Trusted Disk Loading in the Emulab Network Testbed

Cody Cutler, Mike Hibler, Eric Eide, Rob Ricci

Emulab

- Public network testbed
- Create complex experiments quickly
- 500+ nodes at Utah Emulab

Emulab Nodes

- Physical nodes
- Users have root
- Space/time shared

Artifacts from previous experiment may persist on node

Node Corruption

Why Reset State?

- Experiment fidelity depends on a fresh start
- “Contaminated start” is unacceptable for security sensitive experiments

Disk Reloading

- Control server forces reboot and directs node re-imaging over network
- Disk reloading network is shared with other nodes

In current system state reset is not guaranteed and is not tamper-proof

Goals

- Disk reloading must be reliable
- Must be flexible for many boot paths
- Must scale to size of testbed

Solution: Trusted Disk Loading System (TDLS)

If the experiment is created successfully, node state is reset

Contributions

- Design and implementation of secure disk loading protocol
- Flexible and secure reloading software scalable to size of testbed

Node Reloading

download

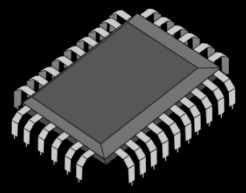
boot ROM

TDLS Fundamentals

- Establish trust
- Verify every stage of node reloading with control server

Approach: use the Trusted Platform Module

Trusted Platform Module (TPM)



- Secure key storage
- Measurement
- Remote attestation (quotes)

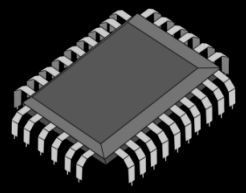
Secure Key Storage

- Keys are always encrypted before they leave the TPM
- Keys are only usable on the same TPM where they were created
- Control server can identify nodes by the public portion of these keys

TDLS Fundamentals

- ✓Establish trust
- Verify every stage of node reloading with control server

Trusted Platform Module (TPM)



- Secure key storage
- Measurement
- Remote attestation (quotes)

Measurement

- Measuring is when we hash a region of memory and extend a certain PCR with the resulting hash

- Platform Configuration Registers (PCR)
  - TPMs generally have 24 PCRs
  - Holds a hash
  - PCRs can only be modified through extension
  - Extending:

PCR = hash(previous value of PCR + a new hash)

Secure Boot Chain with TPM

1. Immutable part of BIOS measures the rest of BIOS
2. BIOS measures boot device
3. Boot device then measures whatever it loads
4. etc.

