSITY

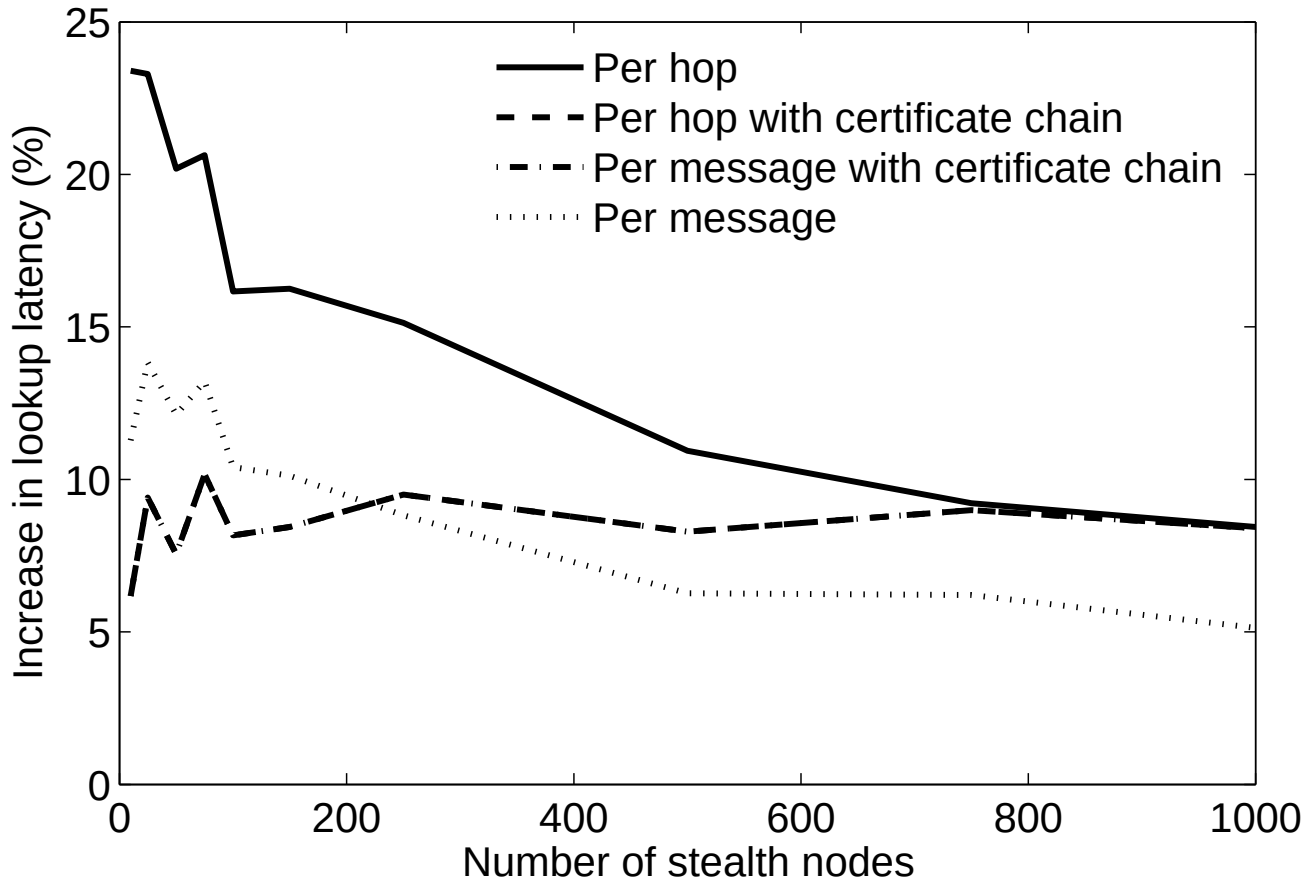
What implementation considerations exist?

- **Authentication Granularity**
  - *How often are messages validated?*
    - Per-**hop**?
    - Per-**message**?
    - Per-**session**?
  - *Are complete certificates chains included in messages?*
    - **Increased** message size, but **no extra messaging** required

Percentage increase in number of messages...



Percentage increase in lookup latency...



Conclusion

- Stealth DHTs were originally proposed to separate **powerful from less powerful** nodes
- This distinction can be extended to **trusted and untrusted** nodes
- By **limiting** the operations stealth nodes can perform, untrusted nodes **can still be serviced**
- By **enforcing** the separation through a **PKI**, Stealth DHTs can supply a **secure, resilient overlay**

LANCASTER UNIVERSITY

Thank you for listening.

Any questions?

Authentication in Stealth Distributed Hash Tables

Andrew MacQuire, Andrew Brampton, Idris A. Rai,
Nicholas J. P. Race and Laurent Mathy
{macquire,brampton,rai,race,laurent}@comp.lancs.ac.uk
Computing Department
Lancaster University, UK

Wednesday 30th August
Euromicro 2006

LANCASTER UNIVERSITY