



Linx Technologies CipherLinx(TM) Protocol and HS Series *Evaluation and Summary Report*

Independent Security Evaluators
www.securityevaluators.com

August 28, 2006

© Independent Security Evaluators 2006. All rights reserved



Executive Summary

This report details the findings of Independent Security Evaluators (ISE) in examining the design of the Linx Technologies HS Series encoders and decoders. These are the first devices to utilize CipherLinx(TM), a security protocol that includes Skipjack encryption and strong authentication. The HS Series provides a high degree of security against the assumed threat model (described in detail in this document). In particular, attackers in the threat model are unable to forge messages that will be accepted by a decoder, cause the encoder and decoder to become unsynchronized, or use a legitimate encoder with a constrained set of privileges to gain more privileges. In all cases, the attacker is assumed to know exactly how the HS Series works. Only the keys are secret; there is no “security through obscurity.”



Contents

- Executive Summary 2
- Contents 3

- 1 Overview and Assumptions 4**
- 1.1 Threat Model 4
- 1.2 Scope of Evaluation 6

- 2 Analysis 7**
- 2.1 Setup Phase 7
- 2.1.1 Key Generation 7
- 2.1.2 Key Transmission 8
- 2.1.3 Encoder Capabilities 8
- 2.2 Main Operation 8
- 2.2.1 Encryption 9
- 2.2.2 Confidentiality 10
- 2.2.3 Authenticity 10
- 2.2.4 Remaining Goals 10

- 3 Conclusions 12**

Chapter 1

Overview and Assumptions

The LinX HS Series consists of two modules: an encoder and a decoder. The basic mode of operation is as follows. While any of eight parallel lines on an encoder are activated, a data signal is generated containing an identifier for that encoder and an encrypted, authenticated block that specifies which of the eight lines are active. Upon reception of this signal, a decoder will attempt to decrypt the block using the appropriate key (indexed by the encoder identifier). If the block decrypts and authenticates correctly, the eight output lines on the decoder will be set to the state specified in the block.

Before they can be used together, the encoder and decoder must be *paired*. First, a key is generated on the decoder by extracting randomness from a series of button presses (or other random source). This key is then transmitted to the encoder via a wired or wireless interface. During this setup process, the user can also limit the capabilities of the encoder by specifying which output lines on the decoder it is able to control. Once the setup process is completed, all further communication is unidirectional from the encoder to the decoder.

1.1 Threat Model

A threat model defines the assumed environment and capabilities of the adversary against which a security product is evaluated. As no product can ever be “perfectly secure,” the model captures the attacks that the product is designed to mitigate. The threat model for the LinX HS Series is as follows:

- The attacker knows exactly how the HS Series works. There is no “security through obscurity.” That is, the security of the system does not rely on the design of the system being

secret. Instead, the security is derived only from the secret cryptographic keys generated by the device.

- The attacker is able to eavesdrop on any communications between the encoder and the decoder subsequent to the initial setup phase.
- The attacker is able to send arbitrary packets to the decoder. He may construct these packets based on information derived by eavesdropping on communications subsequent to the setup phase.
- The attacker is able to prevent legitimate transmissions from being received by the decoder (perhaps by jamming the signal) after the initial setup phase.
- The attacker does *not* have physical access to the decoder he wishes to attack, though he may have access to other decoders.

In the face of such an attacker, the system should meet the following goals:

1. If a correctly generated packet sent by an authorized encoder (an “authorized packet”) is received correctly by the decoder, the appropriate output lines must be activated.
2. If an authorized packet that was correctly received is ever retransmitted (“replayed”) by an adversary, the decoder will detect this condition and reject the packet.
3. If the decoder receives packets “out of order” it will detect this condition. Specifically, if the decoder receives an authorized packet P generated at time t , and *subsequently* receives an authorized packet P' generated at a time prior to t , the decoder will reject packet P' .
4. Any packet constructed by an adversary will be rejected by the decoder, provided that the adversary has not had access to a legitimate (paired) encoder.
5. Given physical access to any set of legitimate encoders, an adversary will be unable to generate an authorized packet for another encoder not in the set.
6. Given physical access to a legitimate encoder, an adversary will be unable to activate an output line on the decoder if this function was disallowed during the initial setup process.
7. Without observing the actions taken by a decoder, an attacker should learn nothing about a command sent from an encoder to a decoder (other than which encoder sent the command).

Goal #3 above is necessary due to the unidirectional nature of the link. Absent a synchronized clock or challenge-response protocol, the decoder has no way to determine whether a received packet is “fresh”, unless it has already received a more recent authorized packet.

