Report on the
Symmetric Key Block Cipher
Modes of Operation Workshop

October 20, 2000

Sponsored by the National Institute of Standards and Technology (NIST)

A workshop was held to discuss the modes of operation for symmetric key block cipher algorithms on October 20, 2000 at the Baltimore Convention Center in Baltimore Maryland.

1. Welcome and Overview of Intent

Elaine Barker extended a welcome to the workshop attendees and served as the workshop moderator. Elaine stated that the purpose of this workshop was to discuss the modes for protecting data using symmetric key block cipher techniques such as the Advanced Encryption Standard (AES). NIST plans to develop a new modes standard that is written to be independent of specific key or block sizes for specific algorithms, and to include the four DES modes (ECB, CBC, ECB, OFB) that were originally defined in Federal Information Processing Standard (FIPS) 81. Since FIPS 81 was written to be specific to DES and its key and block size, a new standard is needed that will address other symmetric key block cipher algorithms such as AES. Since the world has advanced beyond the world of the 1980s, other modes for protecting data for applications using these technologies are required. The intent of this workshop was to discuss additional modes, the security they afford and their applications. NIST would like to minimize the number of additional modes in order to avoid unnecessary implementation costs and promote interoperability.

2. Presentations

Several papers were provided to NIST prior to the workshop as public comments. These papers and the associated presentations are provided on the NIST modes web page (http://www.nist.gov/modes), along with other comments received.

2.1 Comparing Cryptographic Modes of Operation Using Flow Diagrams

Lyndon Pierson of Sandia National Laboratories presented the workshop attendees with an analysis of the four modes of operation specified in FIPS 81 (ECB, CBC, CFB and OFB) plus counter mode using flow diagrams to depict the operation of the modes. He urged the use of such a method during the analysis of any modes to be considered for the standard. In addition, a table was provided that specifies the security, implementation issues, fault tolerance and synchronization properties of the four FIPS 81 modes.

2.2 Encryption Modes with Almost Free Message Integrity

Charanjit Jutla of IBM presented two new modes, each of which provides both confidentiality and message integrity: Integrity Aware Cipher Block Chaining Mode (IACBC) and Integrity Aware Parallizable Mode (IAPM). He asserted that almost all encryption applications need message security; the new modes provide the additional service at a much smaller cost in

performance than can be achieved when encryption and message integrity are provided separately.

Both IACBC and IAPM modes have proofs of security for both confidentiality and message integrity, assuming that the underlying block cipher algorithm is secure. The proofs of integrity are equivalent to those available for the CBC mode, and the proofs of message integrity are equivalent to those available for CBC-MAC, which is a message authentication code (MAC) based on the CBC mode. A paper containing these proofs is available at the ePrint archive at http://www.iacr.org.

IACBC is a non-parallelizable mode that is similar to the CBC mode, except that IACBC also specifies whitening of the output blocks with a pairwise independent random sequence. Two methods for generating this random sequence are provided in the paper. An implementation of IACBC using a DES engine had a throughput of over 90% of the throughput of a standard CBC implementation. Thus, the cost of message integrity is relatively small compared to the cost of supplementing the CBC mode with a separate MAC.

IAPM is a parallelizable mode that specifies both input and output whitening with a pairwise independent sequence. Thus, IAPM is similar to the ECB mode in its form, but similar to CBC in its proofs of security, which are not available for the ECB mode. Although the IAPM mode has not yet been implemented, similar performance is expected for a serial implementation as that achieved for the IACBC mode.

A letter was provided by IBM concerning their patent policy for these schemes (available at http://www.nist.gov/modes). Charanjit will investigate whether licenses can be granted on a royalty free basis.

2.3 **On Message Integrity in Symmetric Encryption** and **Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes**

Virgil Gligor of VDG, Inc. presented two papers written by himself and Pompiliu Donescu. In the first paper, “On Message Integrity in Symmetric Encryption,” different notions of message integrity for block-oriented symmetric encryption are explored, along with their relationships. These notions are expressed as a combination of integrity goals (e.g., protection against existential forgery and assurance of plaintext integrity) to be achieved in the face of different types of attacks (e.g., chosen-plaintext and ciphertext-only attacks). The integrity notions are partially ordered by a dominance relation. Defining the notions of integrity in terms of this dominance relation, enables a characterization of the relative strengths of various symmetric encryption modes.

