SEL-3021 SERIAL ENCRYPTING TRANSCEIVER IS IMMUNE TO THE SHA-1 HASHING ALGORITHM COMPROMISE

WHITE PAPER

David Whitehead, P.E.
Schweitzer Engineering Laboratories, Inc.
Pullman, WA USA

SUMMARY

On February 14, 2005, computer magazines and websites reported that a research team in China compromised Secure Hash Algorithm 1 (SHA-1 [FIPS-180-1]). Several people asked SEL what this supposed breach of the algorithm means to users of the SEL-3021 Serial Encrypting Transceiver. The short answer is that the reported SHA-1 compromise does not impact the SEL-3021.

The encrypted serial-port link between SEL-3021 transceivers DOES NOT employ the SHA-1 algorithm, but instead uses Advanced Encryption Standard (AES) encryption with 128-bit keys.

The SEL-3021 wireless operator interface, which is used for settings and diagnostics, implements a combination of SHA-1 and 128-bit secret key called HMAC SHA-1. The claimed weakness in SHA-1 is associated only with the basic algorithm and does not affect the implementation of HMAC SHA-1 in the SEL-3021. The SEL-3021 protects data sent across the wireless operator interface by generating and appending a user-defined keyed HMAC SHA-1 checksum to the original message. The SEL-3021 then applies 128-bit AES encryption to the combination of the original message and HMAC SHA-1 checksum. This double protection (HMAC SHA-1 and 128-bit AES) means that all data are extremely well protected, with much more than 128 bits of data security.

WHAT IS SHA-1?

The National Institute of Standards and Technology (NIST) developed the SHA-1 one-way hash algorithm in 1993. A sender runs the message through the SHA-1 algorithm, which appends the resulting unique, fixed-length identifier (checksum) to the message as a digital signature or fingerprint. The message/checksum recipient passes the message portion again through the SHA-1 algorithm and compares the resulting checksum against the checksum appended to the original message. These checksums must match to ensure that there was no unintentional change or malicious tampering to the message during transmission.

A cryptographically strong hash function has the following properties:

- Given a hash function, $H(m)$, and its output, h, it is extremely difficult to derive a message, m, such that $H(m) = h$.
- Given a message, m, it is extremely difficult to find another message, m', such that $m \neq m'$ and $H(m) = H(m')$.

The second point can be stated another way:

- Given a message pair m and m', such that m ≠ m', it is extremely unlikely that H(m) = H(m')

The second and third points above are slightly different. The second point states that if you know message m, it is very difficult to find another message m' that results in the same hash. However, finding a message pair m and m' that result in the same hash is much easier [1].

WHAT ARE THE POTENTIAL ISSUES WITH SHA-1?

Because a checksum generally has fewer bits than the message from which it was computed, multiple messages can have the same checksum output. Consider a cryptographic checksum function that computes hashes of three bits for a set of files that contain five bits. The total for possible hashes is $2^3 = 8$ and for $2^5 = 32$ files. This results in at least four different files that have the same hash output [1].

A hash collision occurs when two different messages, m and m', produce the same hash output. This situation can be dangerous because it leads to situations where an attacker replaces the original message contents, m, with another message, m', and the substitution is not detectable by comparing the transmitted checksum against the checksum computed from the received message. For a perfect hash function with an output size of N bits, an attacker can expect to find a hash collision in an average of $2^{N/2}$ operations.

To carry out a successful attack, a malicious individual precomputes a large number of m and m' collision pairs, waits for one of the messages (m or m') to be transmitted by the authenticated user, and substitutes the transmitted message with the other message from the appropriate collision pair. The success of the attack depends on the attacker’s ability to identify collision pairs that consist of a message, m, that is likely to be transmitted (i.e., is meaningful in the target system) and a corresponding message, m', that is both meaningful in the target system and produces a desired, malicious outcome. These conditions are clearly not satisfied by the vast majority of potential m and m' collision pairs. In fact, for most systems, the majority of the collision pairs do not result in a meaningful and useful attack.

SHA-1 produces a hash output of 160 bits. The probability of finding a message that corresponds to a given hash is 2^{160} operations. However, the probability of finding two messages with the same hash is theoretically only 2^{80} operations. Note that Schneier refers to these 2^{80} operations as the theoretical strength of SHA-1.