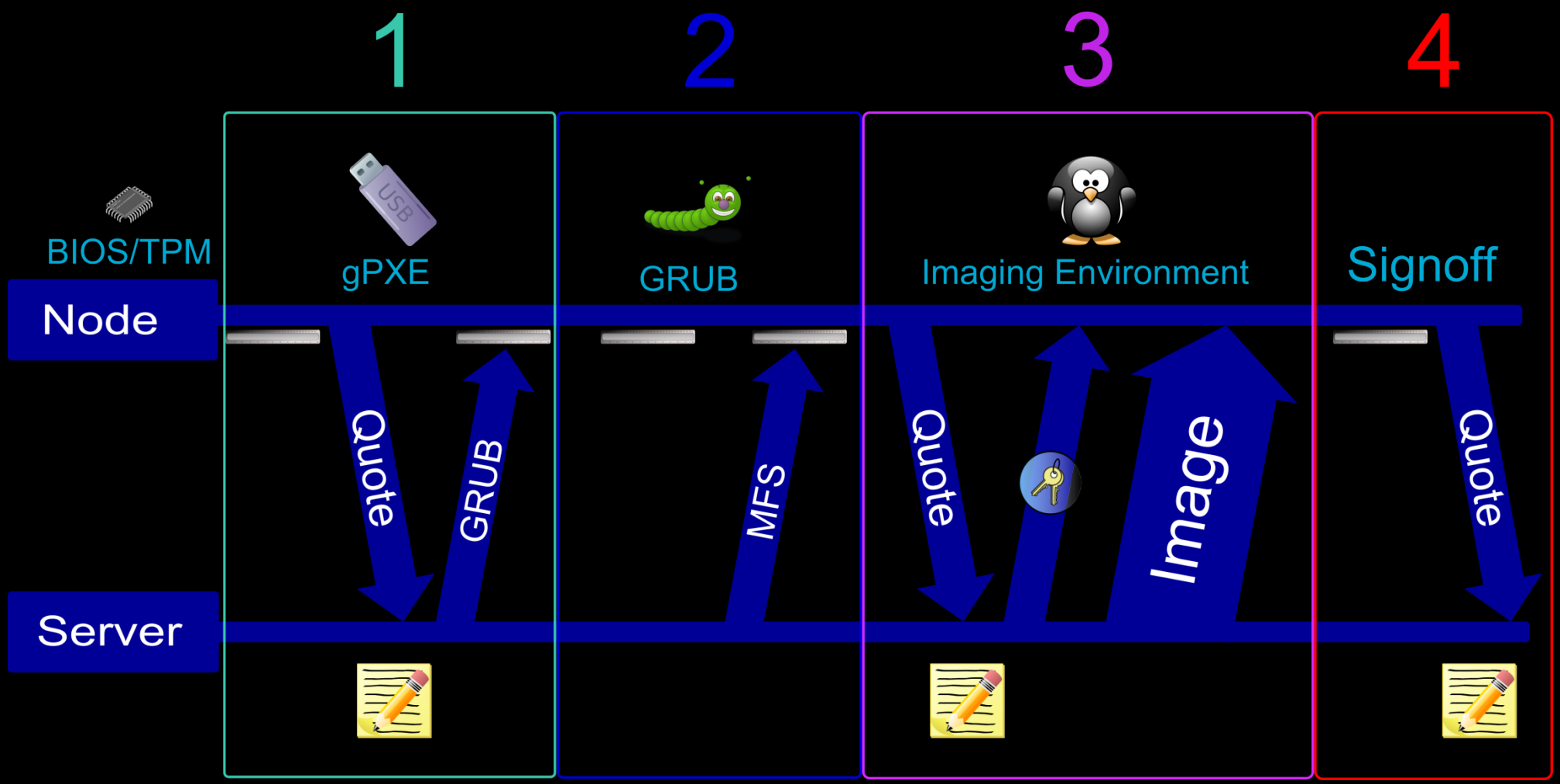
Remote Attestation

- TPM packages up the desired PCRs and signs them
  - This is called a quote
- Tamper-proof as it is signed by the TPM
- Very easy to differentiate between a genuine quote and arbitrary data signed by TPM

