ProASIC3/E Security

Introduction to Security in ProASIC3/E

The need for security on FPGA programmable logic devices has never been greater than today. If the contents of the FPGA can be read by an external source, the intellectual property of the system is vulnerable to unauthorized copying. Actel ProASIC3/E devices contain state-of-the-art circuitry to make the Flash-based devices secure during and after programming. The new ProASIC3/E Flash FPGAs consist of an FPGA array core and FlashROM (FROM). The FROM and the FPGA core fabric can be securely programmed independently of each other, allowing the FROM to be updated without changing the FPGA core fabric.

Actel has incorporated the Advanced Encryption Standard (AES) decryption core into the new ProASIC3/E devices, and has also included the Actel Flash-based lock technology, FlashLock™. Together they provide leading-edge security in a programmable logic device. Configuration data loaded in ProASIC3/E can be decrypted prior to being written to the FPGA core using the AES 128-bit block cipher standard. The AES encryption key is stored in on-chip, nonvolatile Flash memory.

ProASIC3/E devices have been designed with the most comprehensive programming logic design security in the industry. In the architecture of ProASIC3/E devices, security has been designed into the fabric of ProASIC3/E devices. The Flash cells are located beneath seven metal layers and the use of many device design and layout techniques makes invasive attacks difficult. Since device layers cannot be removed without disturbing the charge on the programmed (or erased) Flash gates, devices cannot be easily deconstructed to decode the design. ProASIC3/E is unique in being reprogrammable and having inherent resistance to both invasive and noninvasive attacks on valuable intellectual property (IP). Secure remote in-system programming (ISP) is now possible with AES encryption capability for the programming file during electronic transfer. Figure 1 shows a view of the AES decryption core inside ProASIC3/E devices. The AES decryption core is used to decrypt the encrypted programming file when programming.

This application note outlines the security features offered in ProASIC3/E devices, some applications and uses, as well as the different software settings for each application.

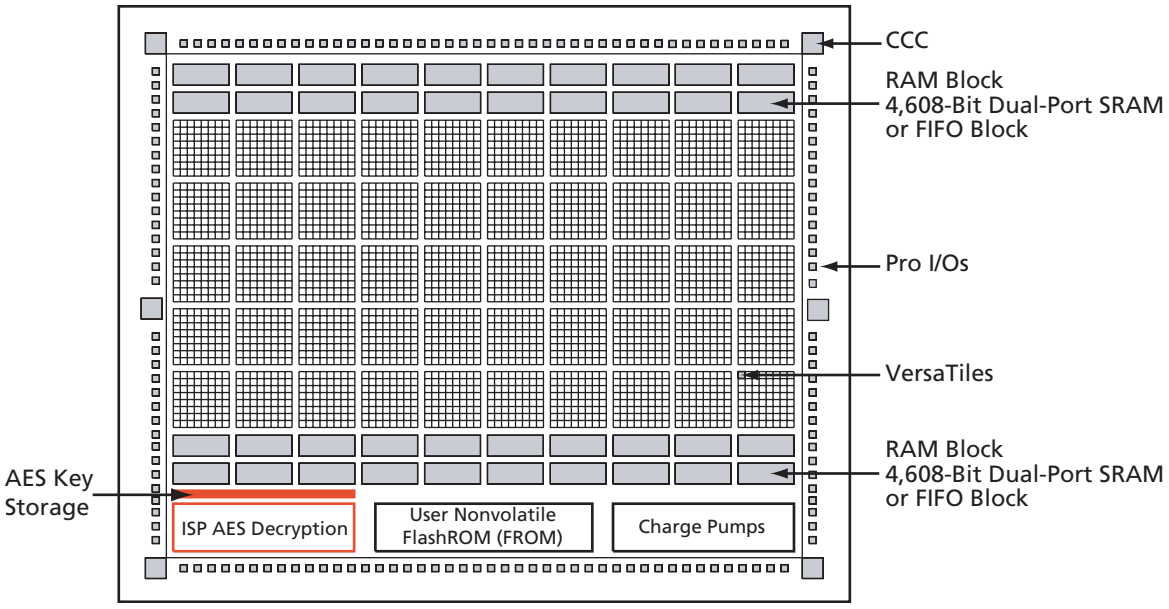


Figure 1 • **AES Decryption Core inside the ProASIC3/E FPGA (graphical representation only)**

ProASIC3/E Security Features

Consider ProASIC3/E devices as having two entities inside: FlashROM (FROM) memory and the FPGA core fabric. The two parts can be programmed or updated independently with a STAPL programming file. The programming files can be AES-encrypted or plain text. This allows maximum flexibility in providing security to the entire device. Refer to the *ProASIC3/E FlashROM (FROM)* application note for the FROM structure.

Unlike SRAM-based FPGA devices, which require a separate boot PROM to store programming data, ProASIC3/E devices are nonvolatile, and the configuration data is stored in on-chip Flash cells that are part of the FPGA fabric. Once programmed, this data is an inherent part of the FPGA array and does not need to be loaded at system power-up. SRAM-based FPGAs load the configuration bitstream upon power-up; therefore, the configuration is exposed and can be read easily.

The built-in FPGA core and FROM support programming files encrypted with the 128-bit AES (FIPS-192) block ciphers. The AES key is stored in dedicated, on-chip, Flash memory and can be programmed before the device is shipped to other parties (allowing secure remote field updates).

AES Encryption of Programming Files

ProASIC3/E devices employ the Advanced Encryption Standard (AES) as part of the security mechanism that prevents invasive and noninvasive attacks. The mechanism works by encrypting the programming file with AES encryption, then passing the programming file through the AES decryption core, which is embedded in the device. The file is decrypted there and the device is successfully programmed.

This AES key is protected by a FlashLock security Pass Key that is also implemented in ProASIC3/E devices. This FlashLock Pass Key technology is exclusive to the Actel Flash-based device families offered by Actel. FlashLock Pass Key technology can also be implemented without the AES encryption option, providing different security levels choices.

In essence, security features can be categorized into the following three options:

- AES encryption with FlashLock Pass Key protection
- FlashLock protection only (no AES encryption)
- No protection

Each of the above options will be explained in more detail in the following sections with application examples and software implementation options.

Advanced Encryption Standard (AES)

AES is a cryptographic algorithm that complies with Federal Information Processing Standard (FIPS) Publication 192, used by U.S. Government organizations to protect sensitive, unclassified information.

Actel has implemented the 128-bit AES (Rijndael) algorithm in ProASIC3/E devices. With this key size, there are approximately 3.4×10^{38} possible 128-bit keys. The DES standard has a 56-bit key size, which provides approximately 7.2×10^{16} possible keys. In their AES fact sheet, the National Institute of Standards and Technology (NIST) uses the following hypothetical example to illustrate the theoretical security provided by AES. If one were to assume that a computing system existed that could recover a DES key in a second, it would take that same machine approximately 149 trillion years to crack a 128-bit AES key. NIST continues to make their point by stating the universe is believed to be less than 20 billion years old.[1]

The AES key is securely stored on-chip in dedicated ProASIC3/E Flash memory and cannot be read out. In the first step, the AES key is generated and programmed into the device (for example, in a secure or trusted programming site). The Actel Designer software tool provides AES key generation capability. After the key has been programmed into the device, the device will only correctly decrypt the programming file(s) that has been encrypted with the same key. If the individual programming file content is incorrect, a Message Authentication Control (MAC) mechanism inside the device will fail in authenticating the programming file. In other words, when an encrypted programming file is being loaded to a device that

1. *National Institute of Standards and Technology, "ADVANCED ENCRYPTION STANDARD (AES) Questions and Answers," 28 January 2002, <http://csrc.nist.gov/CryptoToolkit/aes/aesfact.html> (10 January 2005).*

has a different programmed AES key, the MAC will prevent this incorrect data from being loaded into the device, preventing possible device damage. See Figure 2 for a graphical representation of this process.

It is important to note that the user decides what level of protection will be implemented for the device. When AES protection is desired, then the FlashLock Pass Key is also required to be set. The AES key is a content protection mechanism, while the FlashLock Pass Key is a device protection mechanism. When the AES key is programmed into the device, the device still needs the Pass Key to protect the FPGA and FROM content and the security settings, including the AES key. Using the FlashLock Pass Key prevents modification of the design contents by means of simply programming the device with a different AES key.

AES Decryption in ProASIC3/E Devices

ProASIC3/E has a built-in 128-bit AES decryption core. The AES core in ProASIC3/E decrypts the encrypted programming file and performs a MAC check that authenticates the programming file prior to programming. This will ensure the following scenarios:

- Correct decryption of the encrypted programming file
- Prevention of erroneous or corrupted data being programmed during the programming file transfer
- Correct bitstream passed to the device for decryption

Figure 2 shows the use of AES in the ProASIC3/E devices.