In the second paper, “Fast Encryption and Authentication: XCBC and XECB Authentication Mode,” two mode types are proposed: the XCBC mode and the XECB-MAC mode. These families of modes are similar to the IACBC and IAPM modes discussed in Section 2.2, but the whitening sequences are not required to be pairwise independent; this allows better performance at the cost of relaxing the security bounds that can be proven.

The XCBC modes provide both confidentiality and integrity protection in a single pass over the data. These modes detect integrity violations at a low cost in performance, power and implementation, and can be executed in a parallel or pipelined manner. The performance and security of these modes depends on the performance and security of the underlying block cipher algorithm (e.g., AES). Both stateful and stateless variants are provided. In addition to message integrity, these modes have the following properties:

- Support for real-time message authentication,
- Support for multiple encryption modes (i.e., modes other than CBC could be used),
- Support for interleaved-parallel or pipelined encryption,
- Incremental updates of encrypted data (i.e., the incremental update of data structures is possible),
- Support for architecture-dependent parallel encryption, since there is no ciphertext chaining or requirement for an a priori knowledge of the number of processors, and
- Resistance to key attacks can be implemented, if required, in a manner similar to that of DESX.

The paper also provides evidence for the security of the XCBC modes against both adaptive chosen-plaintext and message-integrity attacks. The performance of the XCBC modes in software implementations is only minimally degraded in comparison to the CBC mode, and is superior to the CBC mode and other similar modes that are used to provide message integrity.

The XECB-MAC modes provide message authentication, can be operated in a fully parallel or pipelined manner, and support incremental updates and out-of-order verification. These modes are intended for use either stand-alone to protect the integrity of plaintext messages, or with encryption modes that have similar properties, whenever separate secret keys are used to provide confidentiality and integrity. Both stateless and stateful variants of XECB-MAC are provided. XECB-MAC's properties include:

- The XECB-MAC modes are intended to be secure against adaptive chosen-plaintext attacks.
- Parallel or pipelined operation is possible.
- The XECB-MAC modes are incremental with respect to block placement.
- Verification of the authentication code can proceed even if the blocks are received out of order.
- The number of block encryption computations for XECB-MAC is the same as the number of block encryption computations for CBC-MAC. In sequential implementations, the performance of XECB-MAC is slightly lower than that of CBC-MAC because of additional processing. However, the ECB-MAC mode can take advantage of parallelism or pipelining to improve its performance.

A third mode was presented at the workshop: the PM-XOR mode. This mode is a stateless fully parallel mode that is similar to Jutla's IAPM mode (see Section 2.2); however, the S_i elements are not pairwise independent. Refer to the presentation for further information.

During the presentation, Virgil proposed three classes of modes for the standard, based on preferred operational environments that require different performance: Low to mid-end (simple extensions of CBC), mid- to high-end (using a single confidentiality and integrity key), and high-end (separate or independent key for confidentiality and integrity).

Virgil stated that three patent applications had been filed for these modes, for the purpose of "bringing people to the table", to avoid having this work "pre-empted".

2.4 CTR-Mode Encryption

Phillip Rogaway of the University of California at Davis presented a paper co-authored with Helger Lipmaa and David Wagner. This paper strongly recommends that a counter (CTR) mode be included in the modes standard. In CTR mode, a series of unique counters are input into the encryption algorithm, and the output is essentially used as a one-time pad to encrypt the data blocks. CTR mode has significant efficiency advantages over the standard encryption modes without weakening the security. Other advantages include:

- Hardware efficiency: The CTR mode is fully parallizable.
- Preprocessing: Because the cryptographic work is independent of the data to be encrypted, preprocessing can be used in some environments to increase speed.
- Random-access: For applications such as hard-disk encryption, the encryption of each block of data can be independent of the encryption of every other block of data.
- Provable security: The concrete security bounds for the CTR-mode are at least as good as those for the CBC mode.
- Simplicity: Both encryption and decryption depend only on the encryption transformation of the underlying block cipher algorithm. Therefore, the decryption transformation need not be implemented.
- Messages of arbitrary bit length: Data may be of any length. No bits are wasted, and no padding is required.

