Advanced Configuration Security Usage Guide for Mach-NX

Technical Note

FPGA-TN-02266-1.0

October 2021

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults and associated risk the responsibility entirely of the Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AES	Advanced Encryption Standard
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ESB	Embedded Security Block
FPGA	Field Programmable Gate Array
JTAG	Joint Test Action Group
ROT	Root of Trust
SHA	Secure Hash Algorithm
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory

1. Introduction

Security is a major concern in various fields due to the increasing number of cyberattacks. To combat security risks, Lattice Semiconductor devices offer advanced security functions that allow you to easily embed state-of-the-art features into security sensitive applications. Hardened architectures and intuitive interfaces allow you to take advantage of these features without knowing esoteric details of the security algorithms.

The Lattice Semiconductor Mach™-NX device family is a next generation FPGA with enhanced security features. Mach-NX devices contain a full suite of enhanced security algorithms and features, including 128-bit/256-bit Advanced Encryption Security (AES-128/AES-256), 256-bit Secure Hash Algorithm (SHA256), and Elliptic Curve Digital Signature Algorithm (ECDSA). With these enhanced bitstream security functions, Mach-NX devices can be used as a Root-of-Trust (ROT) hardware solution in a complex system.

2. Bitstream Security Overview

Mach-NX devices offer enhanced security features for advanced bitstream security options, such as encryption and authentication. The read and write lock access, and password protection are also supported.

To ensure the security and authenticity of the configuration bitstream, Mach-NX devices offer the following features:

- Bitstream Encryption
- Bitstream Authentication
- Bitstream Encryption and Authentication

The above-mentioned security settings can be accessed using the Security Setting tool of the Lattice Diamond® software (Figure 2.1).

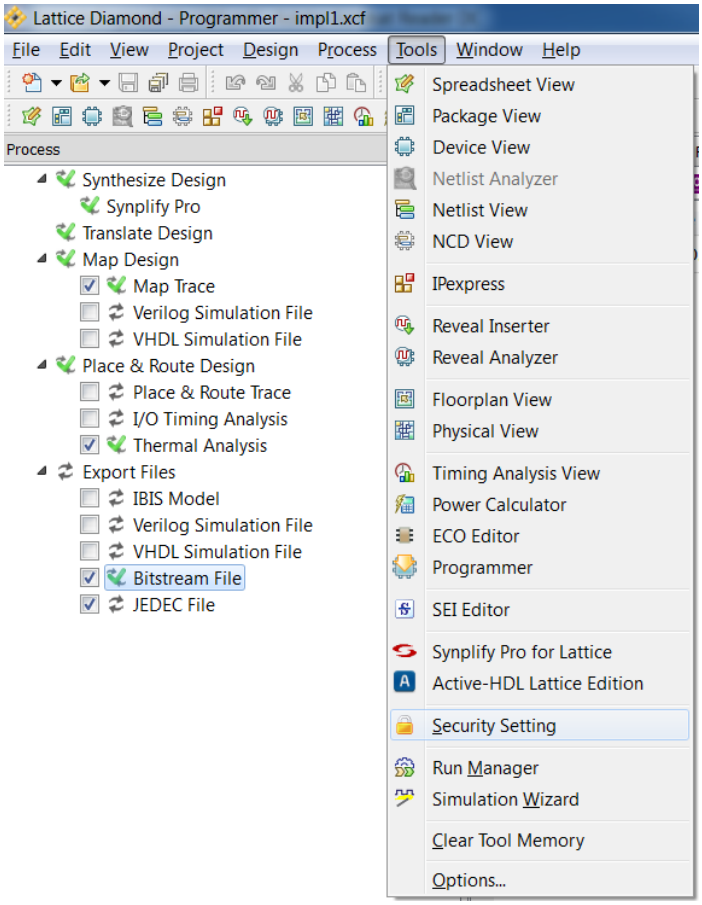

Figure 2.1. Security Setting Tool in the Lattice Diamond Software

Table 2.1. Mach-NX Security Settings for Configuration Bitstreams

Security Protocol	Encryption (AES-256)	Authentication (ECDSA)	Bitstream Format
AES Encryption	Yes, AES key	No	Encrypted Bitstream
Authentication (ECDSA)	No	Yes, ECDSA private and public key	Plain bitstream + Signature generated using ECDSA private key
Authentication (ECDSA) and AES Encryption	Yes, AES key	Yes, ECDSA private and public key	AES Encrypted data (Plain bitstream + Signature generated using ECDSA private key)

The Security Setting Tool has options for bitstream security and password protection as shown below.

