# AES Keywrap Core, Xilinx Edition

August 1, 2008 Product Specification

## Core Facts

| Provided with Core | |
|---|---|
| Documentation | User Manual |
| Design File Formats | VHDL or EDIF Netlist |
| Verification | Test Bench, Test Vectors |
| Instantiation templates | VHDL |
| **Simulation Tool Used** | |
| Model Tech ModelSim XE, Aldec Active HDL | |
| **Support** | |
| Support provided by Algotronix | |

**Algotronix®**
PO Box 23116
Edinburgh, Scotland
United Kingdom, EH8 8YB
Phone: +44 131 556 9242
Fax: +44 870 052 5069
E-mail: cores@algotronix.com
URL: www.algotronix.com

## Features

- Provides a secure method of transferring keys into the Algotronix AES G2 core
- Implements IETF RFC 3394 and the NIST AES Keywrap specification specified as an Approved Key Establishment Technique in FIPS 140-2
- Supports 128, 192 and 256 bit keys
- Compile as Wrap, Unwrap or Wrap/Unwrap
- Supplied as portable VHDL to allow customers to conduct their own code review in high-security applications
- Supplied with test bench implementing all vectors from the IETF and NIST specification.

## Applications

- High security applications requiring FIPS140-2 level 3 or 4 certification
- Key guns
- Government/Military Communications
- Disk Drive Security IEEE P1619
- Financial Security ANSI X9.53

## General Description

This core extends the Algotronix G2 Advanced Encryption Standard (AES) core by providing a secure, standardized method for loading cryptographic keys into the encryption unit. The core is in use by customers in production equipment.

The Keywrap algorithm was developed by National Institute of Standards and Technology (NIST) and was adopted by the Internet Engineering Task Force (IETF) as RFC 3394. Use of the AES Keywrap algorithm is also specified in the W3C XML Security Specification and in the IEEE draft standard for encryption of data on hard disks (IEEE P1619).

In order to meet specifications such as FIPS140-2 level 2 and higher or the equivalent Common Criteria Protection Profiles protection must be provided for Critical Security Parameters (CSPs) such as cryptographic keys entering the encryption unit. The Key-Wrap algorithm provides an approved mechanism for cryptographically protecting CSPs such that they can enter the security module through unprotected channels. FIPS 140-2 specifies that for level 3 or 4 security “Secret and private keys established using manual methods shall be entered or output encrypted or with split knowledge procedures” . The AES Keywrap algorithm is one of two approved key establishment techniques for symmetric keys (see FIPS 140-2 Annex D).

KeyWrap normally operates on relatively short sequences of data and from a performance viewpoint could easily be implemented in software. However, a hardware implementation, integrated with the main hardware encryption engine localizes and protects the sensitive key information making it easier to analyze the security of the overall system. In particular, a hardware implementation can be designed to shield the Critical Security Parameters from the system software ensuring that any faults in the software or software based attack vectors (such as viruses) cannot compromise the keys.

To protect cryptographic keys entering or leaving the security module Keywrap introduces the concept of ‘Key Encryption Key’ or KEK and a new ‘Keywrap’ mode of operation of the cipher. The KEK is stored within the cryptographic module and used to encrypt ‘Traffic Keys’ which can then be transferred into the encryption unit over an insecure channel.

The Algotronix Keywrap core is designed as a ‘wrapper’ which extends the functionality of the G2 AES core by implementing the KeyWrap mode of operation while maintaining all the functionality of the G2 AES core and a very similar system level interface. The KeyWrap core is designed for maximum flexibility with a simple interface that is compatible with many system architectures. A higher level, register based interface which provides storage for the various keys and initial values and is designed for connection to a microprocessor is provided as a reference design. Algotronix can create a customized interface for the core based on this reference design to suit the exact requirements of the user’s application.

