



Software Hardware Binding with Quiddikey



Mass Scale Solution
Against Software Piracy



Software Hardware Binding with Quiddikey

Software-Hardware Binding solutions are typically required for Flash-based systems in which a processor executes software, stored in external Flash memory, that is being transferred to RAM memory before being executed. In these systems, software can easily be spied upon and illegally copied to another equivalent processor if no appropriate protection is applied. This attack is particularly feared in the software licensing and gaming industry where software or game publishers need to protect distributed content from being illegally copied or transferred to illegitimate devices without the appropriate ownership rights.

One well-known strategy to protect against such attacks, also called secure node-locking, is to encrypt the software with a device specific key and to store the software in non-volatile memory in encrypted form. This allows to protect it against copying as well as to keep it confidential. In order to decrypt the software before executing it, the device needs to store the corresponding encryption key in a safe way. Storing secrets safely on chip either requires secure memory such as secure EEPROM or other costly non-volatile elements such as hidden OTP bits or hidden fuses.

The keys stored in this way are not device specific and as soon as one of them leaks due to an attack, the software can be decrypted, re-encrypted to any desired key and copied onto any other device holding such a key. In addition to this, this attack allows reverse-engineering of the software such that the intellectual property on which it is based becomes exposed. From an economic point of view, solutions based on embedded non-volatile memory put often a very heavy burden on the bill of material in high-volume applications.

Ideally software-hardware binding solutions avoid these cloning attacks and specifically lock the encrypted software to a unique device such that it cannot execute on any other, even equivalent, platform. This is precisely what is achieved by the use of Intrinsic-ID's Quiddikey™. Based on Quiddikey, **a software image is bound to a specific device** using its intrinsic security elements hidden in the hardware and a revolutionary approach based on Physically Unclonable Functions. This can be achieved *with device unique random keys as well as with a global protection key*. There is no need for on-chip secure non-volatile memory and the costs for implementation and integration are low since this solution only uses standard components and logic.



How It Works

Quiddikey™ is the sole and unique security solution in the market which allows to *dynamically* reconstruct the on-chip secret key used for secure file and software decryption *without ever storing that key*. This means that the key is not present on the device when it is powered off, but generated on-the-fly using device-dependent physically unclonable functions (PUFs) every time it is needed. It defeats the most advanced invasive hardware attacks on the key itself by simply not storing it.

Instead of storing the key in tamper-resistant non-volatile memory (typically secure EEPROM) or even hard-wiring it into the encryption core, Quiddikey™ allows for secure key extraction from unique physical properties of the underlying hardware. This new approach is called *Hardware Intrinsic Security* or *HIS*.

The principle can best be described as “biometrics for electronic devices” and uses device unique random start-up values of an uninitialized SRAM block. Once in the device’s lifetime, during *Enrollment*, these physically unclonable elements in the silicon are read, PUF data is extracted and turned into a so-called *activation code*. This activation code is **not sensitive**, unique for every device and is stored in the (possibly insecure) non-volatile memory of the chip or anywhere else in the system (where it is accessible to the chip). Every time the key is needed while the chip is in use, during *Reconstruction*, the device-unique PUF data is measured again and combined with the activation code to reproduce the *same* device-unique encryption key.

For applications in which the key is shared between a number of devices of the same family, Quiddikey also virtually allows to “program” a user selected key into the activation code, such that the reconstructed key results in a common key on multiple devices. This is particularly relevant when the software is encrypted ahead of time and sent to the device manufacturers in encrypted form. In both cases, the activation code remains **unique per device** and does not leak **any** information about the underlying key. Copying such an activation code from one device to another results in a **non-functional device** since the SRAM PUF data of another device does not match with this activation code and will result in the wrong key being reconstructed, thus protecting such systems against cloning and counterfeiting attacks simultaneously.