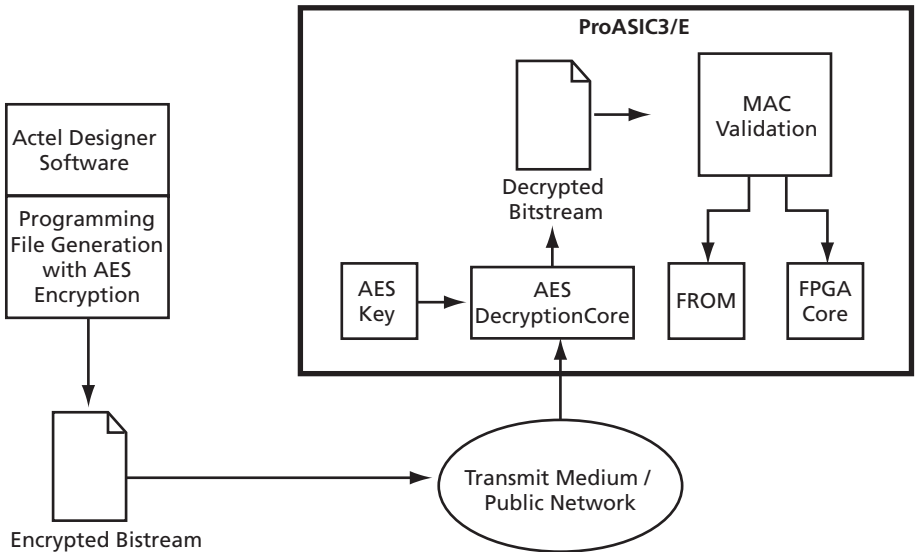


Figure 2 • **Example Application Scenario Using AES in ProASIC3/E Devices**

FlashLock

The 128-bit Flash-based FlashLock feature in ProASIC3/E works via a key mechanism. The user locks or unlocks the device with a user-defined FlashLock Pass Key. When the device is locked, there is no access to the FPGA without the correct FlashLock Pass Key. By default, functions such as device write, verify, and erase are disabled. The user also has the option of prohibiting Write to the FPGA array, but allowing Verify operation of the FPGA array and/or the Read operation of the FROM without the use of the FlashLock Pass Key. This option provides the user the freedom of verifying the FPGA array and/or reading the FROM contents after the device is programmed without having to provide the FlashLock Pass Key. The user can incorporate AES Encryption on the programming files to better enhance the level of security used. Figure 3 on page 4 shows the application scenarios and the corresponding security settings selected during the programming file generation step.

Permanent Security Setting Options

In applications where a permanent lock is not desired, yet the security settings should not be modifiable, ProASIC3/E devices can accommodate this requirement.

This application is particularly useful in cases where a device is located at a remote location, and the device must be reprogrammed with a design or data update. Refer to the "Application 3: Nontrusted Environment – Field Updates/Upgrades" section on page 6 for further discussion and examples of how this can be achieved.

The user must be careful when considering the Permanent FlashLock or Permanent Security Settings option. Once the design is programmed with the permanent settings, it is not possible to reconfigure the security settings already employed to the device. Therefore, exercise careful consideration before programming permanent settings.

Permanent FlashLock

The purpose of the permanent lock feature is to provide the benefits of the highest level of security to ProASIC3/E devices. If selected, the permanent FlashLock feature will create a permanent barrier, preventing any access to the contents of the device. This is achieved by permanently disabling write and verify access to the array, and write and read access of the FROM. After permanently locking the device, the device has been effectively rendered as one-time programmable. This feature is useful if the intended applications do not require design or system updates to the device. Refer to the "Implementing Permanent Security Settings in the Device" section on page 20 for the Permanent FlashLock option setting during programming file generation.

Security in Action

This section illustrates some applications of the security advantages of ProASIC3/E devices (Figure 3).

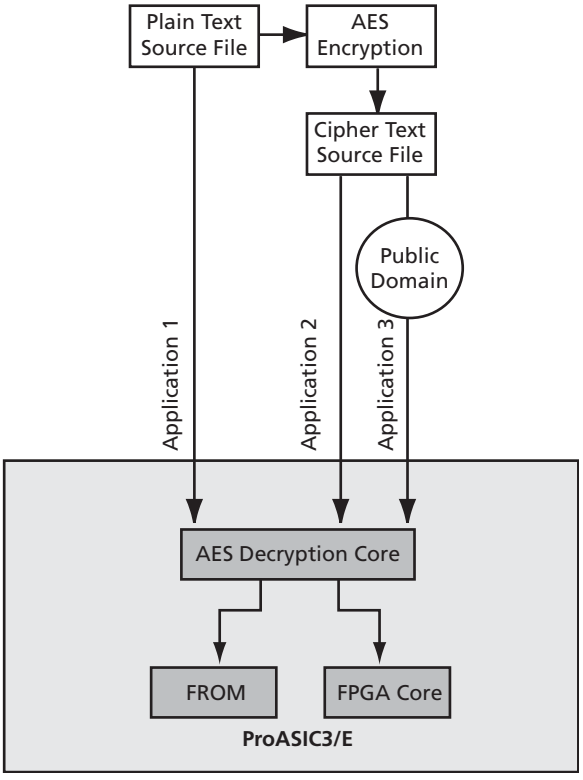


Figure 3 • **Security Options in ProASIC3/E**

Application 1: Trusted Environment

As illustrated in Figure 3 on page 4, this application allows the programming of devices at design locations where research and development take place. Therefore, encryption is not necessary and is optional to the user. This is often a secure way to protect the design since the design program files are not sent elsewhere. In situations where production programming is not available at the design location, programming centers (such as Actel in-house programming) provide a way of programming designs at an alternative, secure, and trusted location. In this scenario, the user will generate a STAPL programming file from the Designer software in plain text format containing information on the entire design, or the portion of the design to be programmed. The user can choose to employ the FlashLock Pass Key feature with the design. Once the design is programmed to unprogrammed devices, the design will be protected by this FlashLock Pass Key. If no future programming is needed, the user can consider permanently securing the ProASIC3/E device, as discussed in the "Permanent FlashLock" section on page 4.

Application 2: Nontrusted Environment – Unsecured Location

Often programming of the devices is not performed in the same location as actual design implementation, in order to reduce manufacturing cost. Overseas programming centers and contract manufacturers are examples of this scenario.

To achieve security in this case with ProASIC3/E, the AES key and the FlashLock Pass Key can be initially programmed in-house (trusted environment). This is done by generating a programming file with only the security settings and no design contents. The design FPGA core and/or FROM contents will be generated in a separate programming file. This programming file will have to be set with the same AES key that was used to program to the device previously, in order for the device to correctly decrypt this encrypted programming file. As a result, this encrypted design content programming file can be safely sent off-site to nontrusted programming locations for design programming. Figure 4 below shows a more detailed flow for this application.