The Algotronix AES G2 core is a complete validated implementation of AES: all standard modes of operation and key lengths are supported. The core is developed in accordance with Federal Information Processing Standards Publication (FIPS PUB 197) “*Advanced Encryption Standard* (AES)” and tested in accordance with the NIST document “*The Advanced Encryption Standard Algorithm Validation Suite* (AESAVS)”, November 15, 2002. The modes of operation are developed in accordance with the NIST document SP800-38A. The core was validated by a NIST approved laboratory in March 2006 and received certificate number 347 from NIST. This document on the AES Keywrap core should be read in conjunction with the Product Description for the G2 core.

The Kewrap testbench implements all the tests in the NIST Keywrap specification, the product is also supplied with the AES G2 core testbench for the AES encryption unit. This is a comprehensive test bench implementing all the NIST Known Answer Tests, Monte Carlo Tests, and the Multi-block Message tests.

Table 1 shows example implementation statistics for a 256 bit encryption/decryption core with keyschedule calculation in hardware supporting Key Unwrap and ECB and CBC mode only. This represents a reasonable configuration for use of AES Keywrap. The AES-Keywrap core can be targetted at all Xilinx FPGA families, including older devices not listed in the table below.

**Table 1: Example Implementation Statistics for AES-Keywrap, 'Push Button' flow with Fmax specified by clock constraint. CBC mode, Encrypt/Decrypt, 256 Bit Key**

| Family | Example Device | Fmax (MHz) | Slices[1] | IOB[2] | GCLK[2] | BRAM | MULT/ DSP48 | DCM | Design Tools |
|---|---|---|---|---|---|---|---|---|---|
| Spartan-3E™ | XC3S500E-5 | 67 | 1748 | 212 | 1 | 5 | 0 | 0 | ISE 10.1 |
| Virtex-4™ | XC4VLX15-12 | 138 | 1999 | 212 | 1 | 5 | 0 | 0 | ISE 10.1 |
| Virtex-5™ | XC5VLX50-3 | 181 | 995 | 212 | 1 | 2[3] | 0 | 0 | ISE 10.1 |

Notes:

1) Actual slice count dependent on percentage of unrelated logic – see Mapping Report File for details

2) IOB count when all core I/Os and clocks are routed off-chip, which is **not** the intended usage. The core interface is designed to provide flexibility inside a larger FPGA design. GClk signal is normally shared with user's design.

3) Virtex 5 RAM blocks contain two 18K RAMs and are approximately twice the capacity of those in the other FPGA families.

The area requirements for AES Keywrap in the table above are significantly greater than those quoted for AES G2 in the AES G2 data sheet and the clock frequency is lower. Some of this area increase is due to the keywrap circuits but it also reflects a different configuration of the AES G2 core being chosen for the illustration with keywrap. The illustration in the AES G2 datasheet is for ECB mode, encrypt only, 128 bit key whereas the configuration here is ECB and CBC mode, encrypt and decrypt with a 256 bit key. It is normal to use a 256 bit key for the key wrapping even when a 128 bit key is used for the traffic. There are many implementation options available for the Keywrap and AES-G2 core which affect area and performance – Algotronix will provide figures for the exact configuration required on request.

## Functional Description

### Related Standards

The AES Keywrap algorithm was developed by NIST with the intention that it would become the US government standard for protecting encryption keys. AES Keywrap was later adopted by IETF as RFC 3394 in September 2002 – RFC3394 is essentially identical to the NIST AES Keywrap specification except for cosmetic formatting. Although AES Keywrap was never adopted as a formal FIPS standard or NIST special Publication it is specified as one of two approved method for encrypting symmetric keys in the influential FIPS 140-2 "Security Requirements for Cryptographic Modules" standard. AES Keywrap with a 256 bit key is also specified as a required algorithm in the W3C Recommendation "XML Encryption Syntax and Processing" and in the IEEE P1619 Draft Standard for Encrypted Shared Storage Media. AES Keywrap has emerged as the de-facto standard for cryptographic protection of cryptographic keys using symmetric cryptography.