According to recent press releases, three researchers in China have supposedly compromised the SHA-1 hashing algorithm. The attack apparently demonstrates that collisions can occur in only 2^{69} hash operations, far fewer than the brute-force attack of 2^{80} operations based on the hash length [2], [3]. Note that 2^{69} operations is still a very large number.

> “In 1999, a group of cryptographers built a DES cracker. It was able to perform 2^{56} DES operations in 56 hours. The machine cost $250K to build, although duplicates could be made in the $50K–$75K range. Extrapolating that machine using Moore's Law, a similar machine built today could perform 2^{60} calculations in 56 hours, and 2^{69} calculations in three and a quarter years. Or, a machine that cost $25M–$38M could do 2^{69} calculations in the same 56 hours.”[3]

Finding collision pairs does not necessarily yield useful information. An attacker might find two messages that yield the same hash, but that attacker is likely to find only random gibberish in either message generating a collision. Note that the attack scheme mentioned above is based solely on the SHA-1 algorithm, which does not use a cryptographic key. To enhance the security of SHA-1, use an HMAC SHA-1 implementation for authentication. A hash message authentication code (HMAC [RFC-2104]) is a cryptographic algorithm that uses a keyless hash function (SHA-1, MD-5, etc.) and a cryptographic key to produce a keyed hash function.

How Is SHA-1 Used?

The public uses SHA-1 every day in many cryptographic-related applications. Such an application is Secure Socket Layer (SSL), the security feature many Web browsers use in transferring private keys in Internet transactions involving the secure transmission of credit card number information.

How Is SHA-1 Used in the SEL-3021?

The SEL-3021 transceiver uses HMAC SHA-1 to authenticate messages only on the wireless operator interface. The SEL-3021 transceiver does NOT use HMAC SHA-1 for encrypting serial port data.

The SEL-3021 uses a variable length message and a user-defined authentication key as inputs to the HMAC SHA-1 operation. The HMAC SHA-1 operation produces a 160-bit-long, fixed-length cryptographic hash output value (see Figure 1). The hash output is a unique fingerprint or signature of the message.

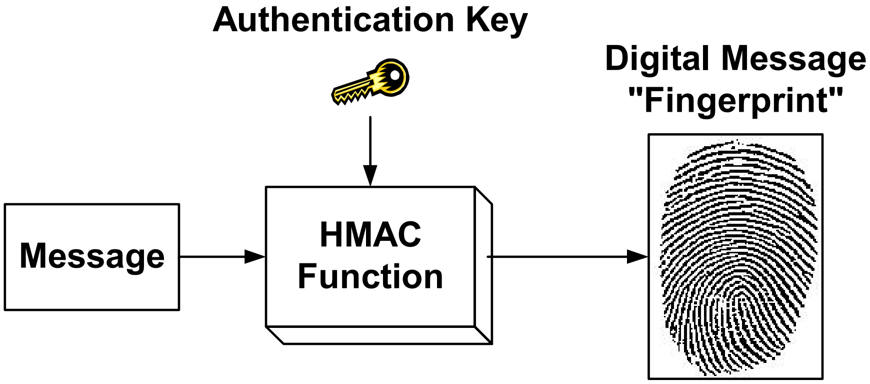

Figure 1 Operation of the HMAC SHA-1 Keyed Hash Authentication Function

The 128-bit secret key gives the HMAC SHA-1 algorithm a strong, built-in authentication capability. If an attacker changes the contents of the message, the hash value appended to the message does not match the value that results if a new hash value were calculated on the new, altered message. Because the HMAC SHA-1 function is keyed (i.e., uses a secret authentication key to form the hash output), an attacker without knowledge of the authentication key is unable to recalculate a new, valid hash value for the altered message, and is unable to hide the fact that the message has been altered.

The SEL-3021 incorporates more than HMAC SHA-1 for message security. The transceiver further protects each transmitted message by applying 128-bit AES encryption to each message/HMAC hash. Figure 2 shows the process. The United States government has adopted AES, also known as Rijndael, and the standard is in use worldwide.