1.2 Scope of Evaluation

ISE's analysis of the design of the HS Series was based on both the design documentation and interviews with members of the Linx team. While ISE had access to both the source code and physical devices, the scope of the evaluation did not include a comprehensive code analysis. However, portions of the code, including the Skipjack core, were compared for correctness to relevant cryptographic design documents. The analysis also did not consider side channel attacks specific to the HS series implementation of the CipherLinx(TM) protocol.

Side channel attacks are those that arise because of the specific physical instantiation of a security design. These would include unintentional RF emanations inherent in electronic device operation, glitching caused by powerful electromagnetic interference, or small time-scale deviations in processing time. Side channels are generally more difficult and time consuming for an attacker to exploit and such attacks often require the use of expensive, specialized equipment.

Chapter 2

Analysis

In this section, we examine the Linx HS Series against the threat model described in section 1.1.

2.1 Setup Phase

During the setup phase, keys are generated on the decoder and sent to the encoders. The capabilities of each encoder are also set.

2.1.1 Key Generation

When generating cryptographic keys, it must be equally likely to choose any of the possible keys. In the case of the Linx HS, each of the 80 bits of the key must be chosen uniformly (equally likely to be a 0 or a 1) and independently (the value of any bit does not depend on what was chosen for any of the other bits).

Keys are generated on the decoder through a series of activations on an input line. A high resolution timer in the decoder is triggered each time that the line is activated or deactivated. The low order bits of this timer are used to create the key.

The Linx documentation suggests connecting a button to this line, in which case activations and deactivations correspond to button presses and releases. Based on experiments done by Linx, this setup produces bits that are uniform and independent. Designers are cautioned against using some deterministic source (such as an implementation of a non-cryptographic random number generator like a linear feedback shift register) to trigger this input line during key generation.

2.1.2 Key Transmission

The decoder supports several methods of conveying the key to the encoder. While this process is considered to be outside the threat model (it is assumed that the adversary cannot eavesdrop during setup), it is always best to use a wired interface to transmit the key to the encoder, rather than an IR or RF interface.

At setup time, the devices each set an initial counter value C equal to 0xFFFFFFFF. The encoder stores the key (K) and the initial counter value C . The decoder stores an association between the ID of the paired encoder, and the values (K, C).

2.1.3 Encoder Capabilities

The HS Series allows specific encoders to be constrained to only activate certain output lines. These restrictions are specified during setup, and a mask of the lines that each encoder is allowed to control is stored in the decoder along with the encoder's ID and key. The threat model assumes that the adversary cannot subvert the setup process or physically access the decoder to change these settings. Provided that the decoder can correctly identify which encoder transmits a command during normal operation, the capability mask will constrain the outputs as required. In the sections below, we discuss how the decoder meets this requirement.

2.2 Main Operation

Once the setup phase is complete, the system operates unidirectionally: packets are sent by the encoder to the decoder. A packet is sent when one or more of the input lines on the encoder is activated. The eight input lines are each mapped to one bit of a command byte, X .

The HS Series uses the CipherLinx(TM) protocol to create packets. Each packet contains:

- The ID of the encoder that transmitted the command
- An encrypted block, which contains:
 - The command byte X
 - The 40-bit counter value C .
 - An 80-bit authentication pattern A , which is a static pattern shared by all HS encoders and decoders.

The value of the encrypted block is computed using a single execution of a cipher $E_K(\cdot)$ where K is the key shared between the encoder and decoder. The encoder decrements the counter C for every packet sent, and a new key must be installed in the device if the counter ever reaches 0. This is unlikely to ever occur given the size of the counter field and the physical rate limiting of the device.

When the decoder receives a packet, it first checks the encoder ID to determine whether it possesses an associated key K and counter value which we refer to as C_{last} . If this information is found, the decoder decipheres the encrypted block using $E_K^{-1}(\cdot)$ to recover the command byte X , the counter C , and the authentication pattern A . When considering a new packet, it first checks to see if the counter value C in the received packet is less than the stored value C_{last} . If not, the decoder rejects the packet. Every time that the decoder decides that a packet is valid (see below), it updates the stored counter value C_{last} with the received value C . (Goals #2 and #3)

The decoder next checks to make sure that the authentication pattern A matches the static pattern used by all HS encoders and decoders. If, so it declares the packet valid and sets the output lines to the values specified in the command byte, X .