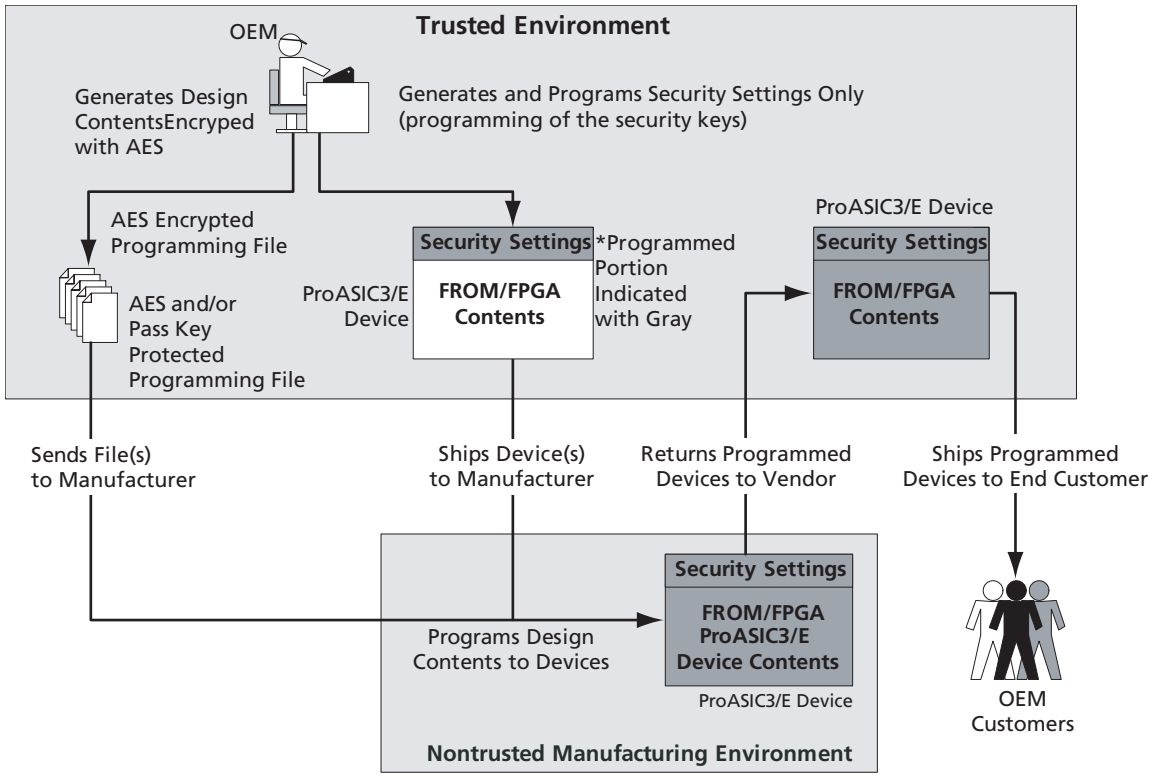


Figure 4 • **Implement Device Programming at Nontrusted Environment**

Application 3: Nontrusted Environment – Field Updates/Upgrades

In addition, programming or reprogramming of devices may occur at remote locations. Reconfiguration of devices in consumer products/equipment through public networks is one example. The remote system typically is already programmed with particular design contents. When design update (FPGA array contents update) and/or data upgrade (FROM contents upgrade) is necessary, an updated programming file with AES encryption can be generated, sent across public networks, and transmitted to the remote system. Reprogramming can then be done using this AES-encrypted programming file, providing easy and secure field upgrades. ProASIC3/E devices support this secure ISP using AES. The detailed flow for this application is indicated in Figure 5 on page 7. Refer to the application note, *Programming ProASIC3/E Using a Microprocessor*, for more information.

To prepare devices for this scenario, the user can initially generate a programming file with the available security setting options. This programming file is programmed to the devices before shipment. During the programming file generation step, the user has the option of making the security settings permanent or not. In situations where no changes to the security settings are necessary, the user can select this feature in the software to generate the programming file with permanent security settings. Actel recommends that the programming file will use encryption with the AES key, especially when ISP is done via public domain.

For example, if the designer wants to use an AES key for the FPGA Array and the FROM, **Permanent** needs to be chosen for this setting. At first, the user would do this by choosing the options to use an AES key for the FPGA Array and the FROM, and then choosing **Permanently lock the security settings**. A unique AES key would be chosen. Once this programming file is generated and programmed to the devices, the AES key is permanently stored in the on-chip memory, where it is secured safely. The devices would be sent to distant locations for the intended application. When an update is needed, a new programming file must be generated. The programming file must use the same AES key for encryption, otherwise the authentication will fail and the file will get programmed in the device. For more information on the different settings allowed by the Designer software, refer to the "Permanent Security Setting Options" section on page 4.

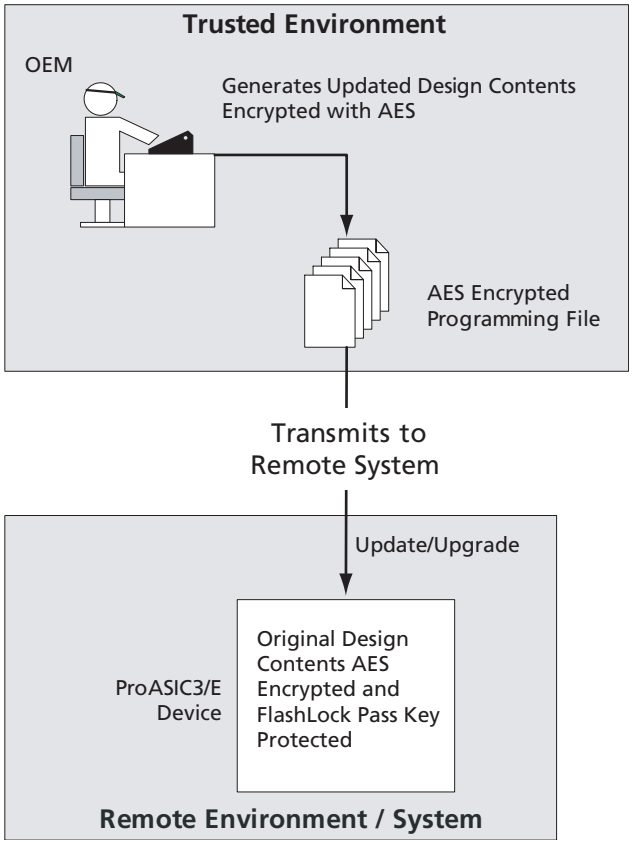


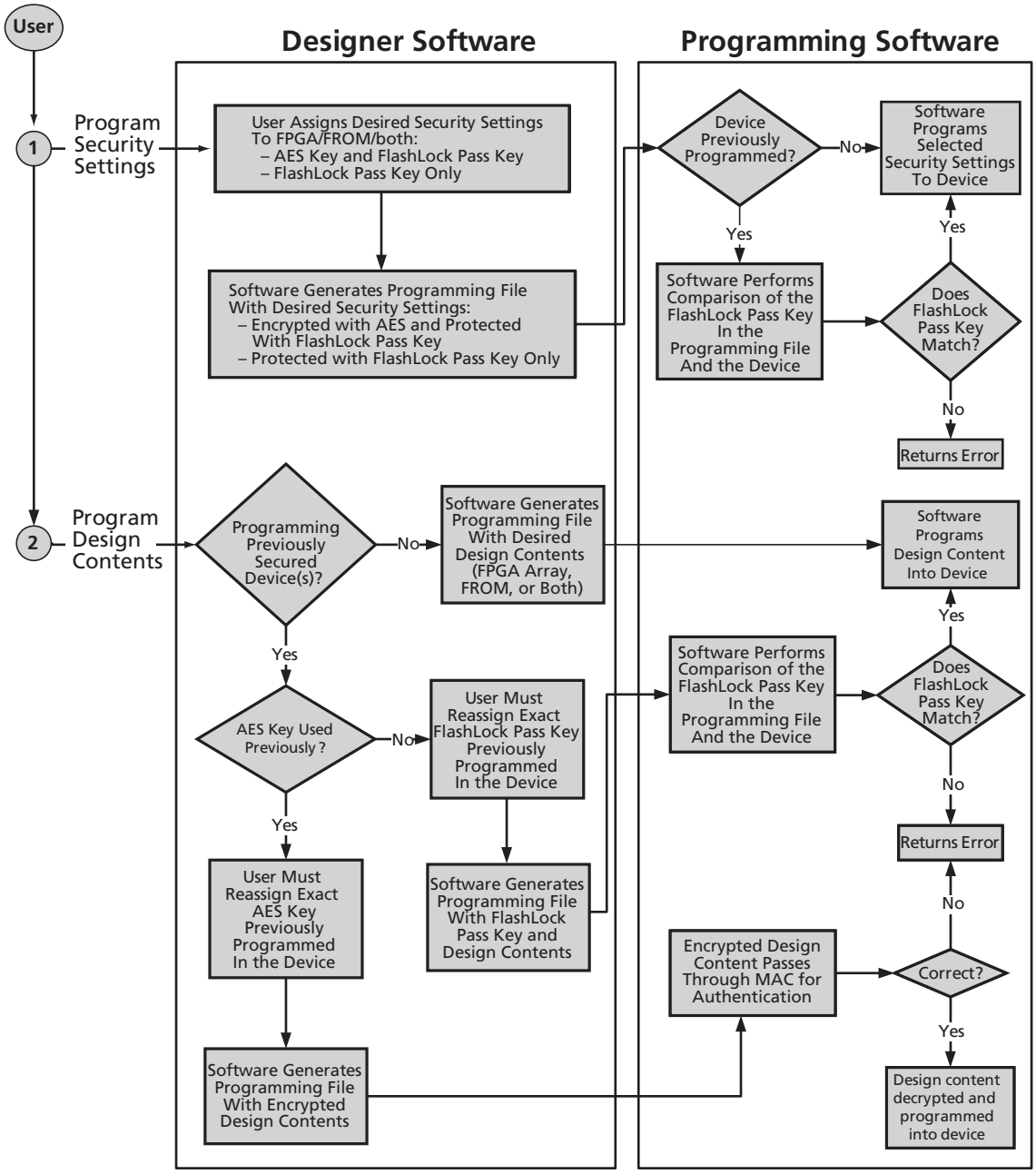
Figure 5 • **Remote Update in Non-Trusted Environment – Field Updates/Upgrades**

ProASIC3/E FROM Security Use Models

Each of the subsequent sections will describe in detail the available selections in Actel Designer software as an aid to understanding security applications and generating appropriate programming files for those applications. Before proceeding, it is helpful to review Figure 6 on page 8, which describes a general overview of the programming file generation flow within the Designer software, and what occurs during the device programming stage. Specific settings are discussed in the following sections.