TDLS Fundamentals

- ✓Establish trust
- ✓Verify every stage of node reloading with control server

TDLS Reloading

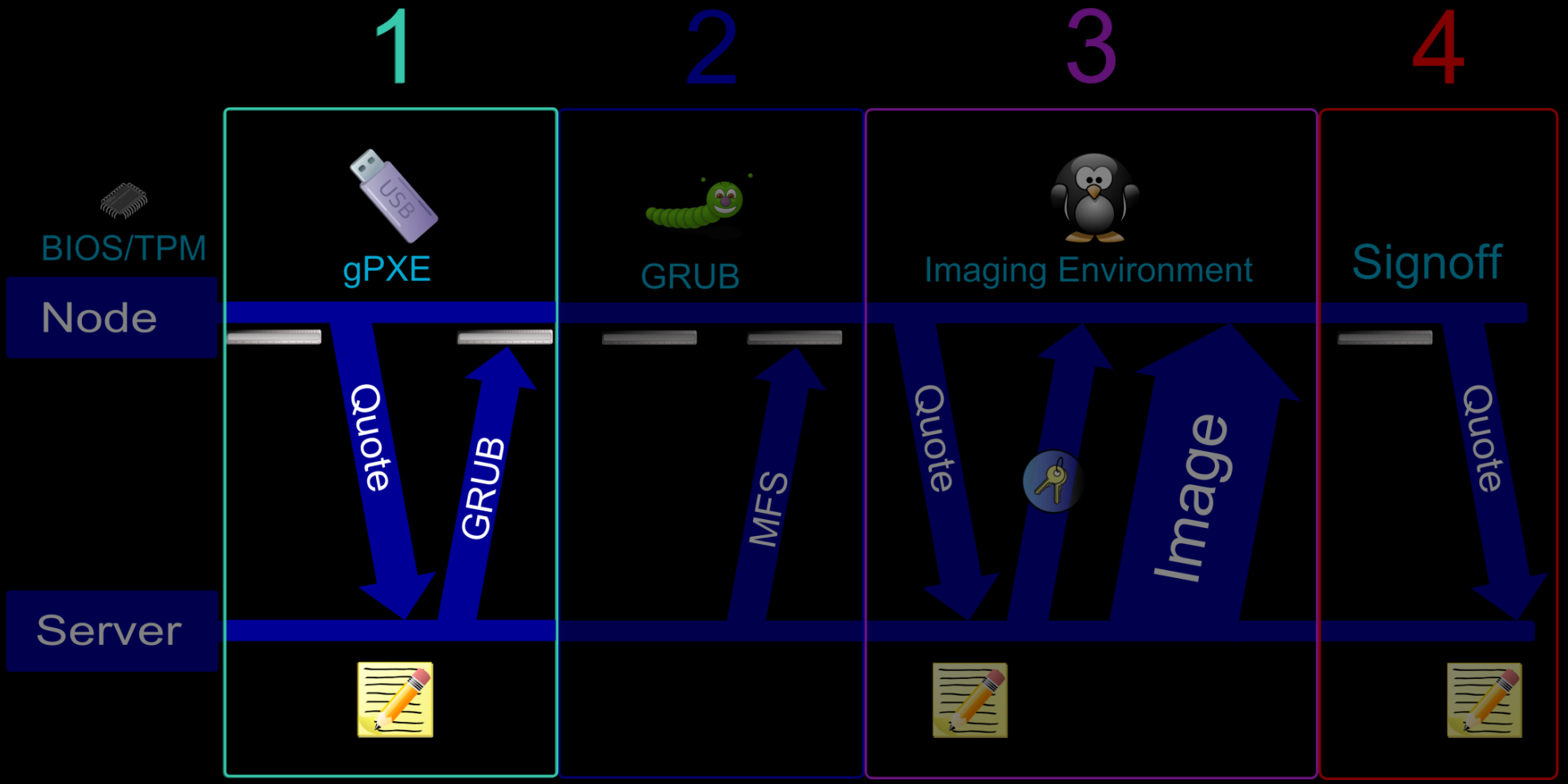


Stage 1: PXE Boot

- PXE is a network boot protocol
- PXE ROMs aren’t TPM aware
- PXE ROMs won't check-in with the control server

Boot to USB dongle with gPXE

Stage 1: gPXE



- Measured by BIOS
- Embedded certificate for server authentication
- Sends a quote to control server

