UCLA

# Revere—Disseminating Security Updates at Internet Scale

**Jun Li**

CS 239

May 29, 2002

1

## Motivation

Revere

- Threats propagate quickly through the Internet
  - Viruses, worms, Trojan horses, etc.
  - e.g., Code Red worm
- Critical security info throughout the Internet is often stale
  - Victims lack up-to-date knowledge of new threats
- We must react at the same speed

2

**introduction** RBone dissemination security measurement conclusions

## Goal of Revere

Revere

- To disseminate security updates throughout the Internet quickly, securely, and with high assurance
  - Early-warning signal
  - Virus signature
  - Intrusion detection event
  - Certificate revocation list
  - Offending characteristics recorded at firewall
  - Security patches
  - . . . . . .

3

**introduction** RBone dissemination security measurement conclusions

## Challenges

Revere

- Fast
  - Must not be slower than the propagation of threats
- Secure
  - Revere is a tempting target for attackers
  - A corrupted Revere can be misused or abused
- Resilient
  - Interruption threats by compromised nodes, or any kind of failure
  - Cryptography does not assure delivery
  - Authenticated acknowledgements are insufficient
- Scalable
  - Any Internet host is a potential recipient
  - Node disconnection will be common

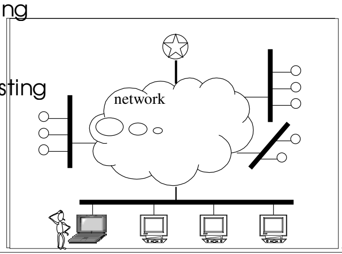
4

**introduction** RBone dissemination security measurement conclusions

## Simple Transmission Techniques

Revere

- Unicasting
- Broadcasting
- Flooding
- IP multicasting




5

**introduction** RBone dissemination security measurement conclusions

## Virus Signature Distribution

Revere

- Let users download from a website
- Set up a central server
- A naïve peer-to-peer approach

6

**introduction** RBone dissemination security measurement conclusions

Revere--Delivering Security Updates at Internet Scale

## A Government Practice

- Typically, a Federal Computer Incident Response Capability team e-mails alerts to agencies
- But when facing the "I Love You" virus, many agencies shut down their email servers
- Thus, phoned and faxed alerts instead, one at a time
  - What a time-consuming tedious procedure !
- Afterwards . . .
  - A completely automated new system is designed that claims to handle 96 phone lines and deliver 800 faxes/hour
  - They also look into an AM radio system for federal employees to check every morning

Diane Frank. "**One if by phone, two if by fax**," *Federal Computer Week*, September 2000



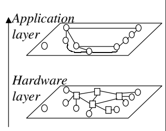
## So, How Would Revere Meet the Challenges?

- Non-centralized delivery structure
- Use redundancy to support information transmission resiliency
- Secure both the dissemination procedure and the delivery structure



## The Revere Solution

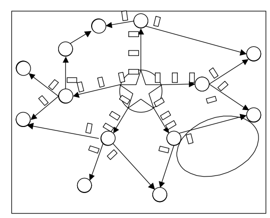
- Revere builds overlay networks, called ***RBones***, on top of the Internet
- . . . . and uses RBone to deliver security updates
  - Every node can also forward updates
  - Disconnected nodes will be handled
- Runs at application level
  - Great flexibility
  - No changes to underlying network infrastructure
  - Implemented in Java
  - Deployment is easy






## RBone: A Self-Organized Resilient Overlay Network

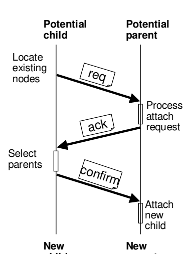
- Redundancy-based resiliency
  - Multiple delivery paths
    - Therefore multiple parents
  - Select as-disjoint-as-possible paths
- Self-organized overlay
  - Easy join
  - Easy withdrawal
  - Broken nodes
  - Broken links