2.2.1 Encryption

The enciphering function, $E_K(\cdot)$, used in the HS Series is based on the U.S. National Security Agency designed cipher “Skipjack.” Skipjack is a block cipher with 80-bit keys and 64-bit blocks. There are currently no known cryptanalytic shortcut attacks on Skipjack and exhaustively searching an 80-bit keyspace is out of the realm of practicality for the foreseeable future. It is important to note that key length only provides an upper bound on the work required of an adversary. If a cryptanalytic attack on a cipher is discovered, the effective key length of a cipher with 128 bit keys might be reduced to only a few bits. Thus far, there have been no such cryptanalytic attacks on the full Skipjack.

Because each packet is longer than 64 bits, Skipjack must be employed in an encryption mode. The particular encryption mode chosen is based on the *CMC* encryption mode, so that the resulting cipher is a special kind of function known as a “strong PRP” (sPRP). The definition of a strong PRP is rather technical, but it essentially says that an adversary is unable to distinguish the given permutation from a random permutation on the same domain when given suitable access to the function and its inverse. This property will be used to show that the encryption is secure.

The encryption mode uses several invocations of Skipjack to encrypt the 128 bits in each data packet.

2.2.2 Confidentiality

The system is designed such that an adversary who eavesdrops on a packet transmitted from an encoder to a decoder does not learn what command byte, X , the encoder sent (though he does learn which encoder did the sending, as the ID is sent in the clear). Because an sPRP is used to encrypt the command byte X along with a counter, a result of Bellare and Rogaway tells us that so long as the counter never repeats, the entire encryption is “semantically secure.”¹ The “semantic security” property means that an adversary learns nothing from seeing the encryption that he did not already know before seeing the encryption, other than possibly the length of the data that was encrypted. Since the data is always a fixed size, the attacker learns nothing at all, and the command byte, X , remains confidential. (Goal #7)

2.2.3 Authenticity

Besides keeping the command byte private, the system should prevent an adversary from forging an unauthorized packet that will be accepted by a decoder. This is enforced by the use of the authentication pattern, A . The same result of Bellare and Rogaway tells us that when a fixed pattern is included in an encryption using a sPRP, an adversary’s probability of successfully forging a packet is negligible so long as the fixed pattern is long. The pattern used by the Linx HS Series is 80 bits in length, so the probability that an illegitimate packet is accepted as valid is approximately 2^{-80} . (Goal #4)

2.2.4 Remaining Goals

Goal #1 is satisfied because every legitimate packet has the correct authentication pattern, a counter lower than the counter in the last legitimate packet, and an encryption under the correct key. These are the only conditions that are required in order for the decoder to accept the packet.

Goal #5 is satisfied because gaining access to a particular encoder only gives the adversary access to the encoder’s key (as the entire design of the encoder is assumed to be known the adversary). Because keys are generated independently, learning the key to one encoder does not provide any additional capabilities with respect to the key of another encoder.

Goal #6 is satisfied because if the attacker does not include the correct ID in the packet, the decoder will not use the correct key to decrypt the packet. If the wrong key is used, the block will not

¹Even though it would take several hundred years for the counter to repeat at the highest baud rate, the Linx HS automatically disables itself until a new key is selected when the counter reaches 0, so that a counter is never reused. In contrast, systems using a 32-bit counter might repeat a value several times a day under continuous use.



decrypt properly (the authentication pattern will not match) and the packet will be rejected. Therefore, the correct ID must be used, which implies that the decoder will use the correct capability mask.

Thus, the Linx HS Series meets all of its security goals.

Chapter 3

Conclusions

The Linx HS Series encoders and decoders provide a high degree of security when evaluated against an appropriate threat model. An adversary is unable to read or modify the commands sent by legitimate encoders. The encoders and decoders never get out of sync due to a 40-bit counter included in each packet. Access to individual output bits of the decoder can be controlled securely.

Additionally, the Linx HS Series avoids the common security problems that routinely cause other “secure devices” to fail:

- Key generation allows for all the possible keys to be chosen with equal probability so that adversaries can not narrow their search to a particular subset.
- The encryption used in the system is stateful so that repeated messages do not produce the same ciphertext.
- The encryption used in the system is authenticated so that adversaries can not modify legitimate messages.
- The counters used in the system are large enough that they should never roll over, though the device prevents roll-over by automatically requiring a new key should a roll-over ever occur. This prevents an adversary from replaying a previous legitimate message.

In short, the CipherLinx(TM) protocol in the HS Series is well-designed and is an excellent choice for applications requiring a secure unidirectional link.