- From the Security Settings tab (Figure 2.2), there are options for:
  - Flash Protection
  - AES Encryption
  - ECDSA Authentication

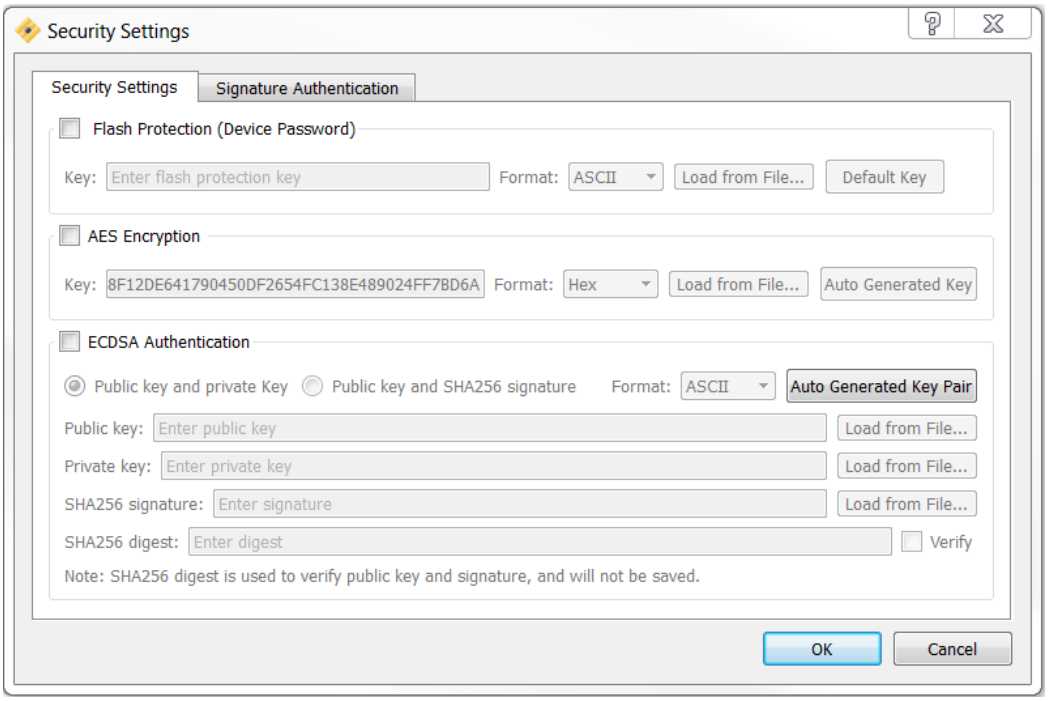

Figure 2.2. Security Settings Options

- From the Signature Authentication tab (Figure 2.3), there are options for:
  - Signature Generation
  - Signature Verification

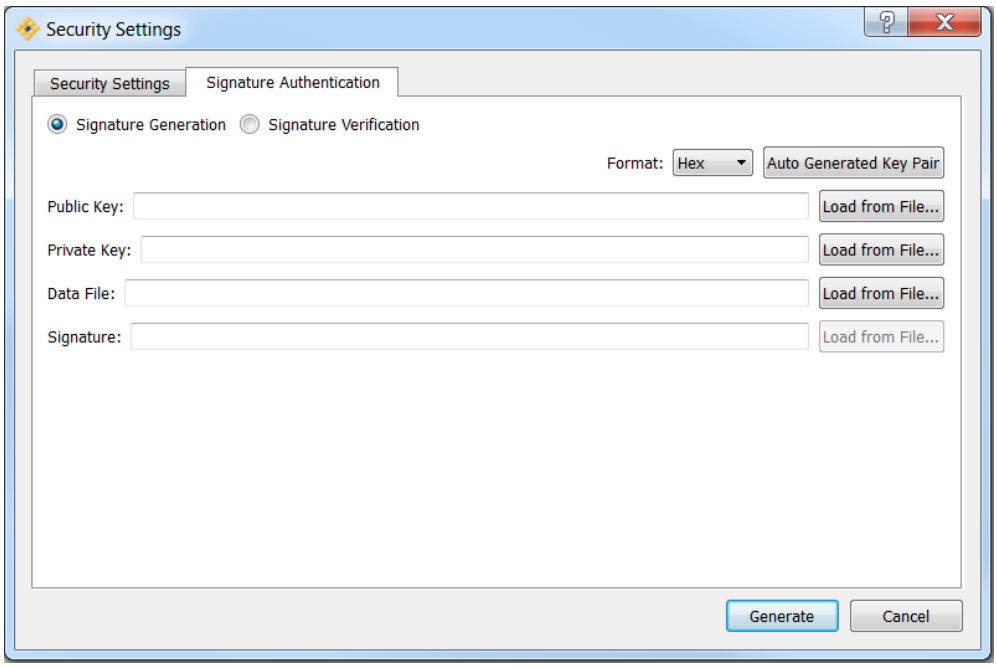

Figure 2.3. Signature Authentication Options

3. Bitstream Encryption

Mach-NX device supports bitstream encryption by using 256-bit Advanced Encryption Standard (AES-256). Figure 3.1 shows the process of bitstream encryption and decryption supported by Mach-NX devices.