## RBone Join Procedure

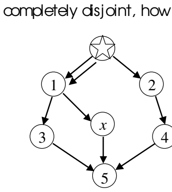
- Search for existing nodes
  - Directory service
  - Multicast-based expanding-ring or expanding-wheel search
  - Contact already-known existing nodes
- Negotiate to select best parents
  - Again, multiple parents are allowed!
- Three-way-handshake negotiation protocol
  - Reciprocal selection






## Parent Selection

- What parental qualities matter?
  - Efficiency: is the delivery via this parent fast?
  - Resiliency: is the delivery via this parent disjoint with other paths?
    - If not completely disjoint, how much is the overlap?



## Parent Selection (cont'd)

- The path vector of a node
  - Describes the fastest path:
    - Latency
    - An ordered list of nodes to cross
    - Denoted as $pv(\mathbf{n})$
- The path vector associated with a parent
  - Described the fastest path through the parent
  - Denoted as $pv(\mathbf{n}, \mathbf{p})$
- The resiliency level of a node's parent
  - Calculated by comparing the path vector associated with the parent and the path vector of the node



## Path-Vector-Based Parent Selection Algorithm

Suppose node *c* is deciding a potential parent *x*

**if** (*c* has not reached the maximum number of parents)
  select *x*;
**else if** $pv(c,x)$ is faster than $pv(c)$
  select *x*;
**else if** *resiliency(x)* better than *resiliency*(a current parent)
  select *x*;
**else**
  do not select x;



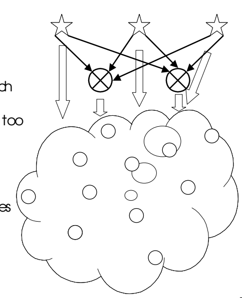
## RBone Maintenance

- Heartbeat messages
  - To verify node liveness
  - To update path info associated with every parent
    - Carry timestamps
    - Deal with the broken parent, or any broken node on a path
- Explicit messages
  - To tear down a parent-child relationship
- Corrupted security updates also trigger adjustment



## RBone with Multiple Dissemination Centers

- If a node wants to hear from multiple centers
  - it joins multiple RBones, each rooted at a different center
  - this becomes undesirable if too many centers
- Build a common RBone rooted at a rendezvous
  - Every center delivers updates to the rendezvous
- Multiple rendezvous points can be set up





## Dissemination Procedure

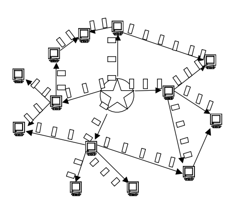
- A dual mechanism
  - Pushing as the main method for broadcasting security updates from a dissemination center
  - Pulling as the supplementary method for catching up with missed security updates
- Security update format

| type | seqno | timestamp | payload | signature |
|---|---|---|---|---|



## Pushing Security Updates

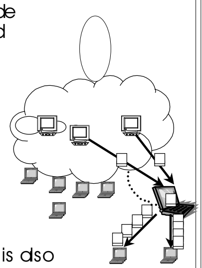
- Adaptive transmission
  - TCP
  - UDP
  - IP multicast
  - etc.
- Duplicate checking
  - Every Revere node remembers the range of historical sequence numbers
- Security checking
  - (Will be addressed later)



## Pulling Security Updates

Revere

- By the time a disconnected node reconnects, it may have missed some security updates
  - Parents do not keep old updates
  - Parents might no longer be parents
  - Retransmission by the dissemination center is not scalable
- Repository servers
  - Nodes that keep old security updates
  - Usually maintain stable connection
  - Clients directly contact those servers
- A newly pulled security update is also forwarded to child nodes, if any



19

introduction RBone **dissemination** security measurement conclusions

## Security Assumptions

Revere

- Center's public key is wellknown
- Large percentage of nodes are cooperative
- Any node could be corrupted
  - The center cannot be corrupted, but its private key could be compromised
- No uniform security scheme to protect node-to-node control messages
  - For example, some nodes may run the *Kerberos* service to authenticate other nodes, some may employ public-key-based authentication

