

---

**Design Security in Stratix III Devices****Introduction**

As FPGAs are increasingly used for critical system functions, protecting designs and intellectual property (IP) implemented inside FPGAs is becoming more important. Altera® Stratix® III devices are the first high-density and high-performance FPGAs to use the advanced encryption standard (AES) with both non-volatile and volatile key programming to protect designs against copying, reverse engineering, and tampering. To make the Stratix III design security solution more secure and to protect the AES key, many security features have been implemented. The solution has been reviewed by external security consultants during the design phase and improvements have been made based on their feedback. This white paper details the security protection provided by the Stratix III design security solution.

**SRAM-Based FPGA Design Security**

SRAM-based FPGAs are volatile and require external memory to store their configuration files, which results in three security risks: copying, reverse engineering, and tampering.

*Copying*

Copying an FPGA involves making identical copies of the design without understanding how it works. A device can be copied by either reading the design out of the memory device or capturing the configuration file when it is sent from the memory device to the FPGA at power-up. The stolen design can then be used to configure other FPGAs. This approach constitutes a primary form of IP theft and can cause significant revenue loss to the designer.

*Reverse Engineering*

Reverse engineering involves analyzing the configuration file to recreate the original design at the register transfer level (RTL) or in schematic form. The recreated design can then be modified to gain a competitive edge. This form of IP theft is more complex than copying and usually requires significant technical expertise. It is also time- and resource-intensive, and sometimes involves more work than creating a design from scratch.

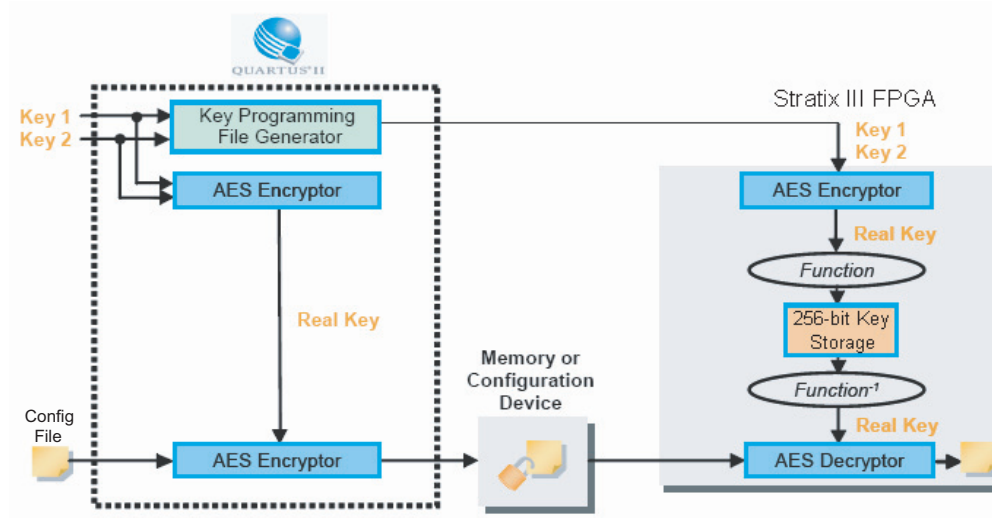
*Tampering*

Modifying the design stored in the device or replacing it with a different design is considered tampering. The tampered device may contain harmful design code capable of causing a system to malfunction or steal sensitive data. This type of design security breach is a particular concern in military, financial, and gaming applications. Today, tampering is also becoming a concern in the consumer market where a design can be modified to access unauthorized or premium services.

**Stratix III Design Security Solution**

Stratix III devices are SRAM-based FPGAs. To provide design security, Stratix III FPGAs use a 256-bit security key for configuration bitstream encryption. The secure configuration flow, shown in [Figure 1](#), can occur after synthesis, fitting, and timing analysis in the Quartus II software. Thus, hardware designers can apply the secure configuration features after optimizing their designs in the Mentor Graphics Precision Synthesis software and verifying their designs in the Mentor Graphics ModelSim® simulation software can apply the security key.