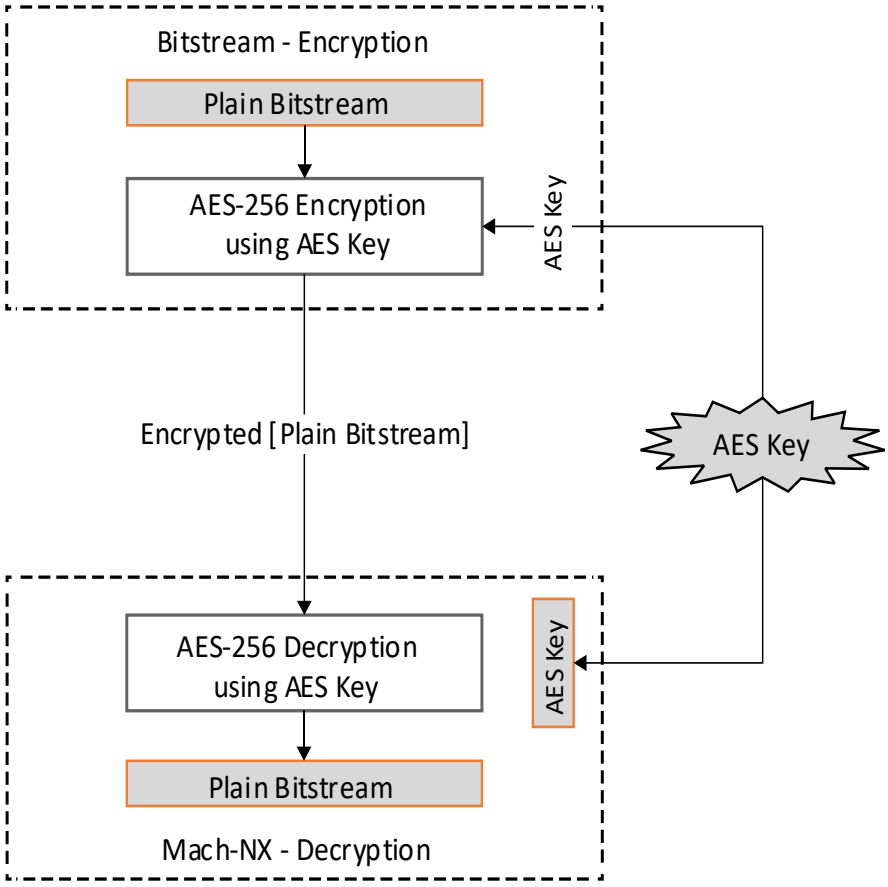

Figure 3.1. Mach-NX Bitstream Encryption and Decryption

3.1. Security Settings to Enable Bitstream Encryption

To enable bitstream encryption:

1. Open the Security Setting tool from Lattice Diamond software (Figure 2.1).
2. Select the AES Encryption option (Figure 2.2). There are two options for the AES key loading:
   - Load from File: this option is used to select a file that contains the 256-bit AES key.
   - Auto Generated Key: this option is used to automatically generate a 256-bit AES key file.
3. Click OK to complete the configurations.

If the Auto Generated Key option is selected, you can find the auto-generated AES key (.bek file) in the security_setting folder. Rerun the process of synthesis, translate, map, and PAR to access the encrypted bitstream files (.bit file and .jed file) in the impl1 folder.