The AES algorithm itself which underlies AES Keywrap is standardized by NIST as FIPS 197. The relevant standard to this implementation is FIPS 197 which specifies the AES algorithm. The NIST also provides a compliance testing standard: "*The Advanced Encryption Standard Algorithm Validation Suite* (AESAVS)", November 15, 2002. This document specifies compliance testing for AES. A third NIST document SP-800A "Recommendation for Block Cipher Modes or Operation" specifies the standard modes of operation for AES. The Algotronix G2 core implements these standards in their entirety, however compilation options are provided so that only circuitry required for a particular application need be included.

The NIST documents mentioned here can be downloaded free of charge from the NIST website (www.csrc.nist.gov) and are also supplied with the core

### Architecture

The top level architecture for the AES Keywrap core is shown in Figure 1 The KeyWrap unit is placed before the AES encryptor and takes control of its input and output signals in order to extend its functionality.   The connectivity around the KeyWrap unit allows it to instruct the Encryption core to decrypt data which is to be used as Key information.

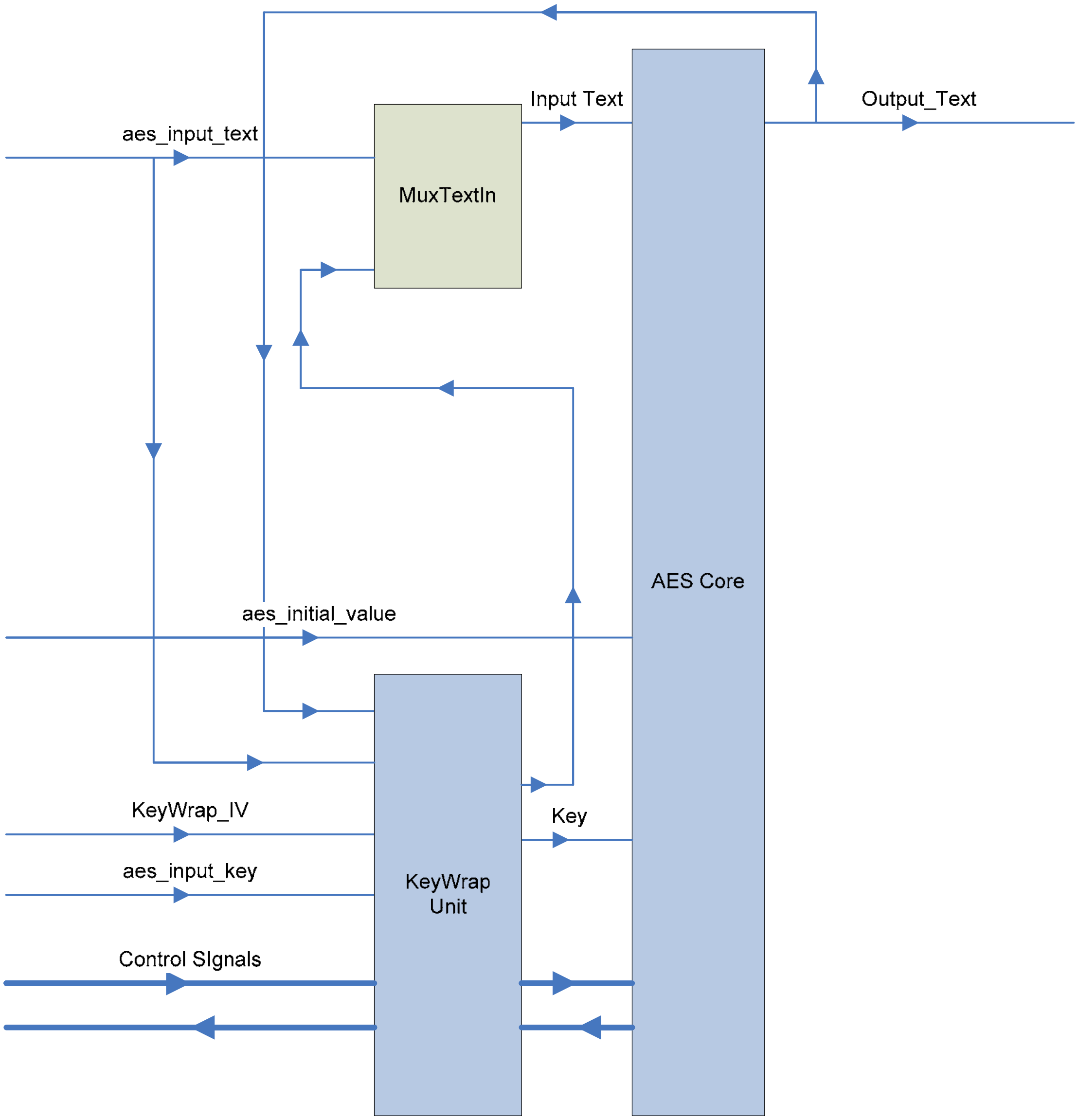

**Figure 1 - AES Keywrap Top Level Architecture**

## Compilation Options

The core can be configured easily using a set of VHDL generic parameters. Normally, it is unnecessary for users to modify the design source code although the code is supplied and they are free to do so if they wish. Algotronix can also customise the core as a service for users with particular requirements which are not met by the standard product.

The following KeyWrap related compilation options are specified by VHDL generic parameters:

- **Wrap_Function –** specifies whether the unit should implement Wrap only, UnWrap only or both Wrap and UnWrap functions.
- **Max_Key_Data_Length -** specifies the maximum length of data to be wrapped or unwrapped.

The remaining generic parameters are passed to the AES G2 core within the KeyWrap unit – these parameters are explained in more detail in the AES G2 Core Product Description.