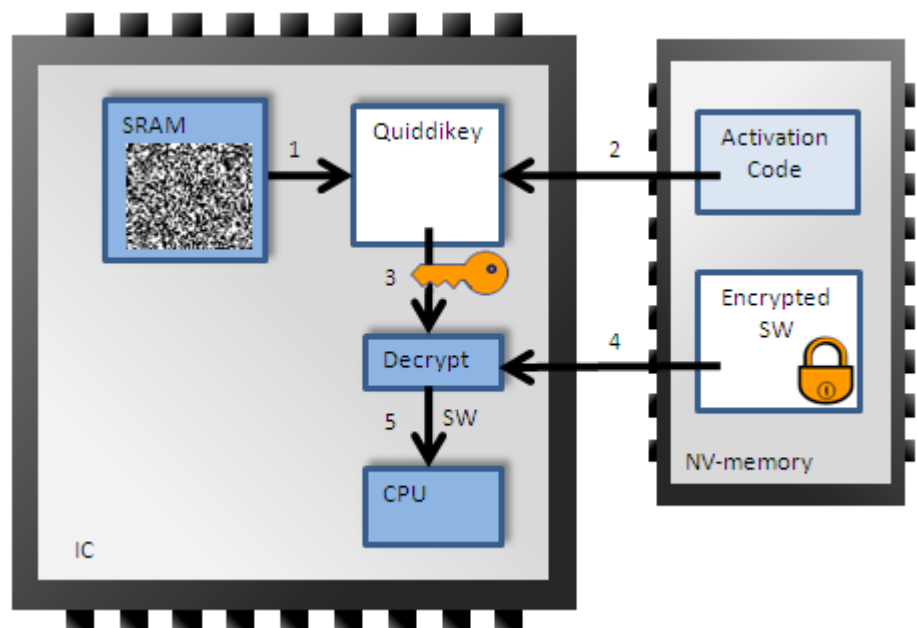
Quiddikey Unique Features

The figure below provides an overview of how key reconstruction works in the field, once a device has been enrolled with the right key.

Quiddikey™ unique features

- No key storage at power-off in the system
- Flexible key programming mechanism
- One die serves multiple customers
- Low implementation cost
- On-chip enrollment procedure
- Standard process steps and hardware components
- Standard manufacturing flow
- All technology nodes supported below 180nm

Quiddikey™ integrates seamlessly into existing customer platforms and devices.



This figure shows how the decryption key needed in step 3 is reconstructed from an SRAM reading in step 1 combined with an activation code reading in step 2 using the Quiddikey module. In step 3, the key is used by the AES decryption core to decrypt the encrypted software (step 4) stored in non-volatile memory (for example in Flash). In step 5, the decrypted software is executed by the CPU.

Components

- **Hardware Quiddikey IP**
 - **Functionality**
 - Generates an activation code during enrollment
 - Reconstructs user selected key(s) or device unique random key(s)
 - Different key lengths possible
 - **Deliverables**
 - Synthesized RTL (VHDL or Verilog), synchronous logic design
 - Hardware interface specifications
 - Test benches
 - Integration guidelines
 - Integration support
- **Standalone or integrated AES encryption/decryption core**
 - **Functionality**
 - Receives a decryption key from Quiddikey HW module
 - Decrypts encrypted software on-chip
 - 128-bit or 256-bit key, ECB and CBC modes
 - **Deliverables**
 - Synthesized RTL (VHDL or Verilog), synchronous logic design
 - Hardware interface specifications
 - Test benches
 - Integration guidelines
 - Integration support
- **Customer Software Encryption tool**
 - **Functionality**
 - Receives user selected key or device unique random key
 - PC tool generates the encrypted version of customer software
 - **Deliverables**
 - Software specification
 - Installation and user manuals
 - Interface specification

System Requirements

Quiddikey Hardware IP for a 128-bit Security Level

Reserved SRAM	Gates	Area (90nm TSMC)	Reconstruction Performance	Enrollment Performance
1 Kbyte	20k Gates	~ 0.08 mm ²	10k clock cycles	10k clock cycles

AES Encryption and Decryption Hardware IP for a 128-bit Security Level

Fully Integrated AES core**	Area (90nm TSMC)	Standalone AES core*	Area (90nm TSMC)	Decryption Performance	Key setup (Decryption)
~12k Gates	~0.03 mm ²	~6k Gates	~ 0.015 mm ²	56 cycles per 128-bit block	64 cycles

Notes:

* other trade-offs between gatecount and performance available on request

** includes data flow controller, BIST, APB state-machine and interface registers

For more information please contact: sales@intrinsic-id.com

About Intrinsic-ID: Intrinsic-ID is a security company that spun out from Philips in 2008, providing system level security solutions for content and data protection, unique device identification and cost effective key storage. Its solutions are based on patented Hardware Intrinsic Security Technology, hence cost effective and suitable for very high security applications. At the same time they offer the advantage of using only standard CMOS components as well as a standard design and manufacturing flow when deployed in hardware.

For more information
please contact:

sales@intrinsic-id.com

High Tech Campus 9
5656 AE Eindhoven
The Netherlands
Tel: +31 40 851 90 20

2033 Gateway Place
Suite 500
San Jose, CA 95110
USA
Tel: +1 408 573 6186



www.intrinsic-id.com