Figure 1. Secure Configuration Flow




Secure configuration can be carried out in the following three steps:

1. *Program the security key into the Stratix III FPGA:* The Quartus® II software requires 256-bit user-defined keys (Key 1 and Key 2) to generate a key programming file. The key programming file containing Key 1 and Key 2 information is then loaded into the Stratix III FPGA through the JTAG interface. The Stratix III built-in AES encryption engine generates the real key (used to decrypt configuration data in Step 3) by encrypting Key 2 with Key 1. The real key is then processed by a proprietary function prior to being stored in the 256-bit key storage, which can either be volatile (SRAM-based) or non-volatile (poly fuse-based).
2. *Encrypt the configuration file and store it in the external memory:* The Quartus II software requires the same 256-bit user-defined keys (Key 1 and Key 2) used in step 1 to encrypt the configuration file. The Quartus II AES encryption engine generates the real key by encrypting Key 2 with Key 1. The real key is used to encrypt the configuration file. The encrypted configuration file is then loaded into the external memory, such as a configuration or flash device.
3. *Configure Stratix III FPGA:* At system power-up, the external memory device sends the encrypted configuration file to the Stratix III FPGA. The 256-bit key within the Stratix III FPGA is processed by the inverse of the proprietary function to generate the real key. The Stratix III built-in AES decryption engine then uses the real key to decrypt the configuration file and configure itself.

## Stratix III Key Programming Solutions

Altera provides different types of solutions for design security key programming via the JTAG interface, supporting on-board and off-board key programming.

 The steps for programming the volatile and non-volatile key are included in the Stratix III Design Security application note. Please contact your local Altera sales representative for more information.

## AES Encryption Algorithm

AES is a Federal Information Processing Standard (FIPS-197) and has been approved to be used by U.S. government organizations to protect sensitive, classified information. It is also expected to be widely adopted both commercially and globally.

AES is a symmetric block cipher that encrypts and decrypts data in blocks of 128 bits. The encrypted data is subject to a series of transformations including byte substitutions, data mixing, data shifting, and key additions. AES comes

in three different key sizes: 128 bits, 192 bits, and 256 bits. The 256-bit AES key size is used in Stratix III FPGAs for both security and efficiency. According to the National Institute of Standards and Technology (NIST), studies have shown that if one could build a machine that could discover a data encryption standard (DES) key in seconds, then it would take that same machine more than 149 trillion years to discover a 256-bit AES key. The Stratix III AES implementation has been validated as conforming to the FIPS-197 standard.

### *AES Decryption Block*

The main functions of the decryption block are:

- Determine whether the configuration data needs to be decrypted.
- Determine the security mode.
- Decrypt the data stream and decompress the data, if needed; otherwise, configure the device.

Prior to receiving encrypted data, the 256-bit security key must be entered and stored in the device. You can choose between a non-volatile security key and a volatile security key with battery backup. The non-volatile key and the poly fuse key verify bit (which indicates a poly fuse key is present) are stored in one-time programmable poly fuses, whereas the 256-bit volatile key and the volatile key verify bit (which indicates a volatile key is present) are stored in volatile key registers that are backed up with external battery power.

### *Key Storage*

The security key is stored in poly fuses and volatile key registers inside the Stratix III FPGA. Poly fuses are non-volatile and one-time programmable. Volatile key storage requires an external backup battery that allows the key to be stored in the event the device is powered down. The security key can be programmed into the Stratix III FPGA during regular manufacturing flow, with the FPGA either on-board (for both volatile and non-volatile keys) or off-board (for non-volatile key only).

## **Security Protection Provided by Stratix III FPGAs**

With configuration bitstream encryption, Stratix III FPGA designs are protected from copying, reverse engineering, and tampering.