In Table 6 on page 8, the flow consists of two sub-flows. Sub-flow 1 describes programming security settings to the device only, while sub-flow 2 describes programming the design contents only. In Application 1, described on page 5, the user does not need to generate separate files, but can generate one programming file containing both security settings and design contents. Then, programming of the security settings and design contents is done in one step. Both sub-flow 1 and sub-flow 2 are used. In Application 2, described on page 5, the trusted site should follow sub-flow 1 and 2 separately to generate two separate programming files. The programming file from sub-flow 1 will be used in the trusted site to program the device(s) first. The programming file from sub-flow 2 will be sent off-site for production programming. In Application 3, described on page 6, typically only sub-flow 2 will be used due to the fact that only updates to the design content portion are needed, and no security settings need to be changed.

In the event that update of the security settings is necessary, see the "Reprogramming Devices" section on page 14 for details. For more information on programming ProASIC3/E devices, refer to the *In-System Programming (ISP) in ProASIC3/E Using FlashPro3* application note.



Note: *If programming the Security Header only, just perform Step 1.*
If programming design content only, just perform Step 2.

Figure 6 • **ProASIC3/E Security Programming Flows**

Generation of the Programming File in a Trusted Environment – Application 1

As discussed in the "Application 1: Trusted Environment" section on page 5, in a trusted environment the user can choose to program the device with plain text bitstream content. It is possible to use plain text for programming even when the FlashLock Pass Key option has been selected. In this application, it is not necessary to employ AES encryption protection. For AES encryption settings, refer to the next sections.

The generated programming file will include the security setting (if selected), as well as the plain text programming file content for either the FPGA Array, FROM, or both. These options are indicated in Table 1.

Table 1 • **Plain Text Security Options**

Security Protection	FROM Core Only	FPGA Core Only	Both FROM and FPGA
No AES / No FlashLock	✓	✓	✓
FlashLock only	✓	✓	✓
AES and FlashLock	–	–	–

Note: *✓ = options that may be used*

The programming file generation for this scenario can be achieved as follows:

1. Check the Silicon features to be programmed (**Security Settings**, **FPGA Array**, **FlashROM**), as shown in Figure 7. Click **Next**.
   If **Security Settings** is selected (i.e., the FlashLock security Pass Key feature), the following step will be displayed to prompt the you to select the level of security level setting. If no security setting is selected, you will be directed to go to Step 3.

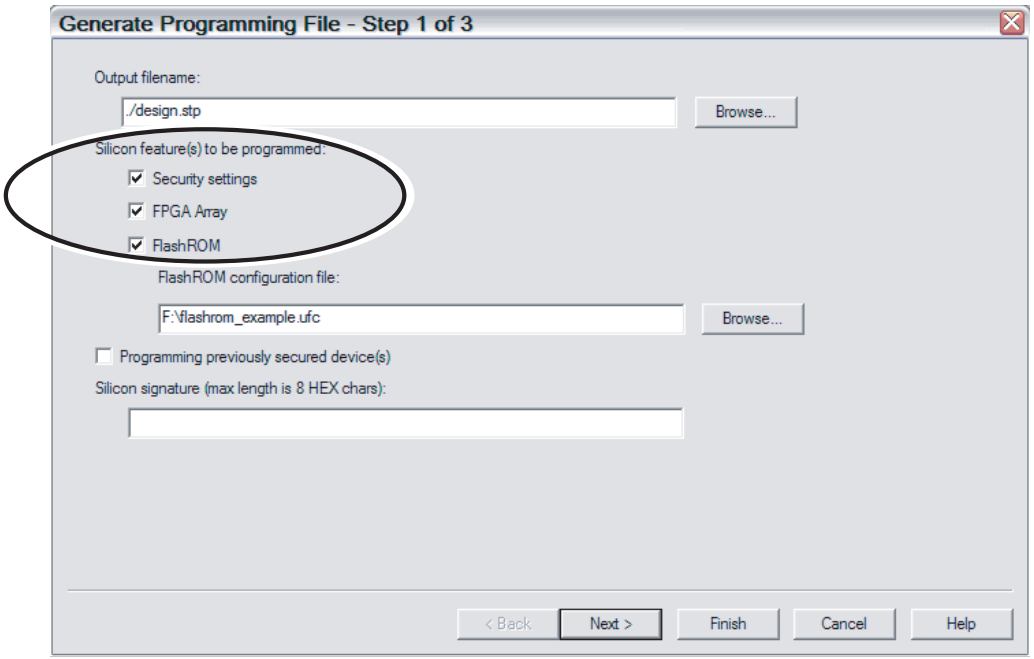


Figure 7 • **All Silicon Features Checked**

2. Choose the appropriate security level setting and enter a FlashLock Pass Key. The default is the **Medium** security level (Figure 8). Click **Next**.
   If you want to select different options for the FPGA and/or FlashROM, this can be set by clicking **Custom Level**. Refer to the subsequent section on Advanced Options for different custom security level options and descriptions of each.

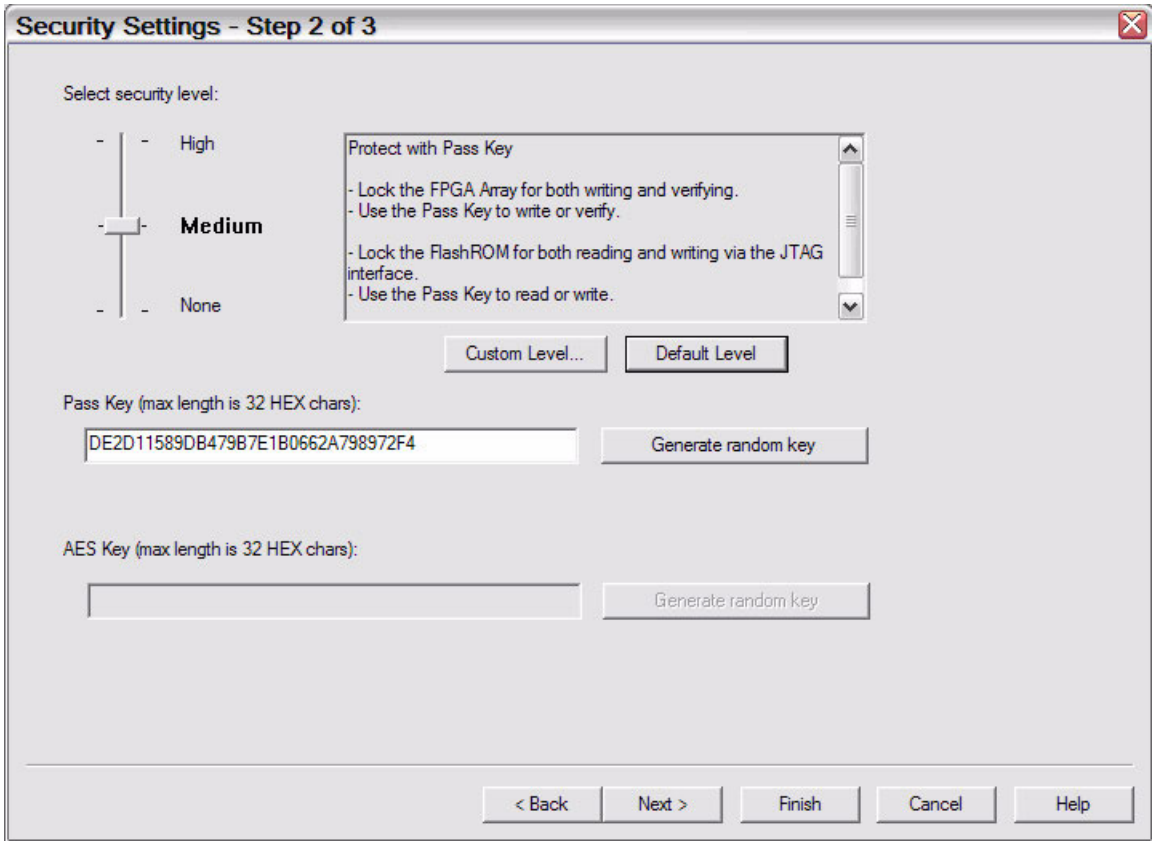


Figure 8 • **Medium Security Level Selected**

3. Choose the desired settings for the FlashROM configurations to be programmed (Figure 9). Click **Finish** to generate the design STAPL programming file.