The following perceived disadvantages were discussed, along with the author's responses:

- No integrity: No message integrity is provided, but the authors point out that this is true of other modes, such as the CBC mode. A message authentication code (MAC) is typically used whenever message integrity is required.
- Error propagation: Any bit-flip in a block of ciphertext is localized to the corresponding bit in the resulting plaintext after decryption. The authors suggest that any error correction or detection should occur at a different architectural level.
- Stateful encryption: Normally, the sender is stateful, in that the state of the counter needs to be maintained. However, in contexts where the state cannot be maintained, and a random ralue can be obtained, the CTR mode need not maintain state. The authors point out that for the CBC mode, the maintenance of the state initialization vectors poses essentially the same issues.
- Sensitivity to usage errors: Counter values must not be re-used; otherwise, a catastrophic breach of security may occur. It was conceded that this concern has some validity, but the authors point out that catastrophic errors can result from improper execution of the mode.

- Interaction with weak ciphers: The security of the CTR mode depends on the security provided by the underlying cipher algorithm. The authors respond that the AES should be regarded as a strong cipher.

The paper recommends that the standard should clearly describe a small number of recommended ways to form the counter.

2.5 Key Feedback Mode: a Keystream Generator with Provable Security

John Håstad of NADA, the Royal Institute of Technology in Sweden, presented a paper written by himself and Mats Naslund. The underlying idea of their work is that it is not necessary to assume (as is implicit for, say, OFB or CTR mode) that a block cipher like the AES is itself a good pseudo-random bit generator. Instead, the authors wish to rely only on the weaker assumption that the block cipher is hard to invert, i.e., it is difficult to derive the key from given ciphertext (and plaintext). From this assumption, it is possible to construct a pseudo-random bit generator from the block cipher with certain provable properties.

There are several such constructions in the literature: the authors' "BMGL" generator builds on the work of Blum, Micali, Goldreich, and Levin. Key feedback (KFB) mode is the authors' name for the mode of operation based on the BMGL generator. In KFB, a fixed random message is used as the input to the block cipher algorithm, and the output is fed back as the key for each successive application of the block cipher. An output generation matrix is applied to the outputs of the block cipher to form the keystream, which can then be applied to the plaintext, as in OFB, CFB, or CTR modes.

Properties of the KFB mode include:

- Error robustness: Errors are confined to the bits in which they occur.
- Synchronization: KFB requires synchronization of the key stream by the sender and receiver.
- Key stream advance/rewind: KFB does not provide a "random access capability" like that provided for the counter mode.

The authors acknowledge that their construction may not be applicable to high-speed, real time applications because the provable security properties come at a cost in efficiency compared to, say, OFB mode. However, they suggest that a cautious user may find KFB to be a useful alternative. The authors are not aware of any patents on this mode.

2.6 OCB Mode: Parallelizable Authenticated Encryption and PMAC: A Parallelizable Message Authentication Code

Phillip Rogaway of the University of California at Davis proposed two new modes: the Offset Codebook Mode (OCB) and the Parallelizable MAC mode.

The OCB mode is based on the work of Jutla (see Section 2.2) and Gligor and Donescu (see Section 2.3). This mode provides both confidentiality and integrity in a manner that is

parallelizable (i.e., different blocks can be processed at the same time). Other properties of this mode include:

- The data to be processed need not be an even multiple of the block length (e.g., if AES is used, the block length is 128 bits; the data need not be forced to a multiple of 128 bits in length).
- Only two extra cipher calls beyond that needed for encryption alone are required to process the data.
- While a non-repeating nonce is required, it need not be unpredictable (e.g., a simple counter may be used).
- The offset (i.e., the whitening) values used in the OCB mode depends only on the key – it only needs to be computed once at the beginning of a key's cryptoperiod.
- Only a single key is used for this mode, as opposed to separate keys for encryption and authentication, as is done in current systems.
- Three variants of this mode are possible.

Proofs of the security properties of the OCB mode are under construction.