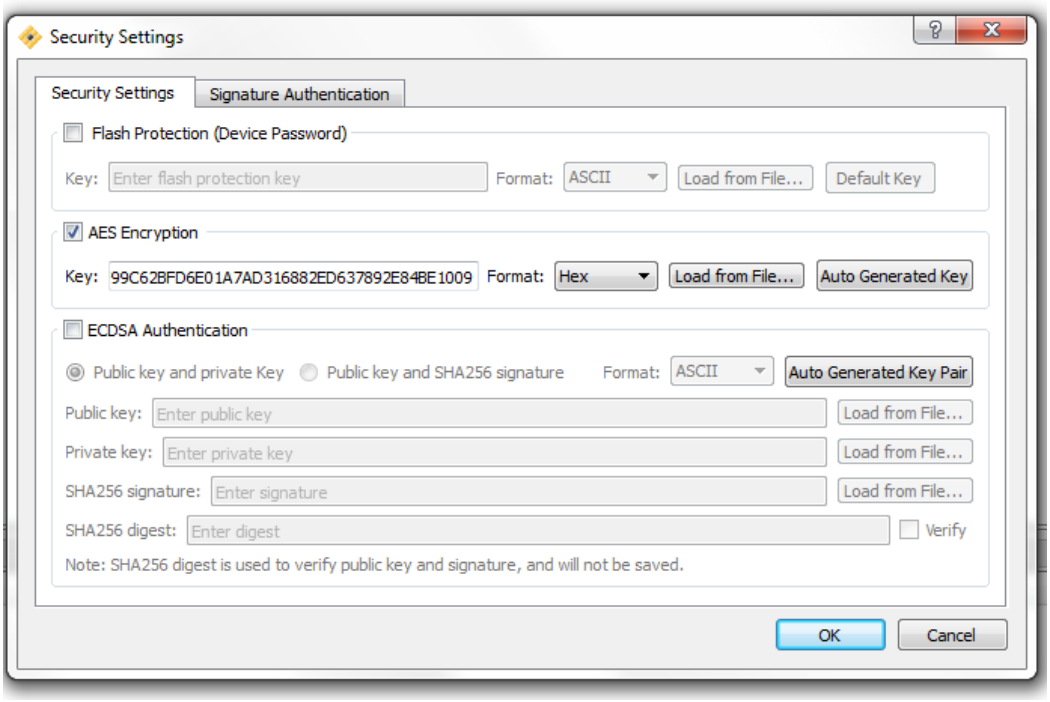

Figure 3.2. Bitstream Encryption for AES-256 Key

3.2. Programming Encrypted Bitstream File and AES-256 Key File

Both the encrypted bitstream file and the AES-256 Key file need to be programmed into the Mach-NX device. The encrypted bitstream file can be programmed into the SRAM, internal flash or external flash using Lattice Diamond Programmer or through embedded programming such as JTAG and SPI.

To program the AES-256 key file, you need to use advanced security commands in Lattice Diamond Programmer as shown in Figure 3.3 below. To program the AES-256 key into the Mach-NX device, make the following selection:

- Access mode: Advanced Security Keys Programming
- Operation: Security Program Encryption Key
- Encryption Key: Load the AES-256 key used for bitstream encryption

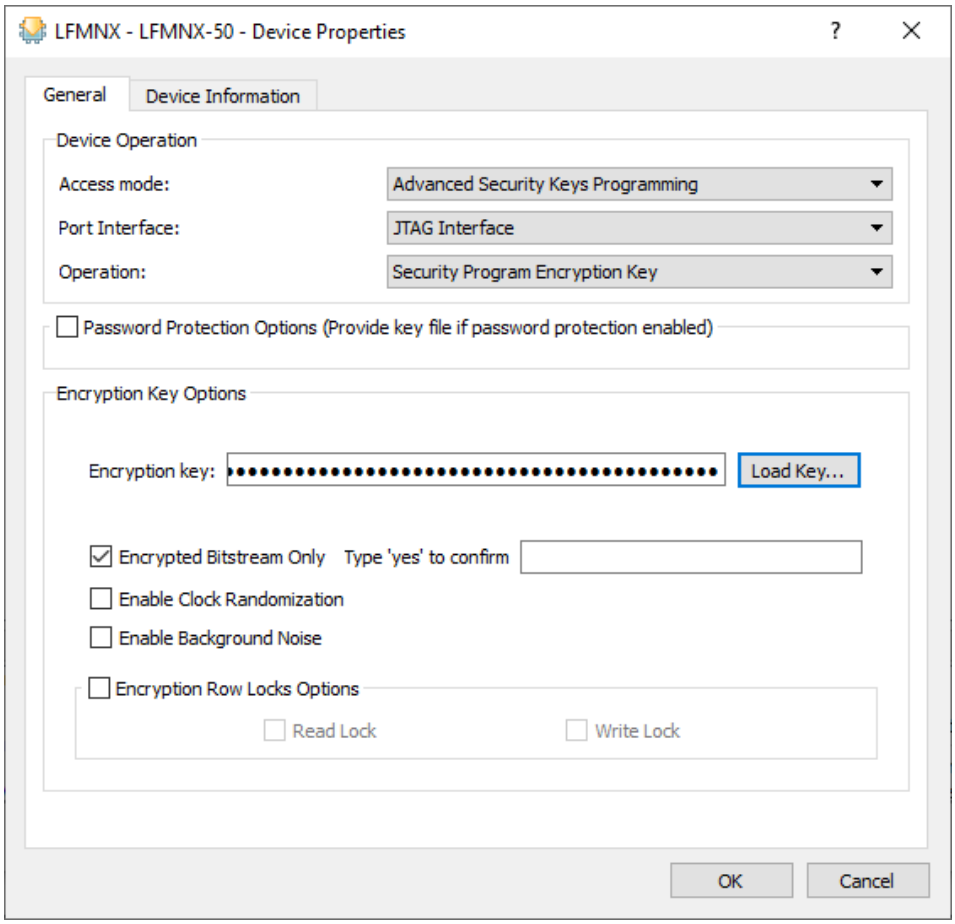

Figure 3.3. Programming the AES-256 Key into the Mach-NX Device

When the Encrypted Bitstream Only option is selected, only the encrypted bitstream is acceptable. With this option selected, booting from the plain bitstream file fails. If this option is not selected, both the encrypted bitstream and the plain bitstream can be acceptable.