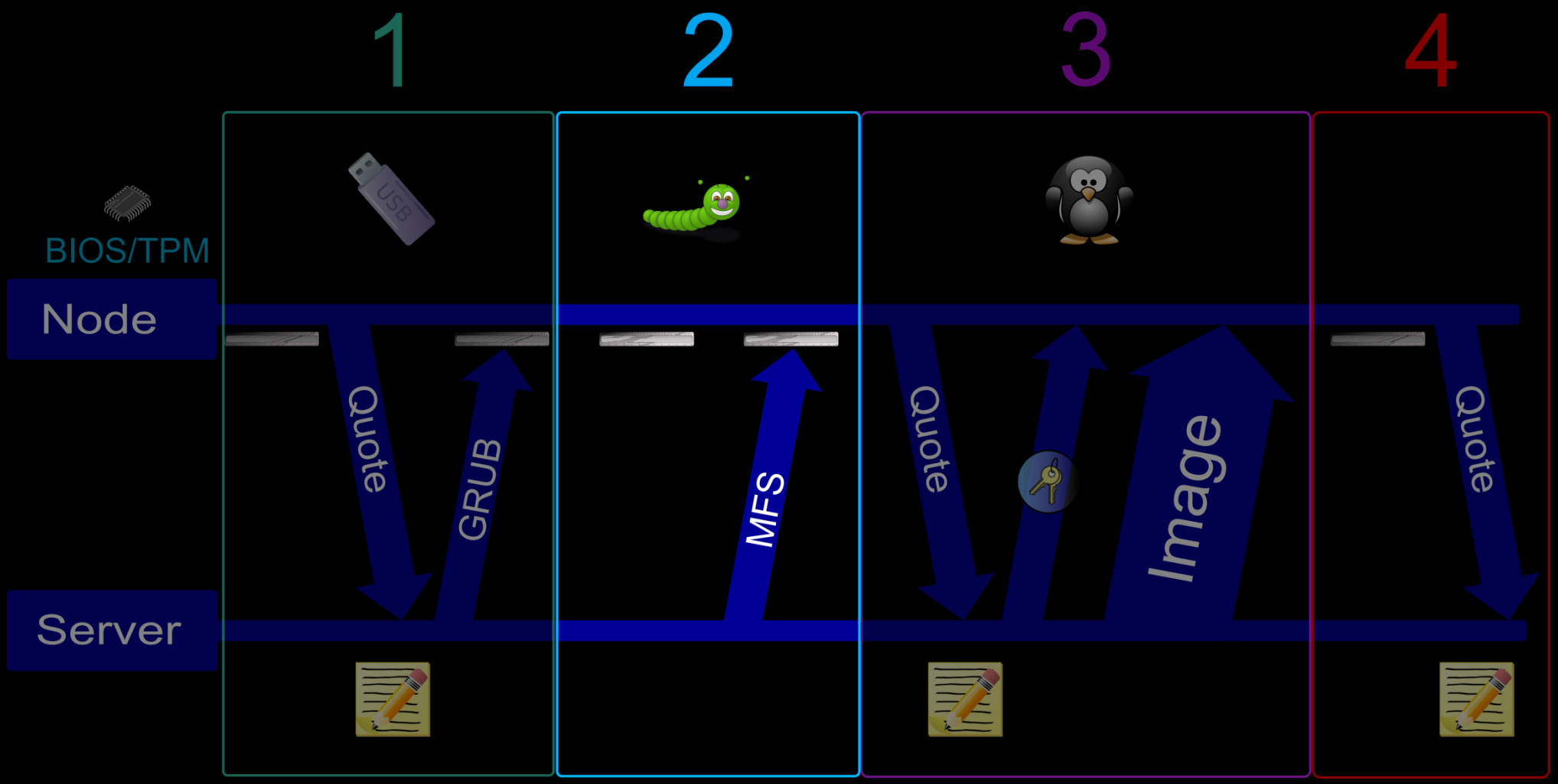
Checking Quotes

- Server compares every PCR in the quote with known good values in the database
- The TPM signature over the quotes is verified
- Quotes contain a nonce from the server to guarantee freshness
- Different stages are measured into different PCRs