PMAC is similar to the XECB mode proposed by Gligor and Donescu (see Section 2.3). This mode also uses an offset. The PMAC mode is fully parallelizable and achieves existential unforgeability under an adaptive chosen-plaintext attack; a proof of this security claim is currently being prepared. Other properties of this mode include:

- No nonces or random values are required.
- A minimum number of invocations of the block cipher algorithm are required: one per data block.
- The length of the data need not be a multiple of the block size of the cipher algorithm.
- Only one key is required.
- Only one invocation of the block cipher algorithm is required to compute the initial offset.
- Three variants of this mode are possible.

Rogaway indicated that the algorithm descriptions for OCB and PMAC are intended to allow for various implementation tricks.

2.7 A Suggestion for Handling Arbitrary-Length Messages with CBC MAC

John Black of the University of Nevada in Reno proposed a version of the CBC MAC mode described in a paper written with Phillip Rogaway. The identified advantages of this mode include:

- The data that is processed using the mode need not be a multiple of the block length of the cipher algorithm used to compute the MAC.
- A minimum number of cipher calls are required, one for each block of the data or fraction thereof.
- The mode is simple and familiar.
- The cipher is invoked with only one key of the three keys that are required in the scheme.
- Standard security properties of the mode have been proven.

Actual and perceived disadvantages of these modes include:

- As with any of the variants of CBC-MAC, it is not possible to extract much parallelism in the computation of this mode.
- While some CBC MAC variants use different keys to improve resistance to key-search attacks, this mode does not provide this resistance. However, when AES is used, this should not be a concern because of the strength of AES.

2.8 Block Chaining Modes of Operation

Bart Preneel of the Katholieke Univeristeit Leuven presented a paper by Lars Knudsen that proposes the Accumulated Block Chaining (ABC) mode that has infinite error propagation (i.e., an error in one ciphertext block propagates to all subsequent ciphertext blocks). A mode with this property is suited for situations where errors during transmission are either unlikely to occur or are accommodated by non-cryptographic means, such as error-detecting codes or retransmissions. The advantages of a mode with error propagation are:

- It has better diffusion properties for both the encryption and decryption operations than modes that provide error recovery.
- In general, a mode of this type is less vulnerable to birthday attacks.
- When encrypting *s*-block messages, the mode resembles a large *sn*-bit block cipher when only one pass through the *s* blocks is allowed.
- There are implementational advantages, since the encryption and decryption operations can be equal.

The ABC mode has been designed with low overhead as a goal - little extra work is required beyond that required for the ECB mode. However, the ABC mode does not provide message integrity; a separate MAC function is required.

3. Discussions

3.1 Are there any other modes that should be considered?

No other modes were suggested by the attendees. However, there was a comment that there would be an advantage if a separate key for integrity was not required; a key derived from the encryption key would be okay. It was noted that the large key lengths required for the new algorithms are stressing the capabilities of the key establishment mechanisms (e.g., with Diffie-Hellman).

3.2 What are the issues with regard to applications and environments?

There were several comments made with regard to IPSEC:

- While a parallelizable mode is attractive, the performance of Rijndael (the proposed AES algorithm) already allows CBC to be used at a high speed.
- Fault propagation is not an issue, since everything is resynchronized for every packet.

- A counter is currently carried in the header, so counter mode will not require anything extra to pass the counter along. Since the protocol will not operate correctly if the counter is mishandled, there should be no concern about bad implementations of counter mode.
- Since IPSEC already takes care of padding, a requirement for padding in a mode should not be an issue.
- For modes with fixed overhead per packet per message encryption, a small number of additional encryptions is not a problem. However, there is a concern about a variable number of additional encryptions based on the message size.
- The computation of a MAC needs to be as fast as the encipherment process. Authentication and encryption are handled separately in IPSEC, so encrypting with integrity (authenticated encryption) is not really useful at present. However, IPSEC can accommodate an authenticated encryption mode with an additional block when the mode is used for one or more of the encryption and authentication processes.

With regard to IKE (Internet Key Exchange protocol): Negotiation is used for establishing the encryption and authentication modes; the availability of an integrated mode (i.e., authenticated encryption) would need a new negotiation capability.

3.3 What about encryption vs. decryption?

Comments were received from Richard Schroeppel and Tom Phinney about switching the encryption and decryption functions (see the email comments at http://www.nist.gov/modes). Vincent Rijmen made the following comment: Rijndael is equally fast for encryption and decryption for implementations using tables; key scheduling can be done either backward or forward, given the correct starting point. No mode is needed to run in backward direction for better encryption performance.