After the encrypted bitstream file and the AES-256 key file are programmed, AES-256 engine starts to detect a pre-defined preamble from the bitstream bytes during booting. Once the encrypted preamble is detected, the following bytes in the bitstream are treated as encrypted and are sent to AES-256 engine to perform decryption using the AES-256 key stored in the Mach-NX device.

4. Bitstream Authentication

The Elliptic Curve Digital Signature Algorithm (ECDSA) offers a variant of the Digital Signature Algorithm (DSA) that uses elliptic curve cryptography. The private or public key pair is used to generate a digital signature for the bitstream using ECDSA algorithm. Mach-NX device supports authentication of internal and external bitstream. Figure 4.1 shows the process of bitstream authentication.

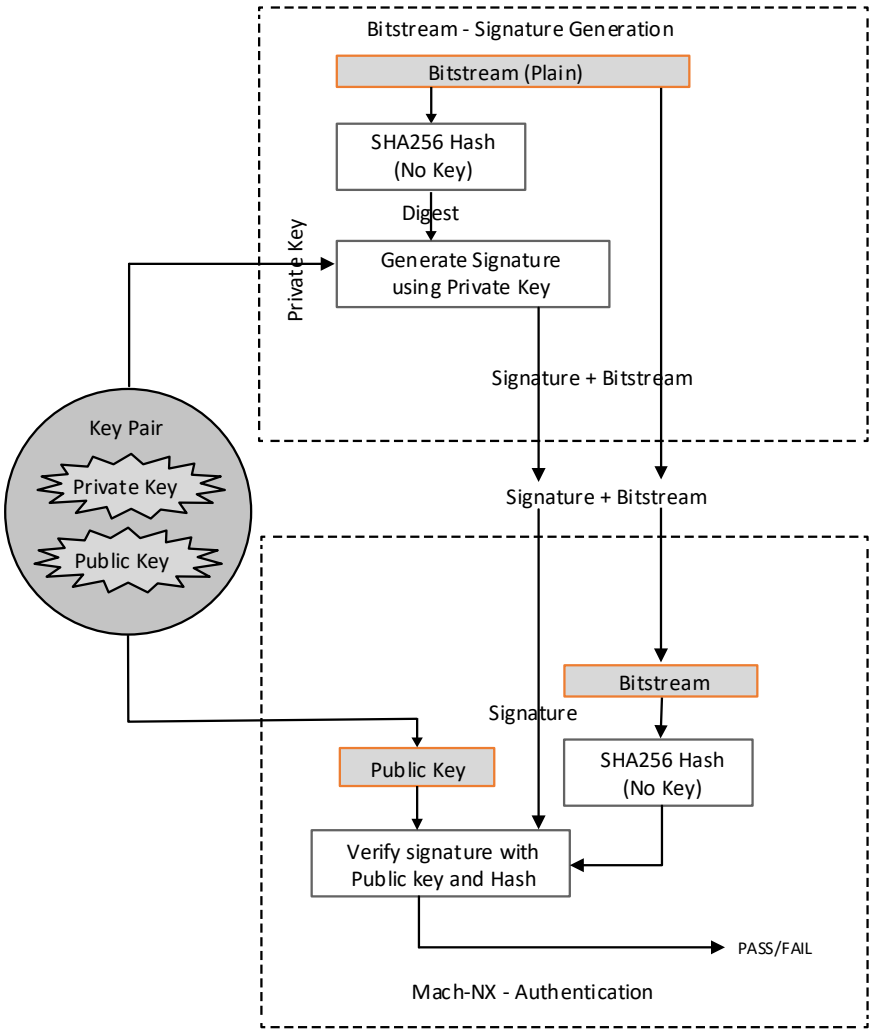

Figure 4.1. Bitstream Authentication

4.1. Security Settings to Enable Bitstream Authentication

You can enable the bitstream authentication by using the Security Setting tool from Lattice Diamond software.

To enable the bitstream authentication:

1. Open the Security Setting tool from Lattice Diamond software (Figure 2.1).
2. Select the ECDSA Authentication option (Figure 2.2).
3. To auto generate the public key and private key, select the Public key and private key option.
4. Click the Auto Generated Key Pair button.
5. Click OK.

