

Joseph Ceirante
CSAW: Embedded System Challenge
October 10, 2008

Introduction:

The Alpha encryption device presents an odd challenge in attacking. Mainly it is not very secure to begin with and it is quite simple. It takes keyboard input, encrypts with AES, and outputs it via serial. The AES encryption is keyed by a hard-coded value which is altered a bit based on the inputs from the device's switches. From this I have deduced two significant assets to attack. The first is the master key. The value of the switches is sent with the ciphertext so it is known anyway. The second asset is the basic functionality of the device. If it doesn't work, then it is worthless. So I have developed two simple Trojans to attack those two assets and then two pranks to cause minor irritation to the users.

Attack 1: Magic Key to Output Master Key

File: aes128_fast_hack.vhd

Effect: The master key (augmented by the switches on the FPGA board) is outputted as though it were ciphertext.

Validates: all inputs except whatever the magic input is set to

Affects on hardware profile: Adds a small amount of logic to handle the extra output case.

Description: This attack is accomplished by simply inserting two lines in aes128_fast in the output process. An extra elsif to check for the magic input and then a line to send the key to data_out is all that is needed to allow for this Trojan. The magic input can be set to anything 128 bits or less though it is probably easiest if the key is exactly 128 bits.

Attack 2: Magic Key to Lock Up Board

File: aes128_fast_hack.vhd

Effect: The aes module goes into a state that prevents it from "finishing" and thus continually does encryption rounds. This state cannot be exited except by powering off the FPGA.

Validates: all inputs except whatever the magic key is set to

Affects on hardware profile: Adds a one-bit register and a small amount of logic to set and check the register.

Description: The signal start_d2 is uncommented and used as the register that when set to one, prevents AES from ever finishing. The output assignment process is used to set and update the register if the magic key (initially set to "xxxxxxxxxxxxxxxx") is inputted. Once the register is set the round counter is then unable to increment and AES stays in that mode forever.

Prank 1: Messing Up the Keyboard

File: kb2ascii_hack.v

Effect: This is a prank that merely changes what keys on the keyboard correspond to what letters in ASCII. This can be done in numerous ways from changing every key to only swapping a few keys.

Validates: all inputs with no x, y, n, or m or anytime the validating party does not look closely enough.

Affects on hardware profile: None

Description: The file kb2ascii.v simply contains a mapping for keyboard inputs to ascii values. This prank merely changes the mapping to be something unexpected to the user. This is not a serious attack on the system.

Prank 2: Messing up the switches and LEDs

File: MainBasys_hack.ucf

Effect: This prank merely changes what inputs correspond to what switches on the FPGA board. It also changes what LEDs light up when.

Validates: All input unless the transmitted key is separated from the message

Affects on hardware profile: None

Description: The ucf file specifies what specific parts of the FPGA board map to what inputs in the hardware description. These can be changes however desired. This particular prank simply changes the light to light up randomly while the user is typing and permutes the switches. However, due to the nature of the overall program, changing the switches may have no effect since that part of the key is transmitted with the message.