Titan: enabling a transparent silicon root of trust for Cloud
Talk outline

01 Motivation and problem statement
02 System View and Integration
03 Chip Architecture
04 Feature Deep Dives
05 Building a community: Open Titan?
Motivation and architecture
The problem:
Example 1: How do we know it is our equipment?

Russian hackers found the 'ultimate' hacking tool buried in the supply chain of laptops.
Solution:
Tag and verify every device
Example 2:
Can we trust our boot chain?

BETRAYING THE BIOS:
WHERE THE GUARDIANS OF THE BIOS ARE FAILING
Solution:
Sign and verify all boot code
Conclusion: We need a silicon root of trust
Cloud security properties

Every element in the datacenter should be securely identifiable:
cryptographic attestation
Cloud security properties

1. Trusted Machine Identity
2. First Instruction Integrity
3. Tamper-evident logging
4. Trusted implementation

The first code executed should be trusted: cryptographically signed and verified firmware, live monitored for protection.
Cloud security properties

1. Trusted Machine Identity
2. First Instruction Integrity
3. Tamper-evident logging
4. Trusted implementation

All activities in the datacenter should be monitored and logged in a tamper resistant manner.
Cloud security properties

1. Trusted Machine Identity
2. First Instruction Integrity
3. Tamper-evident logging
4. Trusted implementation

Own and/or verify every piece of the stack from transistors up to critical firmware
Chip Requirements

- On-chip verified boot
- Cryptographic identity & secure mfg
- Boot Firmware signature check + monitor
- Silicon physical security
- Transparent development, full-stack
System View and Integration
Titan system integration

- CPU
- Chipset
- TITAN
- Boot FW flash

Connections:
- CPU to Chipset: SPI
- Chipset to TITAN: SPI
- TITAN to Boot FW flash: SPI

Subsystems:
- Memory subsystem
- Storage and networking subsystem
- Reset and power control
Titan system integration

CPU → Chipset → TITAN → Boot FW flash

Memory subsystem

Storage and networking subsystem

Reset and power control

Requests first boot instruction
Titan system integration

- CPU
  - Memory subsystem
- Chipset
  - Storage and networking subsystem
- TITAN
  - Reset and power control
- Boot FW flash
  - Contains (signed) boot code

- SPI connections: PCH / BMC to CPU, Chipset, TITAN, Boot FW flash
Titan system integration

- CPU
- Chipset
- PCH / BMC
- Memory subsystem
- Storage and networking subsystem
- TITAN
- Boot FW flash

Authenticates firmware, releases system reset
Reset and power control
Titan system integration

- CPU
- Chipset
- PCH / BMC
- TITAN
- Boot FW flash
- Memory subsystem
- Storage and networking subsystem
- Reset and power control
- Continuous monitoring for illegal activity
Titan system integration

CPU

Chipset

PCH / BMC

TITAN

Boot FW flash

Available for cryptographic attestation and logging

Memory subsystem

Storage and networking subsystem

Reset and power control
Chip architecture
What is Titan?

- Secure low-power microcontroller designed with cloud security as first-class consideration
- Not just a chip, but the supporting system and security architecture + manufacturing flow
Why make our own?

Implementation transparency
Complete ownership, auditability, build local expertise

Agility & velocity
Technology changes, new risk vectors arrive

No existing solutions
Vendor-agnosticity, custom features
## Glossary: a quick security chip primer

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES</td>
<td>Symmetric (shared-key) crypto algorithm</td>
</tr>
<tr>
<td>alert</td>
<td>Security critical event</td>
</tr>
<tr>
<td>BIST</td>
<td>Built in self test</td>
</tr>
<tr>
<td>BL</td>
<td>Boot loader</td>
</tr>
<tr>
<td>CA</td>
<td>Certificate authority</td>
</tr>
<tr>
<td>device state</td>
<td>Temporal state in life cycle of device (test, production, return for test, end of life)</td>
</tr>
<tr>
<td>EC</td>
<td>Elliptic curve: modern crypto algorithm</td>
</tr>
<tr>
<td>HMAC</td>
<td>Hash message authentication code</td>
</tr>
<tr>
<td>I2C</td>
<td>Two-pin low-speed peripheral interface</td>
</tr>
<tr>
<td>key mgr</td>
<td>Management of key and secret storage</td>
</tr>
<tr>
<td>NMI</td>
<td>Non-maskable interrupt</td>
</tr>
<tr>
<td>OTP</td>
<td>One-time programmable (fuse) memory</td>
</tr>
<tr>
<td>PCH</td>
<td>Intel Platform Controller Hub</td>
</tr>
<tr>
<td>PMU</td>
<td>Power Management Unit</td>
</tr>
<tr>
<td>RC</td>
<td>Resistor/capacitor clock circuit</td>
</tr>
<tr>
<td>RSA</td>
<td>Circa 1980s crypto algorithm</td>
</tr>
<tr>
<td>RTC</td>
<td>Real Time Clock</td>
</tr>
<tr>
<td>SHA</td>
<td>Hashing algorithm</td>
</tr>
<tr>
<td>SPI</td>
<td>4+ pin peripheral interface</td>
</tr>
<tr>
<td>TRNG</td>
<td>True random number generator</td>
</tr>
</tbody>
</table>
Titan specifications