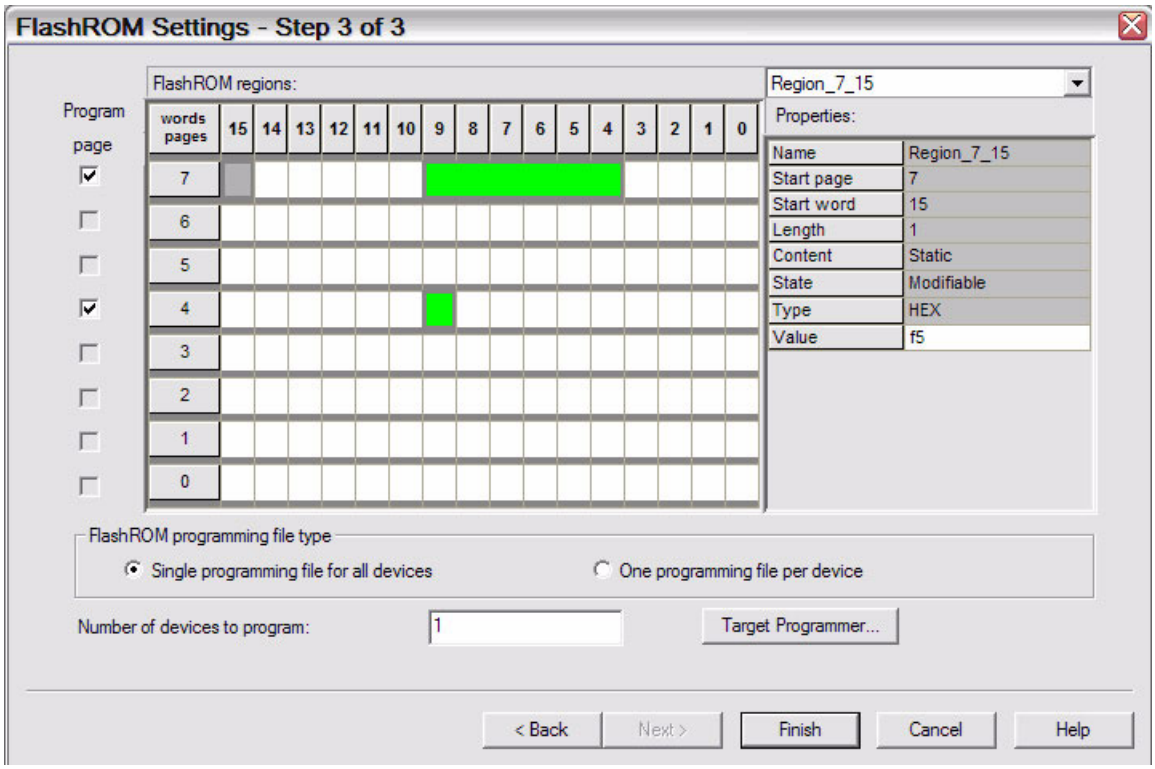


Figure 9 • **FROM Configuration Settings**

Generation of Security Header Programming File Only – Application 2

As mentioned in the "Application 2: Nontrusted Environment – Unsecured Location" section on page 5, the designer may employ FlashLock Pass Key protection or FlashLock Pass Key with AES encryption on the device before sending it to a nontrusted or unsecured location for device programming. To achieve this, the user needs to generate a programming file containing only the security settings desired (Security Header programming file).

Note: If AES encryption is configured, FlashLock Pass Key protection must also be configured.

The security options that can be selected are indicated in Table 2.

Table 2 • **FlashLock Security Options**

Security Option	FROM Core Only	FPGA Core Only	Both FROM and FPGA
No AES / No FlashLock	–	–	–
FlashLock only	✓	✓	✓
AES and FlashLock	✓	✓	✓

Note: *✓ = options that may be used*

The programming file generation for this scenario can be achieved as follows:

1. Check only the **Security Settings** option, as indicated in Figure 10. Click **Next**.

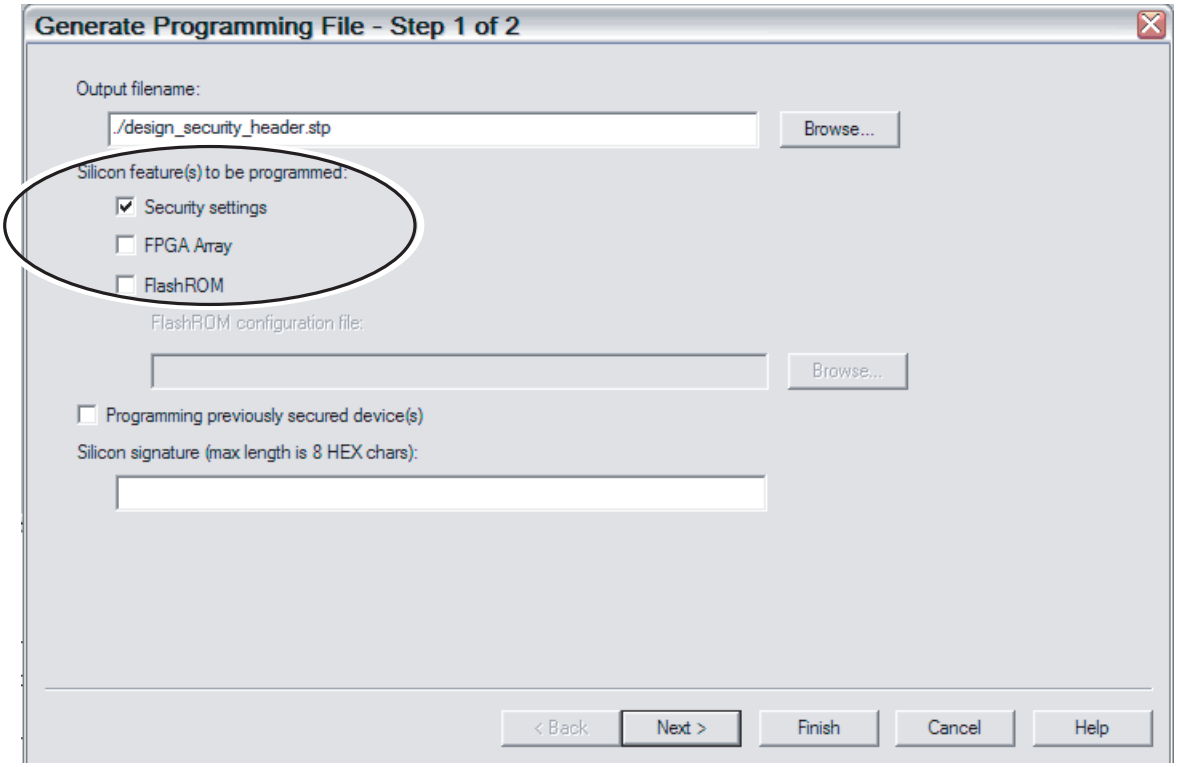


Figure 10 • **Security Settings**

2. Choose the appropriate security level setting desired and enter key(s).
   – A **High** security level employs FlashLock Pass Key with AES Key protection.
   – A **Medium** security level employs FlashLock Pass Key protection only.