3.4 What modes are really needed?

Virgil Gligor proposed three “classes” of modes in his presentation (see Section 2.3).

Phil Rogaway stated that there is no need for an ECB mode, since this mode is not useful for more than one block; OFB and CFB should have someone who supports them to make them worthwhile as modes.

Various attendees felt that a counter mode for high speed hardware implementations is needed, as well as a mode for a parallelizable MAC or for high speed applications.

3.5 How should NIST address intellectual property issues?

The following sentiments were expressed by the attendees:

- The modes should be freely implementable; some systems can’t integrate a patented algorithm because of distribution problems.
- Research organizations have difficulty negotiating patents.
- Patents may not be a difficult issue. The patents could be purchased, as was done for DSA. A patent may not be enforced, since many patent applications are intended to

protect ideas, not necessarily to make money. Patent royalties could be donated to universities. With regard to IBM patents, IBM has been very flexible about the use of their patents in the past;

- NIST should try for royalty-free modes. The AES algorithm itself is intended to be royalty-free worldwide, and each submitter of an AES candidate was required to renounce royalties from the eventual AES algorithm. Therefore, it would seem to be unfair to allow any entity to collect royalties on the use of a particular mode of the AES.
- The modes should be around longer than AES, so the modes and AES need not be treated the same; modes are fundamental algorithms.
- There may be interactions between various IP claims; this may introduce complications when a mode is based on a patented mode.
- Patents would be a barrier to implementation and experimentation. This may not be a problem with commercial products. However, modes with patents would probably not be used by the research community; it's a nuisance to get a license even if it's free.
- Inventing a new mode is not very different from inventing a block cipher algorithm; even if the US government were to buy the patents, patent applications may still exist elsewhere (e.g., Europe).

3.6 Use a "staged" process for defining the modes?

The AES standard is expected to be signed by the Secretary of Commerce during the spring or early summer of 2001. Some modes need to be in place at that time to permit approved implementation and testing. However, other modes appear to need further study. NIST will probably use a two stage process: specify some initial modes by the time that AES is signed (e.g., ECB, CBC, CFB, OFB and one or two varieties of counter mode), and add other modes at a later time. The workshop attendees felt that this was a reasonable approach.

3.7 What is the timeline for developing the modes?

Some modes will be defined next spring by the time the AES standard is signed. NIST would like to make the period for determining extra modes relatively short. A comment period for other modes could close in mid to late spring, for example. A workshop may be desirable for discussing the other modes to be included. NIST would prefer to have the modes standard mostly completed within the next year.

3.8 What are the validation considerations?

The attendees made the following points:

- The modes should be well defined and easy to test.
- Counter mode should be well defined.
- For counter mode, the type of input needs to be flexible; some applications may need sender ID or other information. There may need to be a method for defining the state vector (input) to satisfy everybody.
- If flexibility for determining the input for counter mode is allowed, then methods that are not specified in the standard would need to be individually evaluated. This would significantly increase the cost of validation.

- Counter mode must be tailored to its application in order to be used properly, since some applications have standardized contexts. We should define a few specific inputs/state vectors/examples.
- IPSEC could derive a nonce from the Diffie-Hellman keying material. Using a manually keyed counter mode would be problematic.
- NIST could request submissions for counter mode inputs.

3.9 Other Issues

Other views that were expressed included:

- Proofs are needed for the modes.
- Criteria should be developed and a schedule should be proposed as was done for AES. There should be a call for proposals; the proposed modes could benefit from further study. Another workshop may be appropriate.
- The selection of modes is as important as the selection of the block encryption function.
- Don't rush the selection of modes.
- Consider defining a hash function based on AES. ISO 10118:2 contains 2 schemes by IBM; ISO is currently revising this standard.

4. Conclusion and Future Plans

NIST will take the opinions of the workshop attendees and the public comments into account. The public should continue to send comments to EncryptionModes@nist.gov. As stated above, NIST will specify some initial modes for AES (e.g., ECB, CBC, CFB, OFB and counter mode), followed by other modes at a later time. NIST will develop criteria for the other modes and schedule a comment period for their receipt. A schedule for these activities will be provided on the modes page in the near future. The modes page is available at http://www.nist.gov/modes.