The auto-generated public key file and private key file are automatically saved in the security_setting folder under the impl1 directory.

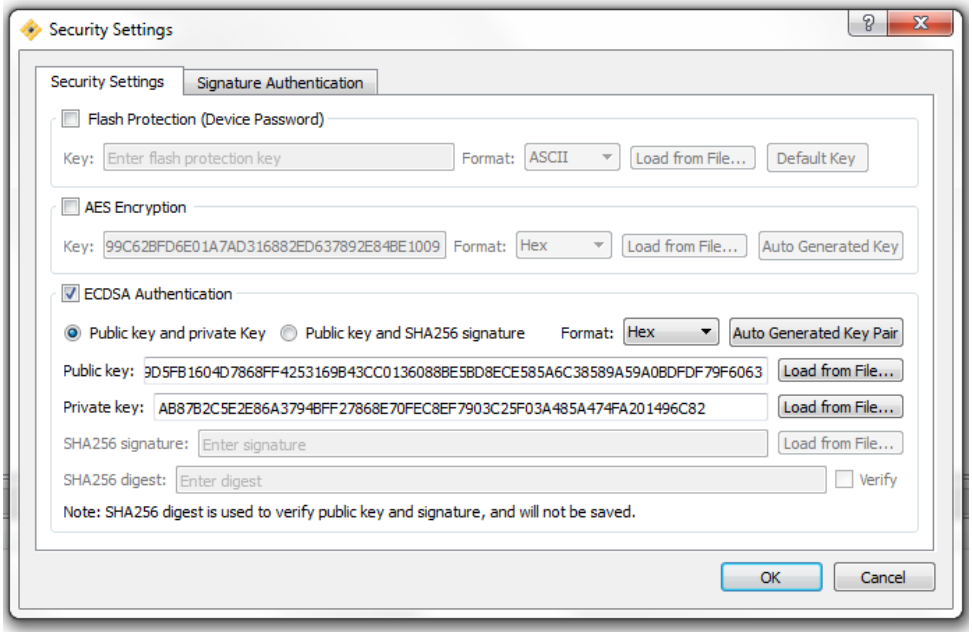

Figure 4.2. Bitstream ECDSA Authentication Selection

Rerun the process of synthesis, translate, map and PAR, to generate the bitstream file, .bit file and .jed file, containing both the signature and the plain bitstream in the impl1 folder. The digital signature is generated using the SHA256 message digest (digest of the plain bitstream using SHA256 Hash) and a private key.

4.2. Programming Bitstream File and Public Key File

Both the bitstream file and the public key file need to be programmed into the Mach-NX devices. The bitstream programming is similar to a normal bitstream programming.

For programming of a public key, the Diamond Programmer requires the configuration as shown in Figure 4.3.

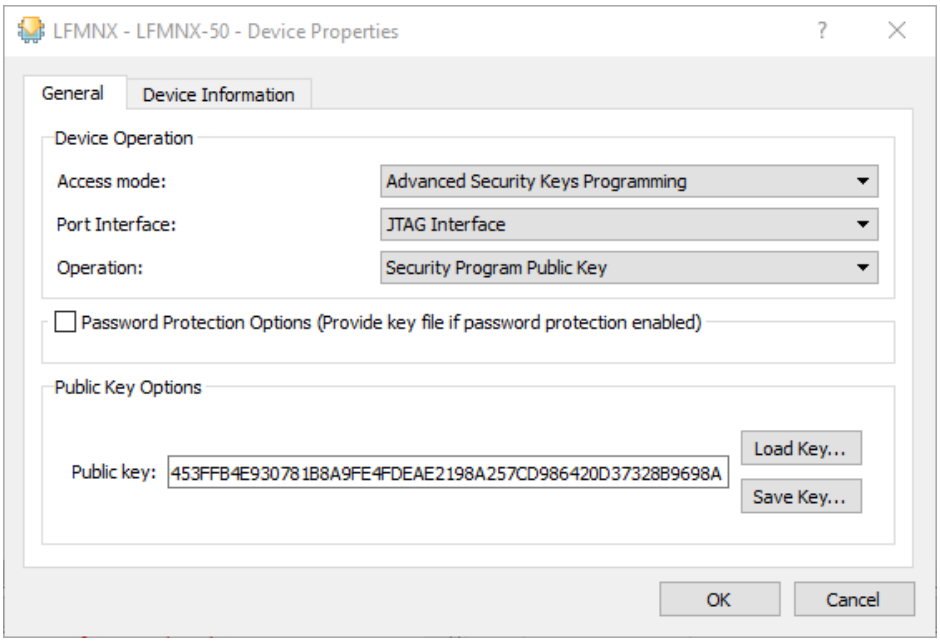

Figure 4.3. Configuration Options for the Public Key Programming

To program a public key:

1. Configure settings in Diamond programmer as shown in Figure 4.3.
2. Load the public key file by checking the Public Row Local Locks Options, Lock Read Access, and Lock Write Access options (Figure 4.3) to protect the public key from being written and read.
3. Click OK to complete the configuration.