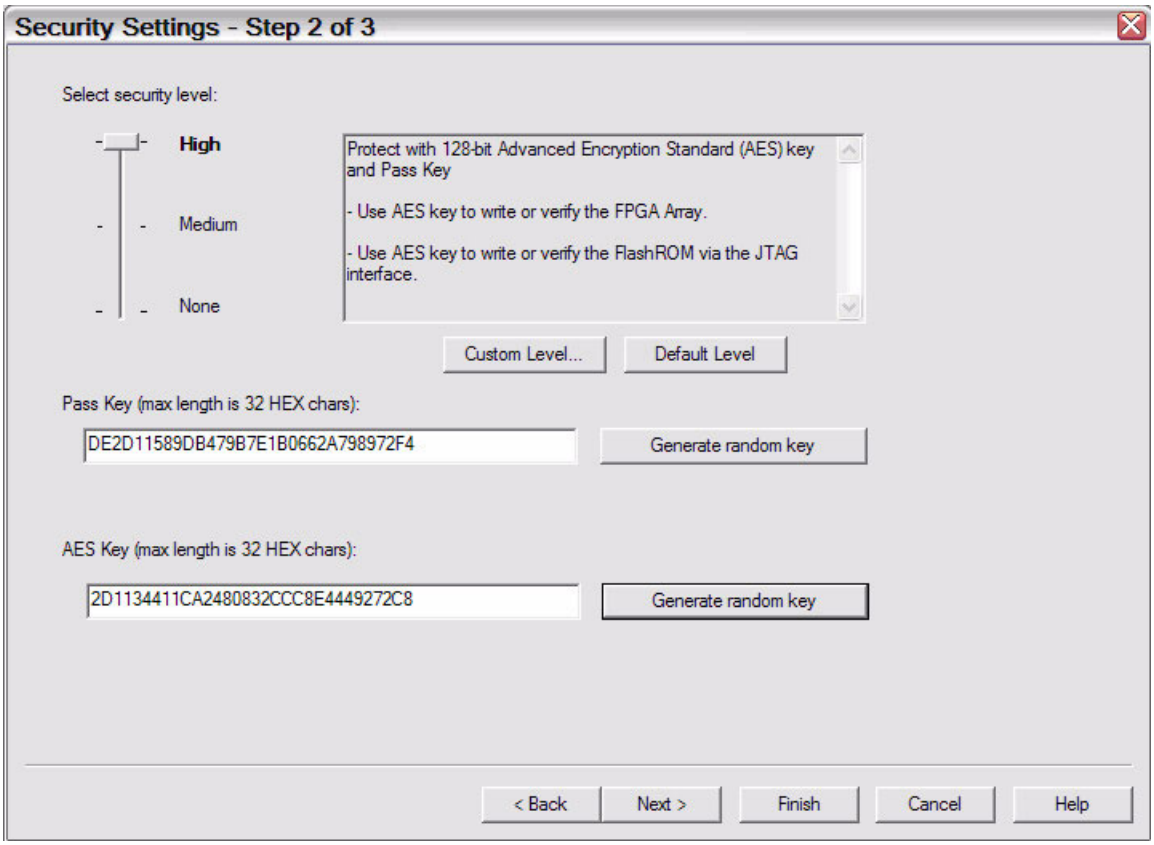


Figure 11 • **Security Level High Selected to Implement Both Flashlock Pass Key and AES Key Protection**

Table 3 shows all options available. If you want to implement custom levels, refer to the "Custom Security Levels" section on page 17 for information on each option and how to set it. When done, click **Finish** to generate the Security Header programming file.

Table 3 • **Header File Security Options**

Security Option	FROM Core Only	FPGA Core Only	Both FROM and FPGA
No AES / No FlashLock	✓	✓	✓
FlashLock only	✓	✓	✓
AES and FlashLock	✓	✓	✓

Note: *✓ = options that may be used*

Reprogramming Devices

A previously programmed ProASIC3/E devices can be reprogrammed using the steps in the "Generation of the Programming File in a Trusted Environment – Application 1" section on page 9 and "Generation of Security Header Programming File Only –Application 2" section on page 11. In the case where a FlashLock Pass Key has been programmed previously, the user must generate the new programming file with a FlashLock Pass Key that matches the one previously programmed to the device. The software will to check the FlashLock Pass Key in the programming file against the FlashLock Pass Key in the device. The keys must match before the device can be unlocked in order to perform further programming with the new programming file.

Figure 7 on page 9 shows the option **Programming previously secured device(s)**, which the user should check before proceeding. Upon going to the next step, the user will be notified that the same FlashLock Pass Key needs to be entered, as shown in Figure 12.

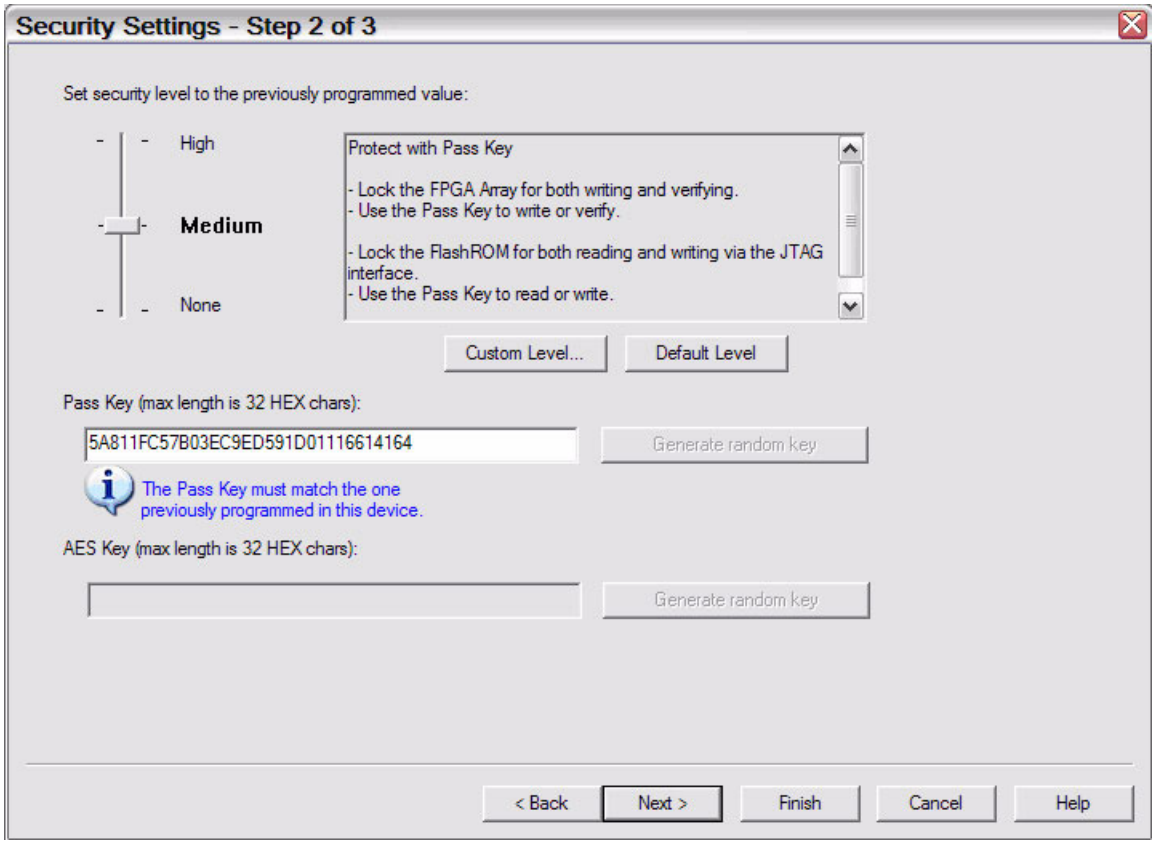


Figure 12 • **FlashLock Pass Key**

It is important to note that when the security settings need to be updated, the user also needs to check the **Security Settings** checkbox in Figure 7 on page 9 to modify the security settings. The user must consider the following:

- If only a new AES key is necessary, the user must re-enter the same Pass Key previously programmed in the device in Designer, and then generate a programming file with same Pass Key and different AES key. This ensures the programming file can be used to access and program the device.
- If a new Pass Key is necessary, the user can generate a new programming file with a new Pass Key (with the same or new AES key if desired). However, for programming, the user must first load the original programming file with the Pass Key that was previously used to unlock the device. Then the new programming file can be used to program the new security settings.

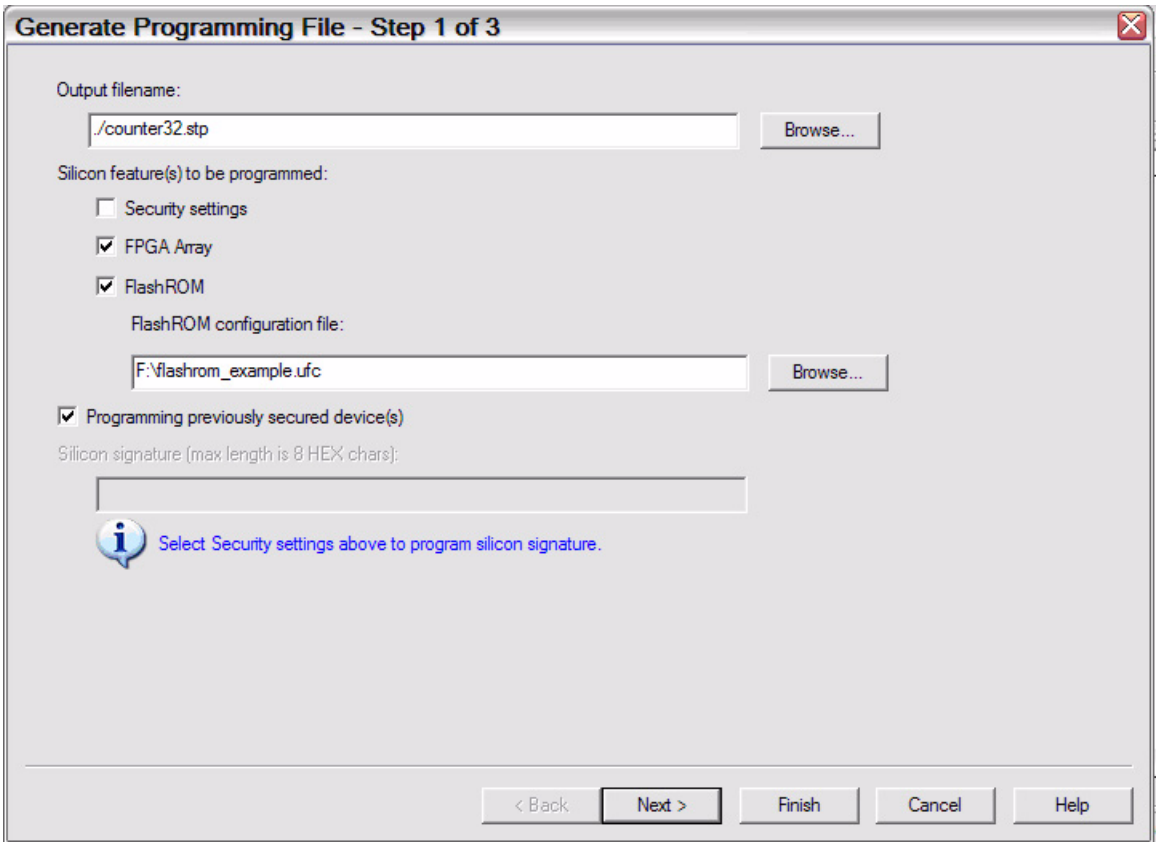
Generation of Programming Files with AES Encryption – Application 3

