Root of Trust (RoT) technology is becoming a requirement for securing connected devices, their data, and, by extension, the entire infrastructure they communicate with.

But just like we learned in other areas of the IoT, hardware can be limiting. Hardware RoTs mean additional upfront expense, increased time to market, and reduced in-field flexibility. As a black box, they can block out threats as well as embedded and IoT software developers working with them. Regardless of the application or use case, engineers are often at the mercy of vendor functions programmed in at manufacture.

There’s also the option of building your own RoT into commodity hardware. Of course, this is risky. In fact, without a team of experienced security professionals, it may be riskier and more costly than utilizing whatever limited off-the-shelf protections are available in today’s most basic silicon.

There has never been much middle ground, until now.

Now, security IP providers are working to integrate state-of-the-art RoT capabilities available in technologies such as SRAM physically-unclonable functions (PUFs) in software development kits (SDKs) that bridge the accessibility gap.

The first of these is Intrinsic ID’s BK Software IP.
When you open your news application each morning, chances are high that one of the headlines is about the latest security breach. The increasing frequency of these headlines is just one more proof point that, while the IoT has brought new dimensions of value to both users and customers, it also opens them up to vulnerabilities and cyber-attacks.

Some of the most significant vulnerabilities in IoT devices today result from the lack of a robust Root-of-Trust (RoT).

An RoT forms a foundation from which system security operations such as secure storage, key management, device identity, secure boot, secure firmware upgrades, and other cryptographic services are derived. In order to act as a true trust anchor, an RoT should be immutable and hardware-based to increase its immunity to attack.

Several techniques can be used to create a hardware RoT, including injecting keys into secure memory regions, on-chip random number generation (RNG), off-the-shelf solutions such as secure elements and cryptographic co-processors, specialized custom hardware, and finally, SRAM physical unclonable functions (PUFs).

The Dark Side of Hardware Security

However, hardware RoT solutions do come with some limitations. For example, many silicon vendors produce secure elements or build hardware RoTs into their processors, which at first glance seems fantastic. However, a deeper dive reveals increased bill of material (BOM) costs, complex RoT transfer mechanisms, limited RoT features, and reduced accessibility to software developers.

These drawbacks may lead you to want to customize your vendor-supplied RoT, which is understandable. However, silicon RoT vendors are often reluctant to perform custom or specialized provisioning beyond their default offerings for anyone but key high-volume customers.

All of that's before considering that the complexity of hardware RoTs usually requires they be integrated at silicon fabrication centers early in the build process, which comes with cost and lead time constraints that disqualify most businesses in need of small- to medium-scale production runs. Given the supply chain issues of the past several years, the last thing you want to do is add the procurement risk of another discrete part like a secure element to your BOM anyway.

In all, it's not uncommon for hardware RoTs to leave users with a sub-optimal, inflexible solution at a higher cost. But for all their limitations, it's essential to note that hardware-based security is still the most robust option and should be considered a must-have for any IoT edge device.

Still, the demand remains for a hardware RoT that eliminates expensive discrete components, reduces reliance on silicon providers, and provides developers the ability to modify its internal functions with ease.
In other words, the IoT engineering community needs hardware RoTs that exhibit the characteristics of software. One technology that can deliver on this demand is the SRAM PUF.

**The randomness of standard SRAM**

SRAM PUFs are a cryptographic technology that’s been gaining momentum since the mid-2000s when papers emerged on their potential for creating “injection-less” RoT keys and managing key vaults.

SRAM PUFs are distinct from other hardware RoT key storage mechanisms in that they generate a device-unique root key from a device’s unique silicon properties. And unlike RNGs that generate new random numbers every time they are called, SRAM PUFs can be used to reconstruct the same device-unique root key repeatedly.

This is possible because SRAM PUF root keys are derived from sub-micron process variations created during the production of SRAM transistors. Applying power across SRAM transistors produces values of 0 or 1 for each SRAM memory cell, which offers a highly random yet repeatable pattern for each chip that can be leveraged in the creation of unique device root keys.

Figure 1 offers a high-level overview of how the technology works.

![Figure 1. SRAM PUFs leverage process variations unique to each piece of silicon that can be used to generate device-specific root keys.](image)

The unique silicon fingerprint of SRAM PUFs means there’s no need to customize hardware, inject keys into a device, manage private keys, or work with silicon vendors or manufacturing facilities. SRAM PUF root keys can encrypt all manner of other keys, data, and security services, meaning they can also function as the foundation of secure vaults on devices with no built-in hardware RoT such as pre-Armv8-M microcontrollers.

In fact, many semiconductor companies use SRAM PUF technology from IP vendors such as Intrinsic ID in the creation of secure semiconductor solutions today. However, to truly democratize silicon RoT technology this hardware IP must make the jump to software IP that can be accessed, understood, and implemented by IoT application developers at scale.
From SRAM to Software: creating a hardware rot with BK Software IP

Intrinsic ID’s BK Software IP is a secure root key generation and management solution for IoT security. As a software library, it streamlines IoT OEM, ODM, and SI security efforts by creating unique, internally generated device identities. But since these are derived from the inherent randomness of SRAM PUFs, it’s actually the only hardware entropy source currently available that doesn’t have to be loaded at silicon fabrication.

In practice, BK Software IP generates internal root keys on any device with uninitialized SRAM and stores them in a local RoT core that’s configured during device enrollment. Keys generated and managed by BK never leave this secure vault, are never exposed to the outside world, and are only ever accessible to BK’s own internal cryptographic functions.

