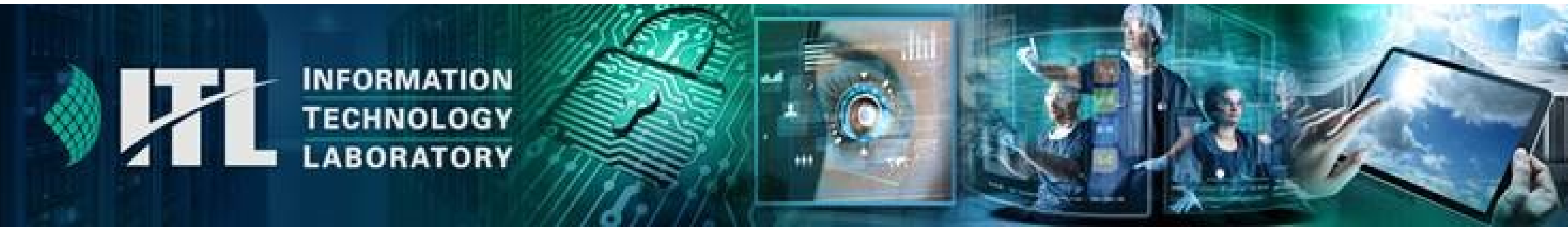


ITL BULLETIN FOR APRIL 2016

NEW NIST SECURITY STANDARD CAN PROTECT CREDIT CARDS, HEALTH INFORMATION

Morris Dworkin, Larry Feldman,[1] and Greg Witte,[1] Editors
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
U.S. Department of Commerce

Background

The cryptography community has long sought a reliable way to encrypt personal and/or confidential data (e.g., credit card number, driver's license ID, social security number) in a manner that preserves the original format. For example, for many years, when you swiped your credit card through a card reader, the entire number would be stored in an unencrypted string of numbers on the device, leaving that data susceptible to criminals' interception. To address this need, NIST has released Special Publication (SP) 800-38G, *Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption*. The guidance in this publication supports sound methods to protect credit card data and many other format-specific types of information.

The engine for many of the techniques in NIST's cryptographic toolkit is a block cipher algorithm, such as the Advanced Encryption Standard (AES) algorithm or the Triple Data Encryption Algorithm (TDEA). A block cipher transforms a set of fixed-length binary data (i.e., a "block") into seemingly random data of the same length. The transformation is determined by the choice of a set of secret data called the "key." The same key is used to reverse the transformation and recover the original block of data. A cryptographic technique that is constructed from a block cipher is called a "mode of operation."

SP 800-38G is the seventh part of a series of recommendation documents regarding the modes of operation of block cipher algorithms. The purpose of this part is to provide approved methods for format-preserving encryption (FPE).

Introduction

A block cipher mode of operation (i.e., "mode") is an algorithm for using a block cipher to cryptographically transform data. Previously approved modes transform binary data, i.e., bit string sequences of ones and zeros. For sequences of non-binary symbols, there has not been a natural and general way to produce encrypted data that looks similar to the original information. For example, financial software, such as that used in credit card readers and billing records, typically expects a 16-digit credit card number. Encrypting a card number through previously approved block cipher modes would result in a longer number, likely causing problems in the software. In contrast,

[1] Larry Feldman and Greg Witte are Guest Researchers from G2, Inc.

the new FPE method produces a result with the same length as the original. The method works on both binary and conventional (decimal) numbers – in fact, sequences may be created from any "alphabet" of symbols.

FPE is particularly helpful for retrofitting encryption technology to legacy applications where a conventional encryption mode might not be feasible. For example, existing database applications may not support changes to the length or format of data fields.

While the main commercial impetus for developing FPE techniques is credit card number encryption, another potential application is the "sanitization" of personally identifiable information (PII) from databases, particularly those containing sensitive medical information. Databases of this sort are invaluable for researching the effects of different treatment methods on diseases, for example, but they often use social security numbers to identify individual patients. The use of format-preserving encryption can help obfuscate this personally identifiable information.

Overview of the Recommendation Development Process

Like many block cipher methods, the FPE described in SP 800-38G specifies modes that are based upon the Feistel design, named for the German-born physicist and cryptographer Horst Feistel. The two format-preserving Feistel-based modes in the publication are abbreviated as FF1 and FF3. These techniques are the result of nearly a decade of collaboration between NIST and industry.

Several other modes have been submitted to NIST for consideration; any interested party may submit new or updated modes for consideration as part of that toolkit, following the process described at the Modes Development page.[2] NIST welcomes public input on whether to approve any of these modes, and comments may be submitted to EncryptionModes@nist.gov.

The subsequent development followed the general approach described below:

1. **Public Comments**: As described above, NIST solicits public input regarding new or improved modes of operation. Guidelines regarding how to submit such proposals are provided on the NIST Computer Security Resource Center (CSRC) website. For applicable submissions, the associated documentation is posted on a "Modes Development" page, along with an open invitation for public review. Any public comments on modes proposals are posted on a "Public Comments" page.