Incorrect Quotes

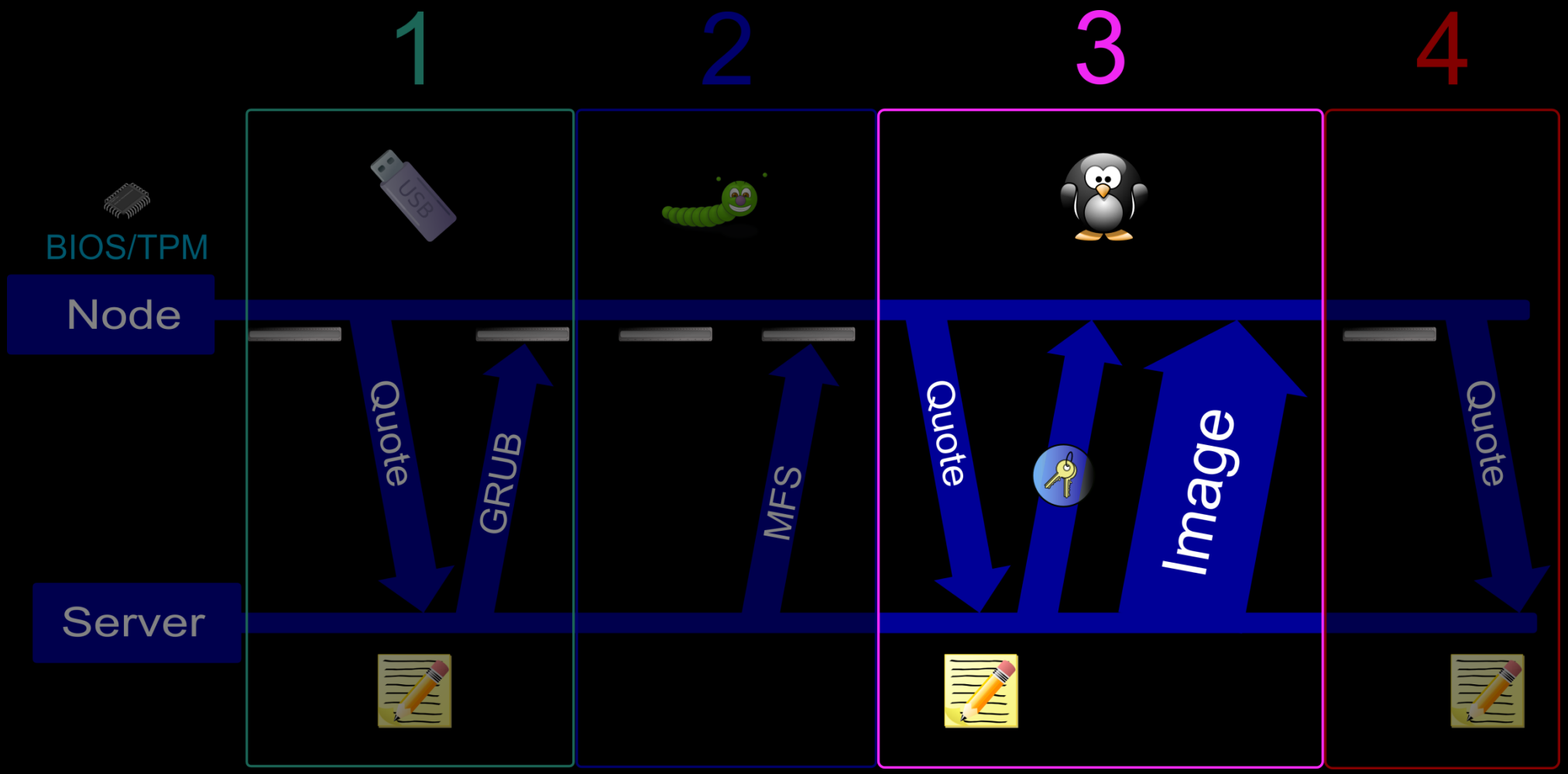
- An incorrect PCR means something was modified
- Failure to send a quote before a timeout is treated as a verification failure
- Control server cuts power to the node and quarantines it

Stage 2: GRUB



- Retrieves, measures, and boots the imaging MFS
- Will boot to disk when necessary