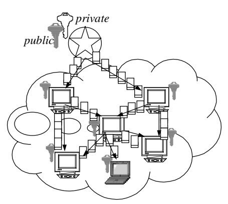
20

introduction RBone dissemination **security** measurement conclusions

## Securing the Dissemination Process

Revere

- Integrity of security updates
  - A dissemination center has a public/private key pair
  - Every security update carries a digital signature signed by the center
- Availability of security updates
  - Redundant delivery



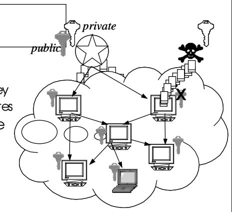

21

introduction RBone dissemination **security** measurement conclusions

## Center Key Disclosure

Revere

- Disastrous if the private key of a center is disclosed
  - The public key must be invalidated
- Public key invalidation
  - Send a key invalidation message
    - Signed with the disclosed private key
    - Delivered in the same way as updates
  - Every recipient verifies the message with the current public key
    - Then discards this public key
    - And switches to the new public key
- How secure is this method?
  - Fine, if an attacker also distributes key invalidation messages
  - Resilient, since it follows the same routes as normal security updates




22

introduction RBone dissemination **security** measurement conclusions

## Securing RBone

Revere

- Every node can enforce several different security schemes
  - Node authentication
  - Message filtering
  - Etc. . . .
- The functionality of a specific security scheme can be easily plugged in
- Node-to-node communication is initiated with security scheme negotiation

23

introduction RBone dissemination **security** measurement conclusions

## Metrics

Revere

- √ RBone maintenance bandwidth
- √ Dissemination bandwidth
- Join bandwidth
- Join latency
- Dissemination latency
- Dissemination resiliency

25

introduction RBone dissemination security **measurement** conclusions

## What's the challenge ?

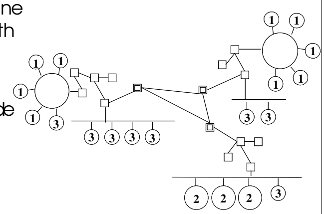
- Revere is a large-scale distributed system
- Empirical experiments incur prohibitive cost
  - Required to obtain, access, configure, maintain, and collect results from more than a few hundred machines
- Simulation is more scalable, but
  - Expensive to develop
  - Slow to run
  - Possibly inaccurate (hidden costs and subtle timing effects) & buggy
  - Must be validated against real system

26



## The "Overloading" Approach

- A physical machine is overloaded with multiple (virtual) Revere nodes
- Each Revere node runs the real software
- Achieves larger scalability using multiple machines




Jun Li, Peter Reiher, Gerald Popek, Mark Yarvis, and Geoffrey Kuenning. "**An approach to measuring large-scale distributed systems**," ***TestCom** 2002*, Berlin, Germany, March 2002.

27



## Measurement Environment

- A testbed of 10 machines
  - Overloaded with up to 3,000 Revere nodes
- Topology
  - GT-ITM topology generator
  - A topology server for node assignment
- Configuration
  - Every node must have 2 parents, but ≤ 10 children

29



## Join Performance

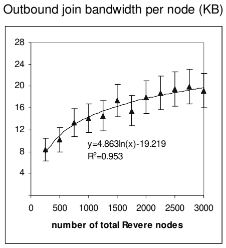
- Join phase
  - Token-controlled resource-locking mechanism
  - One-at-a-time join
  - No contention because of serialization

30

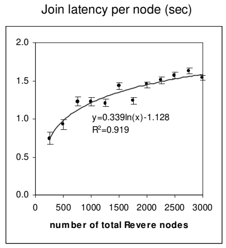


## Join Performance

Outbound join bandwidth per node (KB)




Join latency per node (sec)