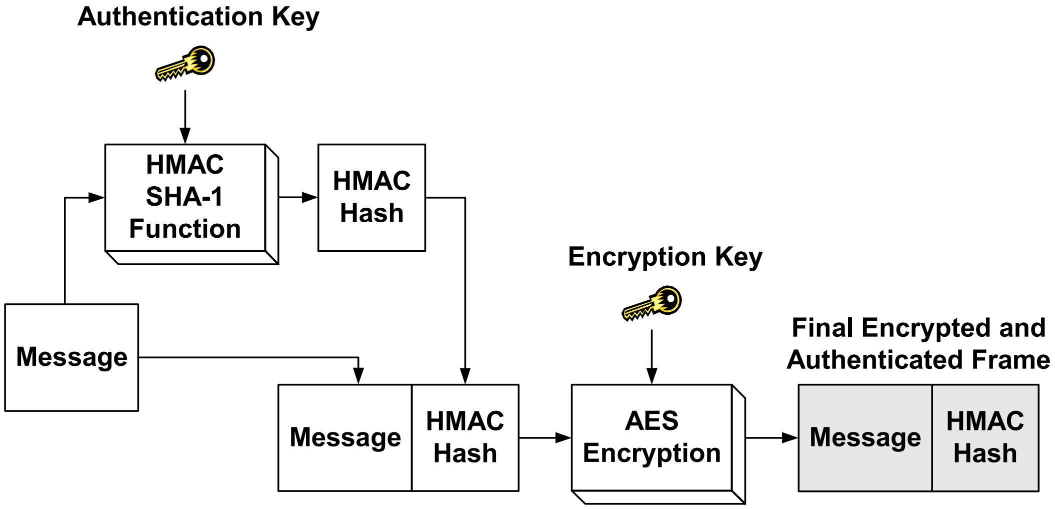

Figure 2 SEL-3021 Encrypted and HMAC SHA-1 Message

An attacker must perform the following steps to defeat the SEL-3021 wireless-operator-interface security features.

- Decrypt the message/HMAC hash
- Find collision-pair messages that match the HMAC SHA-1 secret key hash
- Insert the new message and hope this new message is valid
- Encrypt the message
- Send the message to the SEL-3021

An attacker must perform the previously listed steps quickly (i.e., before the SEL-3021 receives another legitimate message). If the SEL-3021 receives a legitimate message before the attacker sends a modified message, the SEL-3021 rejects the attacker's message.

For a complete description of the SEL-3021 wireless-operator-interface security features, see the SEL whitepaper: "SEL-3021 Wireless Interface Security" at http://www.selinc.com/techpprs/6196_3021WirelessSecurity_AR_DEW_20050214.pdf

What Is the Impact of the SHA-1 Compromise in the SEL-3021?

Some researchers have suggested that the cryptographic security of SHA-1 might be less than the presumed strength. In any case, the SEL-3021 use of the SHA-1 algorithm still provides excellent authentication capabilities for the following reasons:

- The SEL-3021 transceiver uses HMAC in conjunction with SHA-1 to produce a keyed hash function.
- The SEL-3021 transceiver applies 128-bit AES encryption to each HMAC SHA-1 authenticated message.
- Any messages that the SEL-3021 uses have a finite life.

A message in the SEL-3021 is valid only until receipt of the next message or until the session times out. This is very different from static data, such as in financial documents, which are stored for long periods of time and have greater susceptibility to a SHA-1 weakness.

Conclusion

In summary, the SEL-3021 wireless operator interface incorporates double protection (HMAC SHA-1 and 128-bit AES) that results in well over 128 bits of data security in any transmitted message. The SHA-1 security weakness some researchers have reported does not affect the security of the SEL-3021 user interface.

References

[1] Matt Bishop, "Computer Security Art and Science," Addison-Wesley, Boston, MA, 2003.

[2] Bruce Schneier, "Schneier on Security: SHA-1 Broken," http://www.schneier.com/blog/archives/2005/02/sha1_broken.html, February 15, 2005.

[3] Bruce Schneier, "Schneier on Security: Cryptanalysis of SHA-1," http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html, February 18, 2005.

Biography

David Whitehead, P.E. is the Chief Engineer, GSD Division, and a Principle Research Engineer for Schweitzer Engineering Laboratories. Prior to joining SEL, he worked for General Dynamics, Electric Boat Division as a Combat Systems Engineer. He received his BSEE from Washington State University in 1989 and his MSEE from Rensselaer Polytechnic Institute in 1994. He is a registered Professional Engineer in Washington State and Senior Member of the IEEE. Mr. Whitehead holds six patents with several others pending. He designs and manages the design of advanced hardware, embedded firmware, and PC software.

20050407
TP6204-01