- **Cipher_function** - specifies whether an Encryptor, Decryptor or Encryptor/Decryptor is required.
- **Max_Crypt_Size** – specifies the maximum key length the core should implement. The user can select any key length up to and including this using control signals. For example, if Max_Crypt_Size is aes256 then the core would deal with 256, 192 and 128 bit keys.
- **Implement_SBoxes_in_RAM** – specifies that FPGA RAM blocks rather than logic gates should be used to implement SBoxes and Inverse SBoxes. This is the most efficient option if RAM blocks are available after mapping the remainder of the user design.
- **Omit_CBC_Mode, Omit_OFB_Mode, Omit_CTR_Mode, Omit CFB1_Mode, Omit_CFB8_Mode, Omit_CFB128_Mode -** Used to request that logic to support cipher modes that will not be required is omitted from the design. The CTR and CFB modes require quite large amounts of additional logic. There is no option to omit the ECB mode since this mode is required by the KeyWrap algorithm.
- **Keyschedule_Shares_Sboxes** – specifies that the same SBoxes are used for the encryption datapath and the keyschedule unit. When this option is selected the encryption key schedule must be pre-calculated in the same way as decryption keys causing additional latency when the key is changed.
- **Force_output_low_until_valid** – When true the core will hold the output low at all times when valid output data is not present. When this signal is false the circuitry to hold the output zero will be omitted, saving some area. In this case the core output ‘output_text’ will show the values at intermediate rounds of the cipher as well as the final round. This data is not fully encrypted and, if available to an attacker, could compromise security of both the key and data. Therefore, this parameter should only be set to false if the user’s design which contains the core can guarantee that an attacker will not be able to monitor the core output directly.
- **Target_Device** –or ActelAxcelerator FPGA families as the target device.