- **Embedded 32b processor**
  - 8kB ROM
  - 64kB SRAM
  - 512kB Flash
  - 1kB OTP (Fuse)

- **Memory**

- **Peripherals**
  - EC/RSA crypto
  - AES/SHA/HMAC
  - Key manager
  - TRNG
  - Timers
  - USB 1.1
  - UART
  - SPI mstr/slv
  - I2C mstr/slv
  - GPIO

- **Defenses**
  - Shield
  - Temp sense
  - Volt sense
  - Device state
  - Alert resp

- **Testability / MFGability**
  - Jitter RC
  - Timer RC
  - Low speed RC

- **Debug ports**

- **Test ports**

- **Muxable data ports**

---

Google Cloud
Titan specifications

32b microcontroller core

- Boot ROM
- Flash for instr + data
- SRAM scratchpad
- One-time programmable fuses

Embedded 32b processor

Memory

- 8kB ROM
- 64kB SRAM
- 512kB Flash
- 1kB OTP (Fuse)

Peripherals

- EC/RSA crypto
- AES/SHA/HMAC
- Key manager
- TRNG
- timers
- USB 1.1
- UART
- SPI mstr/slv
- I2C mstr/slv
- GPIO

Defenses

- Shield
- Temp sense
- Volt sense
- Device state
- Alert resp

Testability / MFGability

- Test ports
- Low speed RC
- Timer RC
- Jitter RC

Muxable data ports
Titan specifications

Cryptographic acceleration
Key management + storage
Random number generator

Embedded 32b processor

Peripherals:
- EC/RSA crypto
- AES/SHA/HMAC
- Key manager
- TRNG
- Timers

Memory:
- 8kB ROM
- 64kB SRAM
- 512kB Flash
- 1kB OTP (Fuse)

Testability / MFGability:
- Jitter RC
- Timer RC
- Low speed RC

Defenses:
- Shield
- Temp sense
- Volt sense
- Device state
- Alert resp

Peripherals:
- USB 1.1
- UART
- SPI mstr/slave
- I2C mstr/slave
- GPIO

Muxable data ports:
- Test ports

Google Cloud
Titan specifications

Peripheral controllers
  Multipurpose IO
Custom Google features
Titan specifications

- Physical defenses
- Live status checking
- Hardware security alert response

**Memory**
- 8kB ROM
- 64kB SRAM
- 512kB Flash
- 1kB OTP (Fuse)

**Peripherals**
- EC/RSA crypto
- AES/SHA/HMAC
- Key manager
- TRNG
- timers
- GPIO
- USB 1.1
- UART
- SPI mstr/slave
- I2C mstr/slave

**Defenses**
- Shield
- Temp sense
- Volt sense
- Device state
- Alert resp

**Testability / MFGability**
- Test ports
- Jitter RC
- Timer RC
- Low speed RC

**PMU**
- Muxable data ports

**Test ports**
- Muxable data ports

**Physical defenses**
- Live status checking
- Hardware security alert response
Feature Deep Dives
Verified Boot
Verified boot within Titan

- Each stage verifies the next
- Earlier stages do security settings, lock out further access
- Permission levels drop at each stage, protecting critical control points
- Splitting flash code into banks allows two copies: live-updatable
- Code signing taken seriously; multiple key holders, offline logs, playbooks
1. Test logic (LBIST) and ROM (MBIST); if fail ⇒ stay in reset; else jump to ROM
2. Compare bootloader (BL) versions A + B; choose most recent
3. Verify BL signature; if fail, retry with other BL; if fail, freeze
4. Compare firmware application (FW) versions A + B; choose most recent
5. Verify FW signature; if fail, retry with other FW; if fail, freeze
6. Execute successfully verified FW
Trusted identity
Trusted chip identity

- Establish trust at manufacturing
- Each tested device uniquely identified (personalized)
  - Assigned a serial number, unique but not secret
  - Self-generates a cryptographically strong Identity Key
- Identity registered in off-site secure database
- Parts shipped, put onto datacenter devices for production
- Parts available for “attestation”, proof that they are ours
Key manager creates chip identity key

- Dedicated hardware execution
- Processor walks FSM commands
- Keys inaccessible to processor
- Identity = crypto_hash of partial secrets
  - Each comes from a different silicon technology
  - Requires attackers to defeat each
- Export enabled if FSM complete
- Export disabled after manufacture
Trusted identity (registration)