This section discusses how to generate design content programming files needed specifically at unsecured or remote locations to program device(s) with a security header (FlashLock Pass Key and AES Key) already programmed into them ("Application 2: Nontrusted Environment – Unsecured Location" section on page 5 and "Application 3: Nontrusted Environment – Field Updates/Upgrades" section on page 6). In this case, the encrypted programming file must correspond to the AES key already programmed to the device. If AES encryption was previously selected to encrypt both the FROM and FPGA array, then AES encryption must be set when generating the programming file for both FROM and FPGA arrays. The AES encryption can be on the FlashROM only, FPGA array only, or both. The user must ensure both the FlashLock Pass Key and the AES key match the ones already programmed to the device(s), and all security settings must match what was previously programmed. Otherwise, the encryption and/or device unlocking will not be recognized when attempting to program the device with this programming file.

The generated programming file will be AES-encrypted.

The programming file generation for this scenario can be achieved as follows:

1. Check the **Security Settings** and the portion of the device to be programmed. Check the option **Programming previously secured device(s**). When done, click **Next**.



Note: *This settings in this figure are used to show the generation of an AES-encrypted programming file for both the FPGA array and FROM contents. One or both locations may be selected for encryption.*

Figure 13 • **FlashLock Pass Key and AES Key Protected Programming Files Settings**

Choose the **High** security level to employ both FlashLock Pass Key and AES Key protection. Enter both keys and click **Next**.

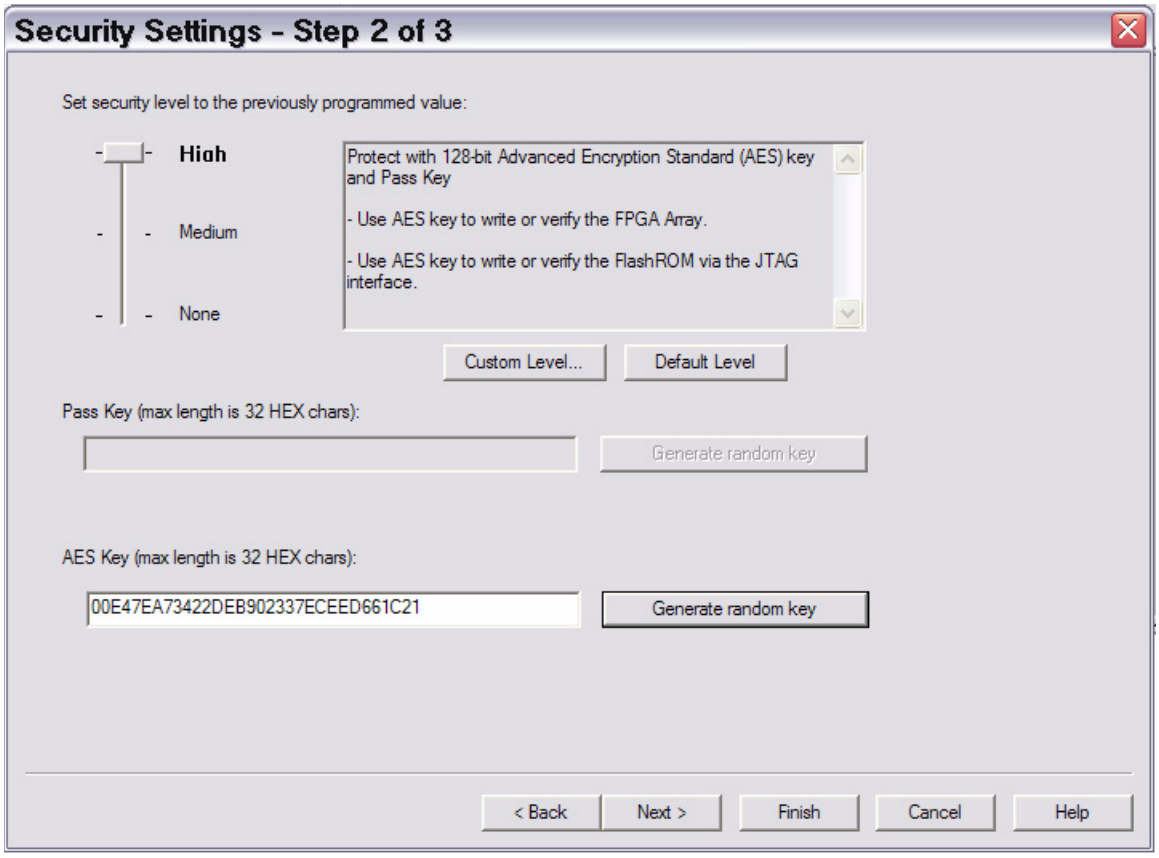


Figure 14 • **Security Level Set High for FlashLock Pass Key and AES Key Protection**

Programming with this file is intended for an unsecured environment. The AES key encrypts the programming file with the same AES key already used in the device, and utilizes it to program the device.

Advanced Options

As mentioned, there may be applications where more complicated security settings are required. The "Custom Security Levels" section on page 17 describes different advanced options available to aid the user in obtaining the best available security settings.

Custom Security Levels

For advanced use, custom security levels are available. Figure 15 shows the options available.

To set custom security levels:

1. Click **Custom Level** in the Setup Security page. The Custom Security dialog box appears.
2. Select the **Security of the FPGA Array** and the **Security of the FlashROM** levels.

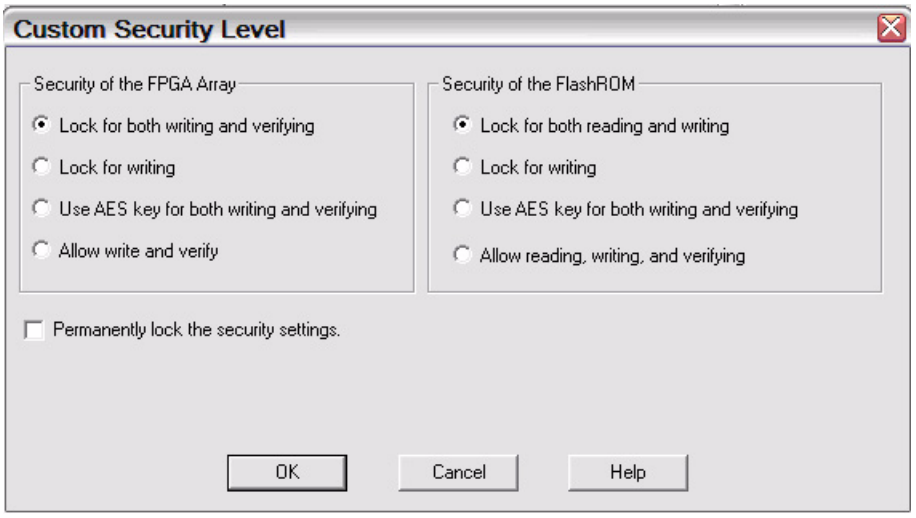


Figure 15 • **Custom Security Level Selected for the FPGA Array and FlashROM**

The FPGA array and the FROM can have different Security Settings. See Table 4 on page 18 for a description of the custom security option levels for FPGA array and FlashROM.