Then program the public key into the Mach-NX device, as shown in Figure 4.4.

Figure 4.4. Programming the Public Key

After programming the bitstream and the public key into the Mach-NX device in the Diamond Programmer, the authentication is enabled and is started upon each booting.

4.3. Verification of Bitstream Authenticity

To verify the authenticity of the programmed bitstream, the Embedded Security Block (ESB) calculates the message digest using the SHA256 message digest and generates a digital signature using the public key that is programmed in the Mach-NX device. In the ECDSA verification process, this calculated digital signature is compared with the stored digital signature to verify the authenticity of the read bitstream. If both signature match with each other, the authentication passes. Otherwise, fails.

After the authentication passes, the bitstream is determined to be legal and the wake-up sequence is invoked to wake up the device, and then switch the device to user mode. If authentication fails, based on the boot up settings, configuration follows the boot sequence to boot from another bitstream (dual-boot mode), or just leaves the device in unprogrammed state (single-boot mode). In dual boot mode, if the authentication fails in the secondary boot, configuration leaves the device in unprogrammed state.

5. Bitstream Authentication and Encryption

Mach-NX devices can combine the bitstream authentication and encryption together to provide a highly secured programming and configuration option. Figure 5.1 shows the process of the bitstream authentication and encryption, which is a two-step process:

1. The ECDSA algorithm generates a digital signature of the plain bitstream.
2. This digital signature along with the plain bitstream is encrypted using the AES-256 encryption.

