Titan silicon root of trust for Google Cloud
Perspective:
We need a silicon root of trust
## Chip Requirements

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusted Machine Identity</td>
<td>First Instruction Integrity</td>
<td>Tamper-evident logging</td>
<td>Trusted implementation</td>
</tr>
</tbody>
</table>

- On-chip verified boot
- Cryptographic identity & secure mfg
- Boot Firmware signature check + monitor
- Silicon physical security
- Transparent development, full-stack
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
<table>
<thead>
<tr>
<th>Why make our own?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Implementation transparency</strong></td>
</tr>
<tr>
<td>Complete ownership, auditability, build local expertise</td>
</tr>
<tr>
<td><strong>Agility &amp; velocity</strong></td>
</tr>
<tr>
<td>Technology changes, new risk vectors arrive</td>
</tr>
<tr>
<td><strong>No existing solutions</strong></td>
</tr>
<tr>
<td>Vendor-agnosticity, custom features</td>
</tr>
</tbody>
</table>
Titan specifications

- Embedded 32b processor
- 8kB ROM
- 64kB SRAM
- 512kB Flash
- 1kB OTP (Fuse)
- EC/RSA crypto
- AES/SHA/HMAC
- Key manager
- TRNG
- USB 1.1
- UART
- SPI mstr/slv
- I2C mstr/slv
- GPIO
- PMU
- Testability / MFGability
- Jitter RC
- Timer RC
- Low speed RC
- Debug ports
- Test ports
- Muxable data ports
- Shield
- Temp sense
- Volt sense
- Device state
- Alert resp
- Memory
- Defenses
- Memory
- Defenses
- Muxable data ports
- Muxable data ports
Interesting subunits

- Flash
  - 2 banks for code storage, in-field upgrades, partial secret material
- Fuse
  - Security settings, partial secret material, device state tracking, feature enablement
- Crypto units
  - AES, SHA/HMAC, big-int accelerator for EC, RSA (microcoded)
- Key manager
  - Custom control of key generation and storage
- TRNG
  - Custom analog design, low power, uses ring-oscillator instability
- Internal clocks
  - Spread-spectrum jittery clock for random behavior, fixed-frequency for communication
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

RAW  MFG Test  PROD  DEV  RMA  RIP

Burn fuse
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
Physical and tamper-resistant security
Physical security & countermeasures

Anti-glitch / anti-tamper mechanisms

- Attack detection (glitch, laser, thermal, voltage, probe)
- 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
- Only internal clocks, internal code
Physical security & countermeasures

- **Physical defenses**
  - Glitch
  - Voltage
  - Light
  - Temperature

- **Alert responder**
  - Alert send

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

- **Events**
  - Interrupt
  - NMI
  - Freeze
  - Reset
That’s a wrap