FPGA Array Security Levels

Table 4 • **Available Custom Security Levels for the FPGA Array**

Security Option	Description
Lock for both writing and verifying	Allows writing/erasing and verification of the FPGA array via the JTAG interface only with a valid Pass Key.
Lock for writing	Allows the writing/erasing of the FPGA array only with a valid Pass Key. Verification is allowed without a valid Pass Key.
Use the AES Key for both Writing And Verifying	Allows the writing/erasing and verification of the FPGA array only with a valid AES Key via the JTAG interface. This configures the device to accept an encrypted bitstream for reprogramming and verification of the FPGA Array. Use this option if you intend to complete final programming at an unsecured site or if you plan to update the design at a remote site in the future. Accessing the device security settings requires a valid Pass Key. The bitstream that is read back from the FlashROM is always unencrypted (plain text).
Allow write and verify	Allows writing/erasing and verification of the FPGA array with plain text bitstream and without requiring a Pass Key or an AES Key. Use this option when you develop your product in-house.

Note: *ProASIC3/E devices FPGA arrays are always read protected regardless of the Pass Key or the AES Key protection.*

FlashROM Security Levels

Table 5 • **Available Custom Security Levels for FlashROM**

Security Option	Description
Lock for both reading and writing	Allows the writing/erasing and reading of the FlashROM via the JTAG interface only with a valid Pass Key. Verification is allowed without a valid Pass Key.
Lock for writing	Allows the writing/erasing of the FlashROM via the JTAG interface only with a valid Pass Key. Reading and verification is allowed without a valid Pass Key.
Use the AES Key for both writing and verifying	Allows the writing/erasing and verification of the FlashROM via the JTAG interface only with a valid AES Key. This configures the device to accept an encrypted bitstream for reprogramming and verification of the FlashROM. Use this option if you complete final programming at an unsecured site or if you plan to update the design at a remote site in the future. The bitstream that is read back from the FlashROM is always unencrypted (plain text).
Allow writing and verifying	Allows writing/erasing, reading and verification of the FlashROM content with a plain text bitstream and without requiring a valid Pass Key or an AES Key.

Figure 16 gives an example of the security settings used in each device.

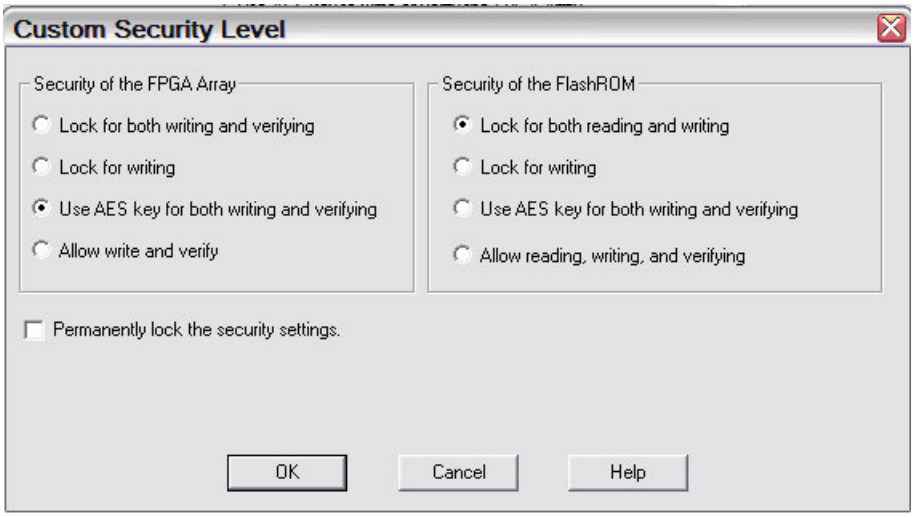


Figure 16 • **Custom Security Settings**

Security settings for FPGA Array:

- Protect FPGA Array content with AES encryption
- Use AES key to write or verify the FPGA Array

Security settings for FlashROM:

- Lock the FlashROM for both reading and writing via the JTAG interface.
- Use the Pass Key to read or write
- No AES encryption on the FlashROM content

With the available security options, settings suitable for any of a number of different applications can be selected, providing the best security to the device possible.

Implementing Permanent Security Settings in the Device

If it is necessary to make the security settings permanent, the user can do so in any of the above scenarios. The option **Permanently lock the security settings** in the Custom Level setting must be checked.

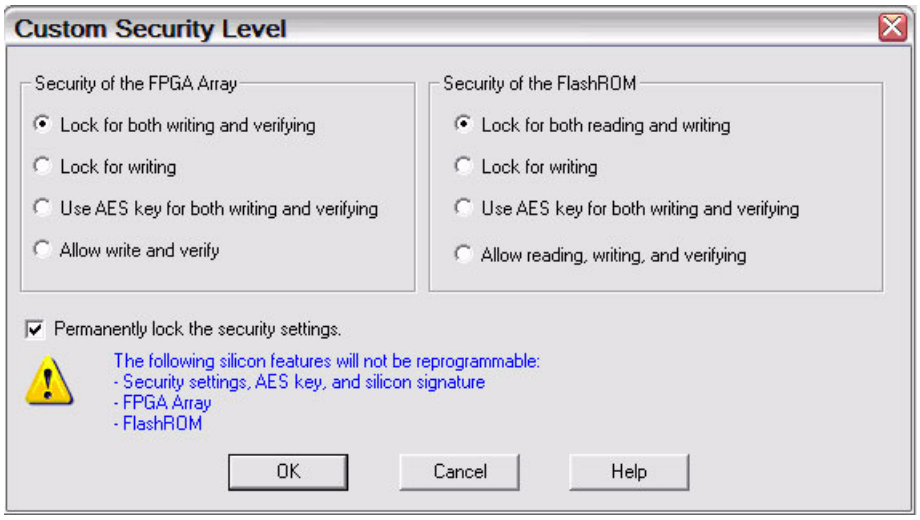


Figure 17 • **Permanently Locking the Security Settings**

Implementing Permanent FlashLock

To use the Permanent FlashLock feature, select the following:

- **Lock for both writing and verifying** for FPGA Array
- **Lock for both reading and writing** for FlashROM
- **Permanently lock the security settings**

This will generate a programming file to program the device and make it one-time-programmable.

Generated Programming File Header Definition

In each STAPL programming file generated, there will be information about how the AES key and the FlashLock Pass Key are configured. Table 6 shows the header definitions in STAPL programming files for different security levels.

Table 6 • **STAPL File Header Definitions for Various Security Levels**

Security Level	STAPL File Header Definition
No security (no FlashLock Pass Key and AES key)	`NOTE "SECURITY" "Disable";`
FlashLock Pass Key with no AES key	`NOTE "SECURITY" "KEYED ";`
FlashLock Pass Key with AES Key	`NOTE "SECURITY" "KEYED ENCRYPT ";`

Conclusion

The new and enhanced security features offered in Actel ProASIC3/E devices provide state-of-the-art security to designs programmed into these Flash-based ProASIC3/E devices. Actel ProASIC3/E devices employ the encryption standard used by NIST and the U.S. government – AES using the 128-bit Rijndael algorithm.

The combination of an on-chip AES decryption engine and Actel FlashLock technology provides the highest level of security against invasive attacks and design theft, implementing the most robust and secure ISP solution. These security features protect intellectual property within the FPGA as well as protecting the system from cloning, wholesale "black box" copying of a design, invasive attacks, or explicit IP or data theft.

Glossary

Term	Explanation
Security Header Programming File	Programming file used to program the AES key and/or FlashLock Pass Key in the FROM and/or FPGA array core of the device.
AES (Encryption) Key	128-bit key defined by the user when the AES encryption option is set inside Actel Designer software when generating the programming file.
FlashLock Pass Key or Pass Key	128-bit key defined by the user when the FlashLock option is set inside Actel Designer software when generating the programming file. The user must enter this FlashLock Pass Key to the programming software before programming (and actual decryption if AES option is configured) is done.
FlashLock	Security features that protect the device content from attacks. This is achieved by the following: • Flash technology which does not require external bitstream to be programmed to the device • FlashLock Pass Key that secures device content and prevents access to the device as necessary • AES Key that allows secure device reprogrammability

Reference

National Institute of Standards and Technology. "ADVANCED ENCRYPTION STANDARD (AES) Questions and Answers." 28 January 2002. <http://csrc.nist.gov/CryptoToolkit/aes/aesfact.html> (10 January 2005).

Related Documents

Application Notes

ProASIC3/E FlashROM (FROM)
http://www.actel.com/documents/PA3_E_FROM_AN.pdf

Programming ProASIC3/E Using a Microprocessor
http://www.actel.com/documents/PA3_E_Microprocessor_AN.pdf

In-System Programming (ISP) in ProASIC3/E Using FlashPro3
http://www.actel.com/documents/PA3_E_ISP_AN.pdf

http://www.actel.com

Actel Corporation

2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.

Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Phone +44 (0) 1276 401 450
Fax +44 (0) 1276 401 490

Actel Japan
www.jp.actel.com

EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668

Actel Hong Kong
www.actel.com.cn

Two Pacific Place
88 Queensway, Admiralty
Hong Kong
Phone +852.2185 6460
Fax +852.2185 6488

51900052-1/1.05