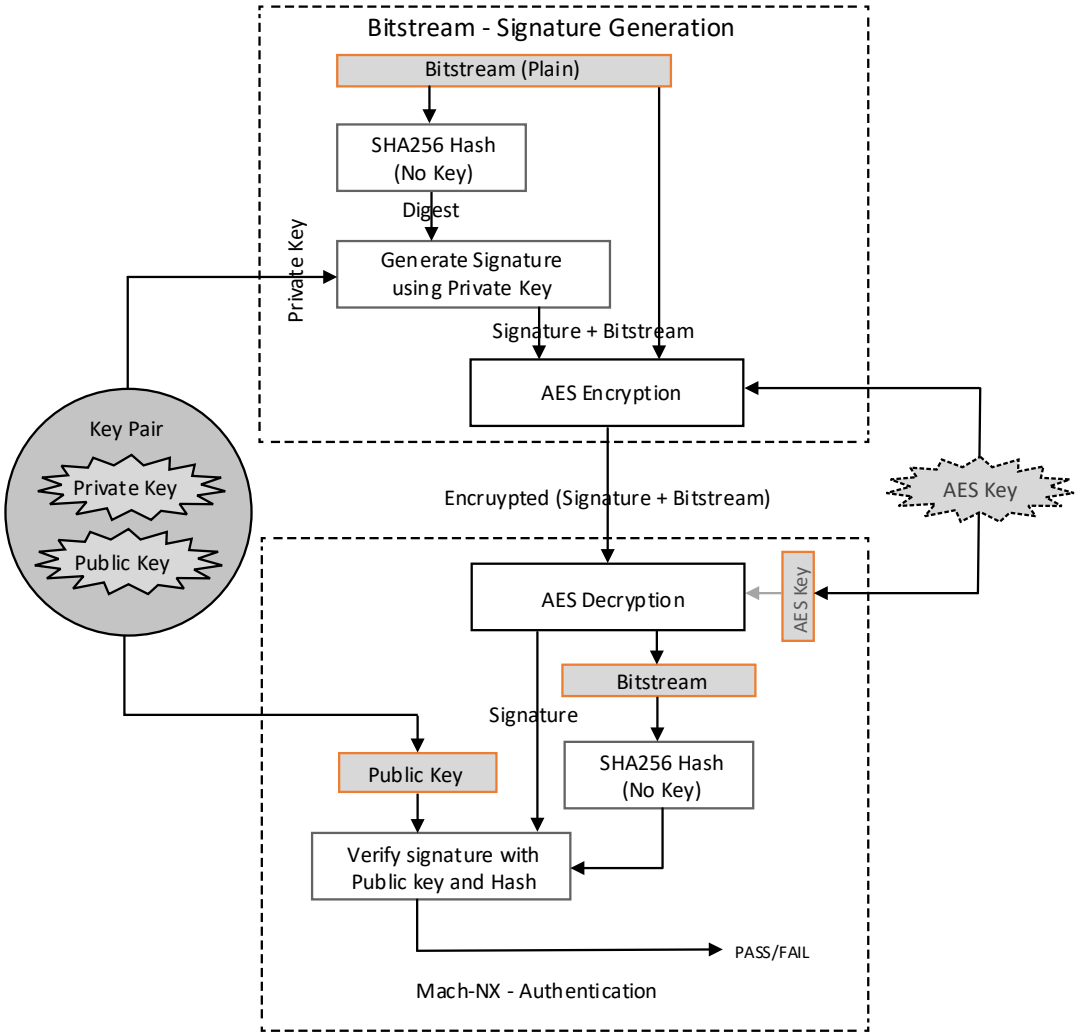

Figure 5.1. Bitstream Authentication and Encryption

5.1. Security Settings to Enable Bitstream Authentication and Encryption

To enable both the bitstream authentication and encryption in the Security Setting tool of Lattice Diamond software:

1. Select both AES Encryption option and ECDSA Authentication option, as shown in Figure 5.2.

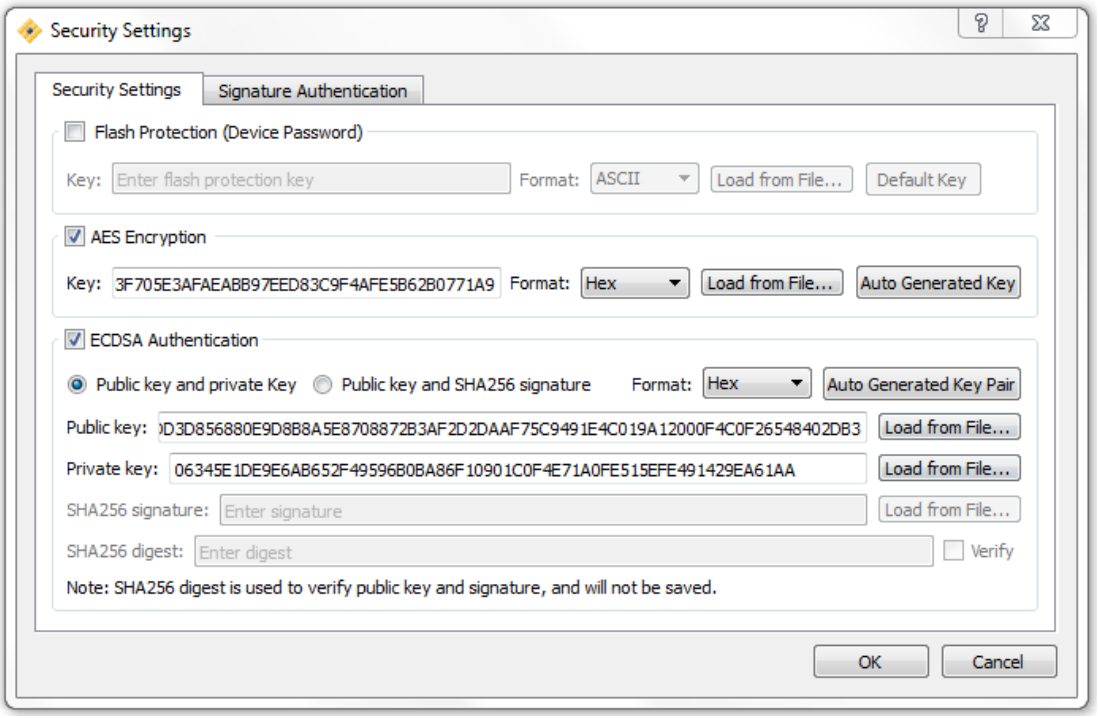

Figure 5.2. Configuration for the Bitstream Authentication and Encryption

2. Click OK to complete the configuration.
3. Rerun the process of synthesis, translate, map, and PAR, to generate the encrypted bitstream file in the impl1 folder.

5.2. Programming Bitstream File and Public Key/Private Key File

After programming the encrypted bitstream file in Lattice Diamond Programmer, the public key and the private key are programmed into the Mach-NX device. Refer to previous Bitstream Encryption section and Bitstream Authentication section. Bitstream authentication and encryption is enabled and is started upon each booting.

References

This is a list of related documents that are available from your Lattice Semiconductor sales representative.

- Mach-NX Device Family Data Sheet (FPGA-DS-02084)
- Mach-NX Programming and Configuration Usage Guide (FPGA-TN-02231)
- Lattice Sentry Demo Board for Mach-NX (FPGA-EB-02045)

Technical Support

For assistance, submit a technical support case at www.latticesemi.com/techsupport.

Revision History

Revision 1.0, October 2021

Section	Change Summary
All	Production release.

www.latticesemi.com