- Personalization firmware loaded
- Chip creates identity message
- Identity exported to registry via secure channel
- Identities signed by offline certificate authority
- Certificate available for installation
- Identity available for later query
Life cycle tracking using OTP Fuses

- After manufacturing, must continue to guarantee authenticity
- Define six stages, and what is enabled in each stage
  - **Raw:** no features enabled, deters wafer theft
  - **Test:** enable test features only, no production features
  - **Development:** enable production-level features for lab bringup
  - **Production:** final production features, no testability, unique keys
  - **RMA (return for test):** re-enable testability, no more production
  - **RIP:** after RMA or mfg failure, permanently disable device
- Burnable fuses track life cycle from manufacturing to production
- Each stage transition a one-way street
Life cycle tracking using OTP Fuses

Burn fuse

RAW → MFG Test → PROD → DEV → RMA → RIP
First instruction integrity
First instruction integrity

- Titan interposes on SPI, between host and system firmware Flash
- At system reset, does signature check of FW
  - Signature OK ⇒ enables system
  - Signature fail ⇒ alerts of failure
- Live monitoring
  - Snoops SPI for illegal activity
  - Unauthorized actions converted to harmless commands
SPI interposition

The challenges of SPI interposition

- Vendor agnostic requires flexibility
- SPI does not have flow control
- Passthrough latency must be minimized
- Chip & board timing a challenge
- Can affect boot latency

Outgoing SPI bus to flash

Incoming SPI bus from host

Safe command

Snoop / control logic
Physical and tamper-resistant security
Physical security & countermeasures

Anti-glitch / anti-tamper mechanisms

- Attack detection (glitch, laser, thermal, voltage)
- Fuse, key storage, clock, and memory integrity checks
- Memory and bus scrambling and protection
- Register — and memory-range address protection and locking
- TRNG entropy monitoring
- Boot-time and live-status checks
Physical security & countermeasures

Physical defenses
- Glitch
- Voltage
- Light
- Temperature

Alert responder
- Interrupt
- NMI
- Freeze
- Reset

Online checks
- Alert send
- Keymgr integrity
- TRNG integrity
- Clk integrity
- Bus parity

Physical security & countermeasures

Google Cloud
Open Titan
Moving from Titan to Open Titan

**Thesis**
The functional security mechanisms, provenance and digital implementation are commodities and thus good candidates for open sourcing

**Evidence**
Credible open ISAs, our RTL repositories, standard crypto primitives

**Outcome**
An open, transparent implementation of a secure cloud root of trust
What would **Open Titan** look like?

**Open Titan**

- Secure RISC - V 32b core
- PMU
- Testability / MFGability
  - jitter RC
  - timer RC
  - Low speed RC
- Debug ports
- Test ports
- Memory
  - ROM
  - SRAM
  - Flash
  - OTP (Fuse)
- Defenses
  - Shield
  - Temp sense
  - Volt sense
  - Device state
  - Alert resp
- Peripherals
  - DMA
  - EC/RSA crypto
  - AES/SHA/HMAC
  - Key manager
  - TRNG
  - timers
  - USB 1.1
  - SPI mstr/slvc
  - UART rx/tx
  - I2C mstr/slvc
  - GPIO
- Open source IP
- Proprietary foundry IP
- analog IP / digital wrap
- USB ports
- SPI ports
- Muxable data ports
- Muxable data ports
What would **Open Titan** look like?

**Open source digital IP**

**Analog wrappers**

- Secure RISC - V 32b core
- PMU
- Testability / MFGability
  - Jitter RC
  - Timer RC
  - Low speed RC
- ROM
- SRAM
- Flash
- OTP (Fuse)
- USB 1.1
- SPI mstr/slv
- UART rx/tx
- I2C mstr/slv
- GPIO
- DMA
- EC/RSA crypto
- AES/SHA/HMAC
- Key manager
- TRNG
- Timers
- Shield
- Temp sense
- Volt sense
- Device state
- Alert resp
- Test ports
- USB ports
- SPI ports
- Muxable data ports
- Muxable data ports

**Defenses**

- Open source IP
- Proprietary foundry IP
- Analog IP / digital wrap

**Peripherals**

**Memory**
What would **Open Titan** look like?

**Required vendor collateral:**
- STDCELL, memories, pads, etc.

### Secure RISC-V 32b core

- PMU
- Testability / MFGability
  - Jitter RC
  - Timer RC
  - Low speed RC

### Memory
- ROM
- SRAM
- Flash
- OTP (Fuse)

### Peripherals
- DMA
- EC/RSA crypto
- AES/SHA/HMAC
- Key manager
- TRNG
- Timers
- USB 1.1
- SPI mstr/slave
- UART rx/tx
- I2C mstr/slave
- GPIO

### Defenses
- Shield
- Temp sense
- Volt sense
- Device state
- Alert resp

### Open source IP
- Proprietary foundry IP
- Analog IP / digital wrap
Questions

For additional information
That’s a wrap