Stage 3: Imaging MFS

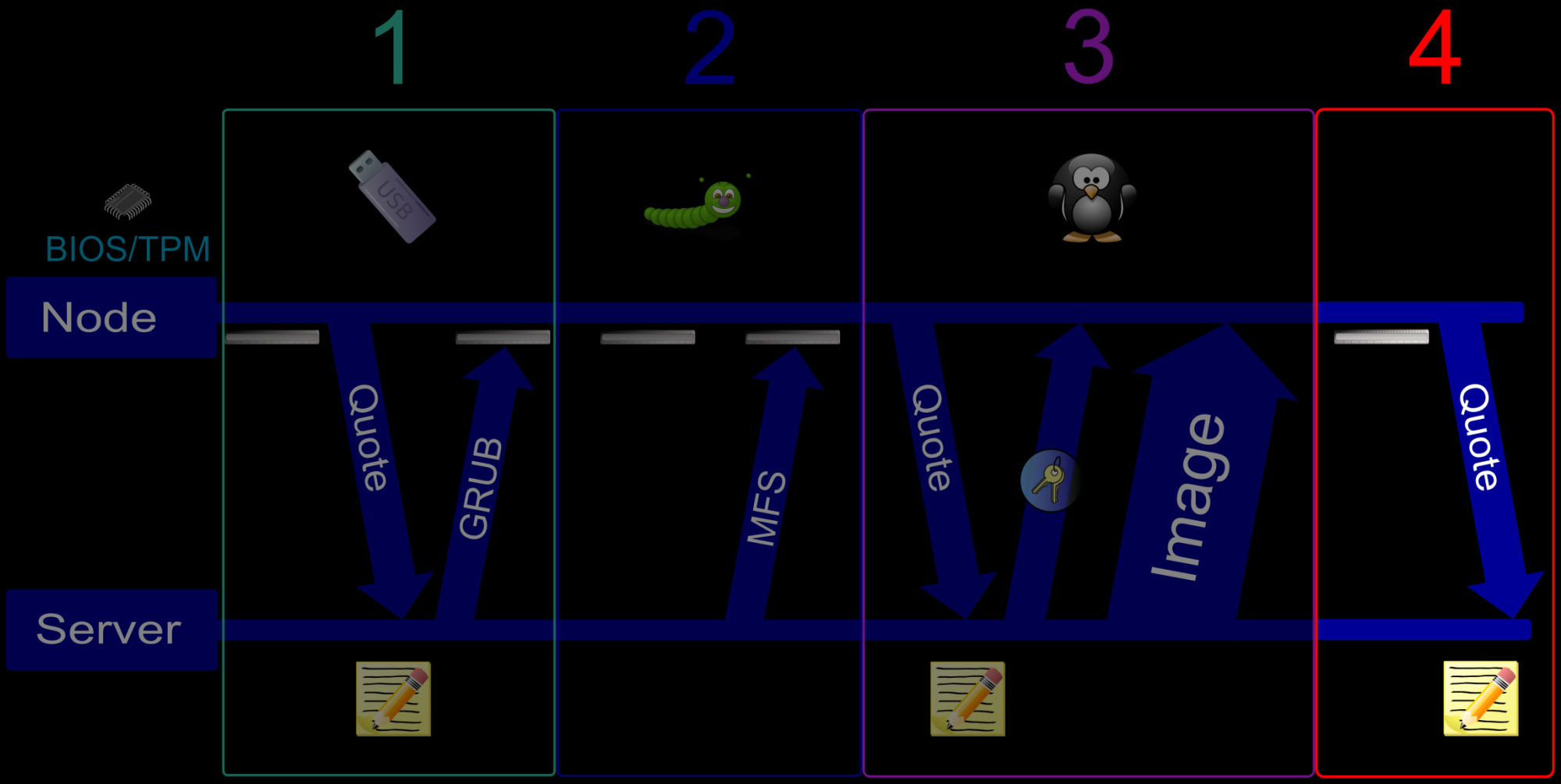


- Sends quote covering everything
- Writes the encrypted image to disk

Sensitive Resources

- Control server closely monitors a node's progress via quotes
- A node can only receive sensitive resources (decryption keys) in a particular state