### *Security Against Copying*

The security key in the Stratix III FPGAs cannot be read out through any interfaces. The poly fuses and volatile key registers storing the security key are hidden under layers of metals among hundreds of other poly fuses. It is very difficult to determine the functionality of a particular fuse or registers by simple visual inspection. The key bits are scrambled and distributed among other logic in the FPGA. In addition, the programming status of the key storage used for other functions can be different from device to device. This randomness makes it more difficult to identify which fuses or registers store the security key. Also, even if the key storage is identified, the real key used for decryption is not revealed because the real key is processed by the proprietary function prior to storage. Without knowing the real key, the design cannot be decrypted.

Stratix III FPGAs are secure against readback attacks since they do not support configuration file readback. This prevents attempts to read back the configuration file after it is decrypted within the FPGA.

Stratix III designs cannot be copied by programming the security key into another FPGA and configuring it with an encrypted configuration file. Two 256-bit keys are required to program the security key into the Stratix III FPGA. Since AES is used to generate the real key, it is virtually impossible to generate Key 1 and Key 2 from the security key. There is no known AES weak key.

### *Security Against Reverse Engineering*

Reverse-engineering any Stratix III design from a configuration file is very difficult and time-consuming, even without encryption. The Stratix III configuration file contains millions of bits and the configuration file formats are

proprietary and confidential. To reverse-engineer a Stratix III design requires reverse-engineering of the FPGA or the Quartus II design software to reveal the mapping from the configuration file to the device resources.

Reverse-engineering the Stratix III FPGA itself is also very challenging. Stratix III FPGAs are manufactured at TSMC on the most advanced 65-nm process technology. Unlike ASICs, standard tools and knowledge are not readily available to reverse-engineer these cutting-edge FPGAs. It can take a significant amount of time and resources to reverse-engineer just one logic block of the Stratix III FPGA and to locate the keys location within the FPGA.

Configuration bitstream encryption makes reverse-engineering even more difficult. Finding the security key to decrypt the configuration file is as difficult as copying it. It may be easier and quicker to build a competitive design from scratch than to reverse-engineer a secured Stratix III design.

### Security Against Tampering

The non-volatile keys are one-time programmable. Once the *tamper protection* bit is set, the FPGA can only accept configuration files encrypted with the same key. In addition, it does not allow further key programming. Attempts to configure the Stratix III FPGA with an unencrypted configuration file or a configuration file encrypted with the wrong key result in configuration failure. A configuration failure signals possible tampering of the design, whether in the external memory, during transmission between the external memory and the FPGA, or during remotely communicated system upgrades.

### Supported Configuration Schemes

The design security feature is available when configuring Stratix III FPGAs using the fast passive parallel (FPP) configuration mode with an external host (such as a MAX<sup>®</sup> II device or microprocessor), or when using active serial (AS) or passive serial (PS) configuration schemes. The design security feature is not available if you configure your Stratix III FPGA using FPP with an enhanced configuration device or JTAG-based configuration. However, depending on the security mode selected when the Stratix III device is encrypted, the device would support only selected configuration schemes.



For more information on the security modes supported in Stratix III devices and the corresponding configuration schemes for each security mode, refer to the *Design Security in Stratix III Devices* chapter of the *Stratix III Devices Handbook*: [www.altera.com/literature/hb/stx3/stx3\\_siii51014.pdf](http://www.altera.com/literature/hb/stx3/stx3_siii51014.pdf).

### Conclusion

Stratix III FPGAs offer a solid design security solution with many security features to protect designers' intellectual property and prevent tampering, making these devices well-suited for secure commercial and general military applications. These security features allow designers to deliver products designed using the Mentor Graphics and Altera design and verification tools to customers safely and quickly, providing a risk-free path for testing innovations with customers and differentiating their products from the competition.

### Resources

- Federal Information Processing Standards (FIPS-197):  
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
- Advanced Encryption Standard Algorithm Validation List:  
<http://csrc.nist.gov/cryptval/aes/aesval.html>



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
<http://www.altera.com>

Copyright © 2006 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.