31



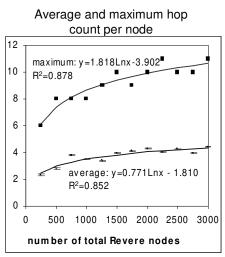
## Dissemination Speed

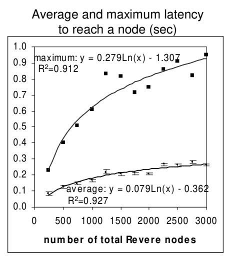
- Measured in dissemination phase
- Dissemination Latency =
  kernel-crossing latency +
  processing latency +
  communication latency
- Measured using divide-and-conquer method
  - Measure each subtask in non-overloaded environments
  - Measure hop counts in full system, and then sum

32

## Dissemination Speed (cont'd)

Revere



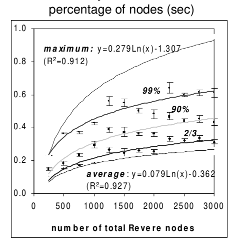

34

introduction RBone dissemination security **measurement** conclusions

## Dissemination Speed (cont'd)

Revere

The latency to reach a certain percentage of nodes (sec)




Trends:
[99%] y=0.156ln(x)-0.628 ($R^2$=0.881)
[90%] y=0.136ln(x)-0.629 ($R^2$=0.913)
[ 2/3 ] y=0.098ln(x)-0.463 ($R^2$=0.914)

- Assuming the trends continue at higher scale, then in a 100M-node RBone:
  - average hop counts: 12
  - maximum hop counts: 30
  - average latency: 1.10s
  - latency to reach 2/3: 1.34s
  - latency to reach 90%: 1.88s
  - latency to reach 99%: 2.25s
  - latency to reach all: 3.83s

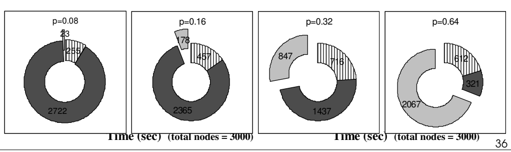
35

introduction RBone dissemination security **measurement** conclusions

## Dissemination Resiliency

Revere

- Resiliency test phase
  - Every node assigned a uniform probability of failure
  - Measured a 3000-node RBone with maintenance disabled
- Results:
  - All reachable with less than 2% failure probability
  - Still very resilient with higher failure probability




36

introduction RBone dissemination security **measurement** conclusions

## Some Related Work

Revere

- RON – resilient overlay network
  - Inserts an overlay network layer between routing and application
  - Allows faster routing failure recovery & application-specific routing
- Other overlay networks
  - Tree-structured dissemination is not resilient
  - Nodes are not always connected at delivery time
  - Security handling is not sufficient
- Multi-path routing
  - A router-level implementation
    - Primarily for load balancing or congestion control
  - Must handle security issues at router level
    - Replay prevention, key distribution . . .
  - Deployment is challenging

37

introduction RBone dissemination security measurement **conclusions**

## Work Summary of Revere Project

Revere

- Designed
  - The structure, the dissemination, the security, the . . .
- Implemented
  - 45,010 lines of Java code in the prototype system
- Measured
  - The number of nodes varies from 250 to 3,000
- Demonstrated
  - DARPA Site Visit
  - UCLA Annual Research Review
- Published and presented
  - NSPW'99, NISSC'99, Testcom'02
  - Also submitted to OSDI'02
  - Dissertation draft is at your hand

38

introduction RBone dissemination security measurement **conclusions**

## Conclusions

Revere

- **Necessary work**: Since attackers already distribute malicious functions rapidly, an even faster notification system is required.
- **Encouraging results**: It is feasible to disseminate security updates to much of today's connected Internet quickly, securely, and with high assurance.
- **Broad applicability**: Revere is not limited to only security updates.

39

introduction RBone dissemination security measurement **conclusions**