Stage 4: Signoff



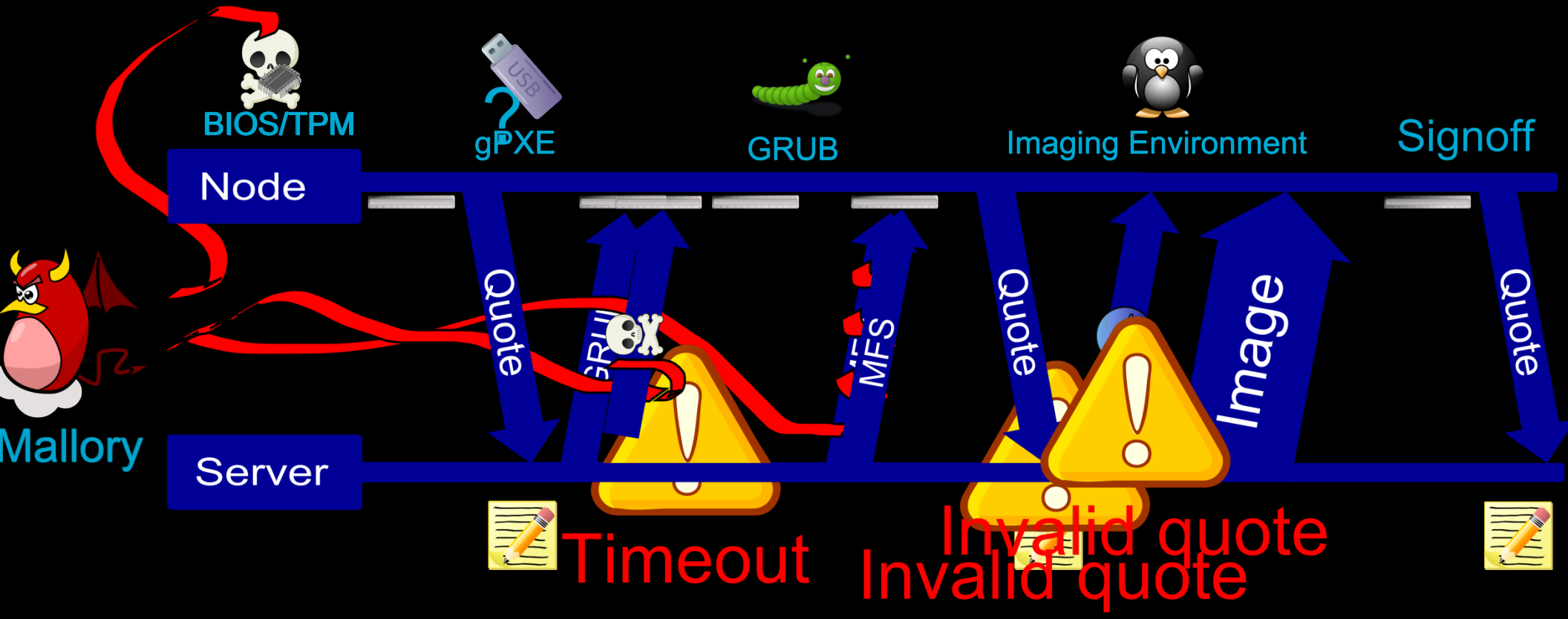
- Disk is imaged
- Extends known value into designated reboot PCR
- Marks the end of the trusted chain

Image Security

- The TDLS writes the user-chosen disk image on a node
- Security researchers want to use both secure and insecure images
- By design, the TDLS does not check the user-chosen image

Attacks That Will Fail

- Any boot stage corruption
- BIOS code or configuration modifications
- Injecting new stages



What this means

We win

Summary

- Node state must be fully reset in a secure way
  - Some testbed properties make this very difficult
- Trusted Disk Loading System
  - Tracks node progress with quotes
  - Guarantees node state is reset
  - Leveraging the Trusted Platform Module
    - Establish trust between the node and server
    - Verify every stage of boot chain
- If experiment creation succeeds the disk has been securely reloaded

Future Work

- Refine the violation model
- Integrate with Emulab UI
- Deploy on 160 TPM-enabled nodes at Utah
- Enable experimenters to verify node state

Questions?

ccutler@cs.utah.edu

http://www.emulab.net