## Core I/O Signals

The core signal I/O have not been fixed to specific FPGA device pins to provide flexibility for interfacing with user logic. Descriptions of all signal I/O are provided in Table 2.

| Signal | Signal Direction | Description |
|---|---|---|
| clock | input | Clock – active on rising edge |
| clear | input | Synchronous clear of controller state and most registers (Some shift registers do not use clear to allow a more area efficient implementation). |
| reset | input | Asynchronous reset – active high.  Usually connected to FPGA global reset. |
| enable | input | Module clock enable – 0: module is inactive, 1: module runs |
| mode | input | Mode signal – specifies which mode of AES is to be implemented – see the NIST SP800-38A publication for a description of available modes.  The KeyWrap core adds two additional modes not provided in the AES G2 core WRAP_MODE and UNWRAP_MODE.  See also the omit_* compilation options in the section below.  If compilation options have specified that logic for a particular mode should be omitted then incorrect behaviour will result if that mode is selected. |
| KeyDataLength | input | Additional signal for KeyWrap core - specifies the length of the **key data** to be wrapped or unwrapped – this is not necessarily the same as the length of the Key which is used in the Wrap or Unwrap operation.  The length of the Key is selected by the KeyLength signal. |
| KeyWrapIV | input | Specifies the 64 bit IV used in the validation check on unwrapping key data.  The standard value for this IV is “A6A6A6A6A6A6A6A6” and in many cases this signal will simply be hard wired by the user. |
| Correct_unwrap | output | Goes to logic ‘1’ if the key unwrap operation has a correct checksum.  If this signal is ‘0’ then the unwrapped key should not be saved. |
| KeyLength | input | Specifies the length of the key that is being used – 128, 192 or 256 bits.  See also the max_crypt_size compilation option.  Only keys up to the size specified in max_crypt_size may be specified e.g. if max_crypt_size generates hardware for a 192 bit key then KeyLength may be 128 or 192 bits but not 256 bits. |
| Do_Encrypt | input | Specifies whether the core should operate in Encrypt or Decrypt mode.  This input is only significant if the compilation option cipher_function is set to EncryptDecrypt i.e. hardware for both encryption and decryption has been included. |
| Start | input | Starts a new encryption operation or block of operations in the chained modes.  The control signals Mode, KeyLength and Do_Encrypt are sampled and the parameters fixed for the next operation. The key is assumed to have changed and the keyschedule is recalculated (or loaded if the compilation option User_Calculates_Keyschedule is active). |
| load_text | output | Load flag – high when input_text is being loaded |

| | | |
|---|---|---|
| Load_key | output | Load flag – high when the key is being loaded |
| Output_Valid | output | Valid flag – high when output_text is valid. |
| Advanced_Output_ Valid | output | High on the four clock cycles immediately preceding output_valid. This signal gives advanced warning that the core is about to input and output data and can by external control circuitry to stop the core using the enable signal until the system is ready to provide new input data and accept output data. |
| Input_Text[31:0] | input | Data input: current 32-bit word of the 128-bit plain text |
| Output_Text[31:0] | output | Data output: current 32-bit word of the 128-bit cipher text |
| initial_value [31:0] | input | current 32-bit word of the 128-bit initial value for the chained modes of operation (ECB mode does not use the initial value). |
| Input_Key[31:0] | input | current 32-bit word of the key – it takes 4, 6 or 8 clock cycles to load a complete key. If the compilation option User_Calculates_Keyschedule is specified the entire keyschedule is input through this signal. |

**Table 2: Core I/O Signals.**

## Description of Operation

The AES Keywrap circuit extends the AES G2 core which has a 32 bit datapath, therefore the main input and output busses to the Keywrap core are also 32 bits wide. One of the complexities in implementing KeyWrap comes from the fact that it is specified in terms of a 64 bit data path where AES manipulates 128 bit blocks using a 32 bit datapath.

The Keywrap algorithm requires 6 basic AES encryption or decryption operations to process each 64 bit word of key data. This is 12 times the computational load of standard AES which processes 128 bits in each operation. AES Keywrap is usually used with a relatively long Key Encryption Key – 192 or 256 bits so the number of rounds in the AES algorithm is also quite large. Therefore it can easily take several hundred clock cycles to wrap or unwrap a key.