The BK API lets IoT developers securely generate cryptographic keys and perform other symmetric key and elliptic curve cryptography functions, which simplifies the handling of keys and implementation of crypto operations. It can also be integrated as a trust anchor for OpenSSL, wolfSSL, mbed TLS, and other libraries that helps extend a chain of trust beyond just a single device.

Compatible with virtually any 5 nm to 350 nm device with access to uninitialized SRAM, BK Software IP eliminates the need for expensive, security-dedicated silicon in all sorts of use cases. And for the first time, it introduces the ability to bring hardware RoT-class security to legions of IoT devices already deployed in the field.

The following describes two ways BK Software IP can be used to bolster security in new and existing IoT systems today.

Using hidden Sram signatures for root Key Generation and Enrollment

The BK Software IP root key generation and enrollment process is only performed once per device. During this process, a unique power-up pattern is read from a pre-defined area of SRAM.

At startup, the selected SRAM region produces a unique yet slightly noisy bit pattern that's processed by BK's internal algorithms to extract a stable root key. Each root key is unique to each device and never has to be stored because it’s derived from the distinct SRAM startup signature. Furthermore, powering off the device removes the key from memory, protecting it from hardware-based attacks that might otherwise reveal the key!

As part of the enrollment process, helper data is generated that allows the root key to be reconstructed from SRAM on subsequent startups (Figure 2). This helper data is used by BK Software's internal error correction mechanisms to remove random noise in SRAM power-up patterns. This helper data does not need to be kept secret and can be stored in the clear.
Figure 2: The enrollment process generates the unique private key and helper data used to retrieve the key consistently over at least 25 years.

This method is stable over a wide range of voltages, temperatures, and other external conditions, and supported by a proprietary anti-aging technique that’s resilient against silicon degradation. It has been proven to work over at least a 25-year timescale.

A Secure Vault for Every Device

But BK Software IP can be used for much more than just generating a root key. It can also be leveraged in the creation of an RoT.

As mentioned, RoTs support a variety of security services including device provisioning, supply chain protections, hardware/software binding, anti-counterfeiting, authentication, and, of course, secure storage of device keys, communication keys, and sensitive data. Just like SRAM PUF hardware IP, BK Software IP can function as a secure vault in these use cases by encrypting keys and data with keys derived from the PUF root key.

One particularly intriguing application of secure vaults based on BK Software IP today is the secure storage of device or system configuration files, which allows IoT device manufacturers to support multiple product variants on the same physical hardware.

Of course, an obvious question is how keys and other data stored in a secure vault like this can be used to communicate with the outside world. It’s a valid question with a straightforward answer, at least to those familiar with public key infrastructure (PKI).

A so-called keypair consisting of a private key and public key can be derived from the root key created during the enrollment phase. That keypair’s public key can be included in a digital certificate that’s signed by a certificate authority and then used to authenticate the device with a cloud provider and conduct secure communications (Figure 3).
Figure 3: BK Software IP generates a private/public key pair that can be used to create a device certificate. That certificate can be used to authenticate a device with your cloud provider of choice and/or secure network communications.

In this example, there is no need to program a root key or even know what it is!

**Getting Started with a software-based hardware root of trust**

Especially if you’ve never heard of SRAM PUF technology before, you’re probably wondering how ready BK Software IP is for use in IoT devices. The short answer is, “it’s ready.” BK Software IP’s underlying technology has already been proven in more than 350 million devices that have been certified by EMVCo, Visa, CC EAL6+, PSA, ioXt, and governments across the globe.

Figure 4 summarizes the different BK Software IP configuration options. Each is available as a precompiled binary for the target architecture that comes with header files specifying library interfaces and full documentation. Code examples for integration with mbed TLS and WolfSSL are also provided.

<table>
<thead>
<tr>
<th>BK Configurations</th>
<th>Safe</th>
<th>Plus</th>
<th>Pro</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security strength (bits)</td>
<td>128/256</td>
<td>128/256</td>
<td>128/256</td>
</tr>
<tr>
<td>SRAM PUF (kB)</td>
<td>0.7/1</td>
<td>0.7/1</td>
<td>0.7/1</td>
</tr>
<tr>
<td>Code size (kB)</td>
<td>7</td>
<td>9</td>
<td>15-26</td>
</tr>
<tr>
<td>Generate device keys and random values</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Wrap and unwrap application keys</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Public key crypto functions (ECDSA and ECDH)*</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>PKI elements: cryptogram, certificate signing request (CSR), self-signed certificates (SSC)</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

* Includes ECDSA sign and verify, ECDH shared secret, elliptic-curve support set: P192, P224, P256, P384, P521

Figure 4: The various configuration options for BK Software IP available today.

**IoT security for all**

One of the most significant issues facing IoT devices today is the development of a robust Root of Trust. Unfortunately, traditional hardware-based RoTs have proven to be costly and inflexible to the needs of software developers and IoT organizations alike.
The need for RoT solutions that exhibit both the performance of hardware and flexibility of software couldn’t be more clear. Currently, the only technology capable of enabling both is the SRAM PUF and, more specifically, Intrinsic ID’s BK Software IP that can be deployed on devices on their way to manufacture or even retrofitted on those already deployed in the field today.

To explore SRAM PUFs in more detail, download BK-Demo Software IP, try out the example projects, or check out the resources below.