2. **Mode Consideration**: NIST determines, based upon internal analysis and public comments, that a version of a mode proposal is appropriate to include in NIST's cryptographic toolkit. The main considerations are:

   - Whether the mode serves an important need;
   - Whether existing modes in the toolkit, or other modes proposals, can adequately provide the needed properties/functionality instead;
   - Whether the mode meets NIST's security requirements; and
   - For patented modes, whether acceptable royalty-free alternatives are available.

[2] http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html

3. **Recommendation Development**: After deciding to proceed with a recommendation publication, NIST develops a draft Special Publication that specifies the mode. Normally, NIST develops the draft in consultation with the mode submitter. After completion of an internal review, the draft is posted on the CSRC website for a public comment period, after which any received comments are also posted. NIST considers the public comments carefully. Based upon the available input, NIST will:

   - Finalize the draft for publication, with appropriate revisions to address any remaining public or internal concerns;
   - Revise the draft for further public review; or
   - Withdraw the proposal altogether.

This development process ensures continual improvement and transparent collaboration with the security technology community.[3]

Tweaks

The Feistel structure employed by FF1 and FF3 also underlies the TDEA. At the core of FF1 and FF3 are different Feistel-round functions that are derived from an approved block cipher with 128-bit blocks, i.e., the AES algorithm.[4]

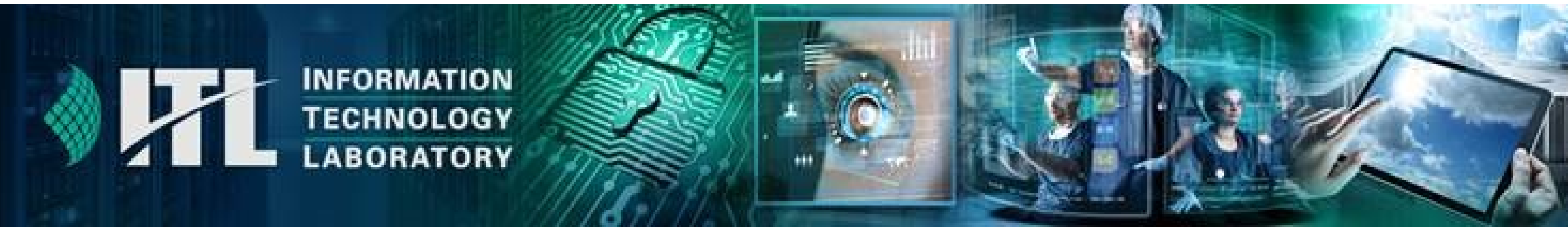
In addition to the formatted data for which the modes provide confidentiality, each mode also takes an additional input called the "tweak," which is not necessarily secret. The tweak can be regarded as a changeable part of the key, because together they determine the encryption and decryption functions. Variable tweaks can be especially important for implementations of FPE modes, because the number of possible values for the confidential data is often relatively small. To illustrate the significance of tweaks for FPE, consider the following example regarding protecting credit card numbers (CCNs):

> *[S]uppose that in an application for CCNs, the leading six digits and the trailing four digits need to be available to the application, so that only the remaining six digits in the middle of the CCNs are encrypted. There are a million different possibilities for these middle-six digits, so in a database of 100 million CCNs, about a hundred distinct CCNs would be expected to share each possible value for these six digits. If the hundred CCNs that shared a given value for the middle-six digits were encrypted with the same tweak, then their ciphertexts would be the same. If, however, the other ten digits had been the tweak for the encryption of the middle-six digits, then the hundred ciphertexts would almost certainly be different.*[5]

In general, if information is available that is statically associated with a plaintext, SP 800-38G recommends using that information as a tweak for the plaintext. Ideally, the non-secret tweak associated with a plaintext is associated only with that plaintext.

---

[3] For details of how NIST develops cryptographic publications, see NIST Internal Report (NISTIR) 7977, *NIST Cryptographic Standards and Guidelines Development Process* (March 2016).
[4] The AES algorithm is specified in Federal Information Processing Standard (FIPS) 197, *Advanced Encryption Standard (AES)* (November 26, 2001).
[5] NIST SP 800-38G, p. 21.



Conclusion

NIST's Computer Security Division is involved in the development, maintenance, and promotion of a number of standards and guidance publications that cover a wide range of cryptographic technology. NIST continues to develop a comprehensive Cryptographic Toolkit that enables users to select cryptographic security components and functionality for protecting data, communications, and operations.

Special Publication 800-38G, *Recommendation for Block Cipher Modes of Operation: Methods for Format-Preserving Encryption*, is the result of many years of industry collaboration to develop reliable processes for encrypting sensitive information while preserving the original format. Preservation of the data format is often an important requirement, such as in support of a software application or legacy database.

NIST will continue to collaborate with industry to address similar cryptographic challenges, particularly in understanding the public need for and availability of reliable block cipher modes.

Additional Resources

FIPS 197, *Advanced Encryption Standard (AES)*

NIST SP 800-67 Revision 1, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*

ITL Bulletin Publisher: Elizabeth B. Lennon
Information Technology Laboratory
National Institute of Standards and Technology
elizabeth.lennon@nist.gov