Timing on the AES Keywrap unit is very similar to that on the AES G2 core. When Keywrap is used to implement standard encryption and decryption operations the timing is the same as for AES G2 and as described in the AES G2 Product Description. When KeyWrap is used to wrap or unwrap keys the only difference is that many more clock cycles are required so there is a much longer delay between starting the operation and obtaining a result.

The input_text, output_text and initial_value busses transfer 128 bit AES block values over 32 bit wide busses (these 32 bit busses are declared as datatype word in the above code fragment) using four successive clock cycles. The input_key bus transfers 128, 192 or 256 bits of key information over a 32 bit bus using 4, 6 or 8 clock cycles respectively. Words are transferred in order starting with the most significant word.

The best way to get an understanding of the timing of the interface signals to the core is to simulate the core using the testbench provided and examine the waveforms on the signals in the table above during operation in the mode of the cipher you wish to use.

## Verification Methods

The AES G2 core has its own extensive testbench which is described in the G2 Core product description. The KeyWrap product includes both the G2 core and the G2 core testbench. Therefore, the function of the KeyWrap core testbench is only to test the additional functionality provided by KeyWrap - from the point of view of the KeyWrap testbench the G2 core itself can be treated as a 'trusted' component.

The AES Keywrap specification provides suitable test vectors and the AES Keywrap testbench applies all these standard tests.

## Customization Service

Algotronix can offer a cost effective customization service for this core in order to tune the implementation for easy integration into a larger system. It is also possible to produce variants with significantly higher performance at the expense of increased area and to create optimized variants of the core targeted at particular FPGA products.

## Recommended Design Experience

It is recommended that the user is familiar with the VHDL language and with the Xilinx design flow and simulation tools. The core can also be instantiated inside a wrapper to allow use with a Verilog design flow.

It is also recommended that the user has a background in data security or takes appropriate advice when considering how to implement AES-G2 in a larger system.

## Ordering Information

This product is available directly from Algotronix under the terms of the SignOnce IP License. Please contact Algotronix for pricing and additional information about this product using the contact information on the front page of this datasheet. To learn more about the SignOnce IP License program, contact Algotronix or visit the web:

Email: commonlicense@xilinx.com
URL: www.xilinx.com/ipcenter/signonce

## Export Control

Strong encryption technology such as AES is the subject of international export regulations. Algotronix is located in the United Kingdom and export of this core is regulated by the UK government.

The core is freely available within the European Union and in addition can be supplied immediately to the following countries: United States, Australia, New Zealand, Canada, Norway, Switzerland, Japan.

Export to other countries requires an export licence. The UK Department of Business, Enterprise and Regulatory Reform publishes information on their website (www.berr.gov.uk) which gives an indication of average export licence processing times for various countries and the percentage of licence requests which are granted. For many countries obtaining an export licence can be done relatively quickly and with only a small amount of paperwork.

It is the the responsibility of the customer to comply with all applicable requirements with respect to re-export of products containing the AES technology.

## Related Information

### Industry Information

The AES standard documents FIPS197, SP800-38A and AESAVS are available from the National Institute of Standards and Technology, Computer Security Resouce Center website (www.csrc.nist.gov).

### Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com

The supply of the product described in this document is the subject of a separate license agreement with Algotronix Ltd. which defines the legal terms and conditions under which the product is supplied. This product description does not constitute an offer for sale, a warranty of any aspects of the product described or a license under the intellectual property rights of Algotronix or others. Algotronix products are continuously being improved and are subject to change without notice. Algotronix products are supplied 'as is' without further warranties, including warranties as to merchantability or suitability for a given purpose. Algotronix' products are not intended for use in safety critical applications.

URL: www.algotronix.com

| **Version Control Information** | |
|---|---|
| Subversion Revision Number | **27** |
| Date | 2009/09/10 14:03:37 |
| Document | Aes Keywrap Data Sheet, Xilinx Edition |
| Status (blank field indicates OK/no warnings) | |
| | (Table auto-updates, do not edit field values by hand) |