Customizable cryptographic architecture for government and military communications applications

Conference Paper · January 2004

DOI: 10.1109/MILCOM.2004.1493284 · Source: IEEE Xplore

3 authors, including:

Michael Kurdziel
Harris Corporation
17 PUBLICATIONS **58** CITATIONS

SEE PROFILE

CUSTOMIZABLE CRYPTOGRAPHIC ARCHITECTURE FOR GOVERNMENT AND MILITARY COMMUNICATIONS APPLICATIONS

Michael T. Kurdziel,
Robert P. Clements,
Gary R. Dennis
Harris Corporation, RF Communications Division
Rochester, New York 14610

ABSTRACT

With the continued international proliferation of military communication technology, equipment vendors are challenged to provide communications security (COMSEC) solutions that are appropriate for these applications. Addressing these challenges is complicated by the fact that no encryption standard exists for international government and military communications applications. Harris Corporation has leveraged its expertise in encryption algorithm design, Type 1 hardware design, and embedded system design to develop a cost-effective, customizable encryption algorithm for military and government communications applications. The algorithm is the second generation of the highly successful Citadel™ cryptographic algorithm. It was designed against a military/government threat model and specifically addresses the rigorous requirements dictated by these applications. It does so in a manner that is both flexible and cost effective. This paper will begin with a brief review of the requirements for military-grade encryption products. It will continue with an overview of the algorithm design along with discussion on the features responsible for its military grade strength.

INTRODUCTION

The wide acceptance of Harris's Citadel™-based encryption solution underscores the increased customer-focus on appropriate encryption solutions for military and government communication systems. Recent changes in U.S. government export laws now allow higher strength crypto solutions with longer key lengths to be offered to international military/government communications customers.

This paper describes Harris' next generation non-Type 1 military/government encryption solution. The core of the solution is the 256-bit Citadel II™ algorithm. This algorithm is designed against a military threat model and provides the enhanced strength required by military/government communications applications. Further, the algorithm provides a configuration that is interoperable with current Citadel I™-based applications and a configuration that is fully disclosable.

Section 2 provides a summary of requirements for military/government encryption solutions. Section 3 provides an overview of the new architecture. Section 4 discusses aspects of the cryptographic and functional performance of the new architecture, the paper is concluded in Section 5 and references are provided in Section 6.

OBJECTIVES/REQUIREMENTS

This section provides a summary of the unique requirements for cryptographic solutions intended for military/government communications applications. Refer to [5] for a detailed discussion. Requirements are presented in three categories; those requirements that apply to the cryptographic algorithm, requirements that apply to cryptographic device that embeds the algorithm, and the functional requirements imposed on the device by the system.

Algorithm Requirements

Military/government communications applications necessitate that additional requirements be considered when identifying an appropriate cryptographic algorithm. Some of these are:

1) The cryptographic algorithm must be designed to be secure against all known cryptanalysis techniques. See [1], [6], [8] and [9]. Commercial algorithms are often fielded with known weaknesses because commercial users deem them un-exploitable. This is unacceptable for military/government applications.
2) The cryptographic solution must be designed so that no weaknesses are observed even if a technique does not yet exist to exploit the weakness. For example, a cryptographic algorithm needs to be designed such that the output is uniformly distributed. Any exception to this requirement means that information is leaking out whether or not a means of exploiting that fact exists.

3) For military/government communications applications, some means must be employed to prevent the algorithm from proliferating to the public sector. This is essential as it reduces the possibility of focused efforts at analyzing the algorithm outside the adversary's national intelligence agency. More importantly, it substantially reduces the possibility of newly developed, target-specific attacks from being published. Options include classifying all or part of the algorithm, holding all or part of the algorithm as proprietary or by employing a customizable algorithm.
4) The cryptographic strength of the algorithm cannot depend on non-disclosure of the algorithm design, access control of the equipment, or on the assumption that data required to mount an attack will be unavailable. All security must lie in the knowledge of the cryptographic key and the strength of the algorithm.

Device Requirements

Further requirements are imposed on the algorithm embedment into a device such as an FPGA or an Application Specific Integrated Circuit (ASIC). These are listed below.

1) Military/government encryption solutions must be designed with no compromise of the objective of high cryptographic strength. The highest possible cryptographic strength must be achieved even at the expense of other metrics (gate count, power consumption, cost, speed etc.).
2) No functional means should exist that allows unsecured data to be extracted once it has been input to the device. For example, key data should not be able to be extracted unless encrypted. Plaintext, of course, should not be able to be extracted unless encrypted.
3) No detectable variation in any parameter that is measurable external to the device should correlate to the structure of sensitive data or operations executing within the device. As an example, changes in the current draw and processing delay of a device are known to reveal information on internal operations [2], [3]. Therefore a military/government encryption engine should be constructed using only constant execution operation implementations. Unfortunately, this means that optimal methods often cannot be used.
4) It must be assumed that the adversary will have access to all equipment, techniques and data required to mount an attack. It should further be assumed that the adversary has become expert on the communication equipment and the algorithm design.
5) The adversary's general purpose computing capability and custom hardware capability must be considered so the threat of a practical brute force search of the key space is adequately addressed. See [5] for additional discussion.

System Requirements

Different modes of communication impose different functional requirements that extend down to the cryptographic engine.

1) Wireless communication channels have higher Bit Error Rates (BER) relative to wired channels, especially as they approach maximum operating range. As such, wireless systems are not tolerant of error extension imposed by a cryptographic engine. Therefore, any cryptographic solution intended for these applications needs to include a cryptographic mode that does not introduce error extension.
2) In some applications error extension is actually desirable. It provides a mechanism whereby targeted manipulation of data on low BER channels can easily be detected. So it is useful to include a cryptographic mode with this property to allow a spoof detection mechanism to be provided to the system.
3) Particular properties of a channel, such as modulation techniques, routing mechanisms, etc. sometimes result in susceptibility to loss of bit synchronization. This, of course, results directly in loss of crypto synchronization. It is therefore, useful to include a cryptographic mode with the ability to self-synchronize.

ALGORITHM OVERVIEW

This section describes a cryptographic algorithm design that, when properly implemented, meets the requirements summarized in the previous section and those described in [5]. A block diagram is shown in Figure 1. The basis of the architecture is a Harris proprietary cryptographic architecture described in [4]. This new architecture builds on the strengths of its predecessor and can be configured to be interoperable with it. Detailed discussion of its design is beyond the scope of this paper. However, the following high-level description is provided.

The architecture consists of one major block called the Block Cipher Algorithm, shown in Figure 1. The block cipher operates in one of the three standard configurations or "modes" described in the previous sections. It is non-recursive and is composed of an input unit (14), twelve encryption stages (1-12), an output unit (15) and a key scheduling unit (13). Both figures illustrate a 128-bit block implementation, which is the current convention.

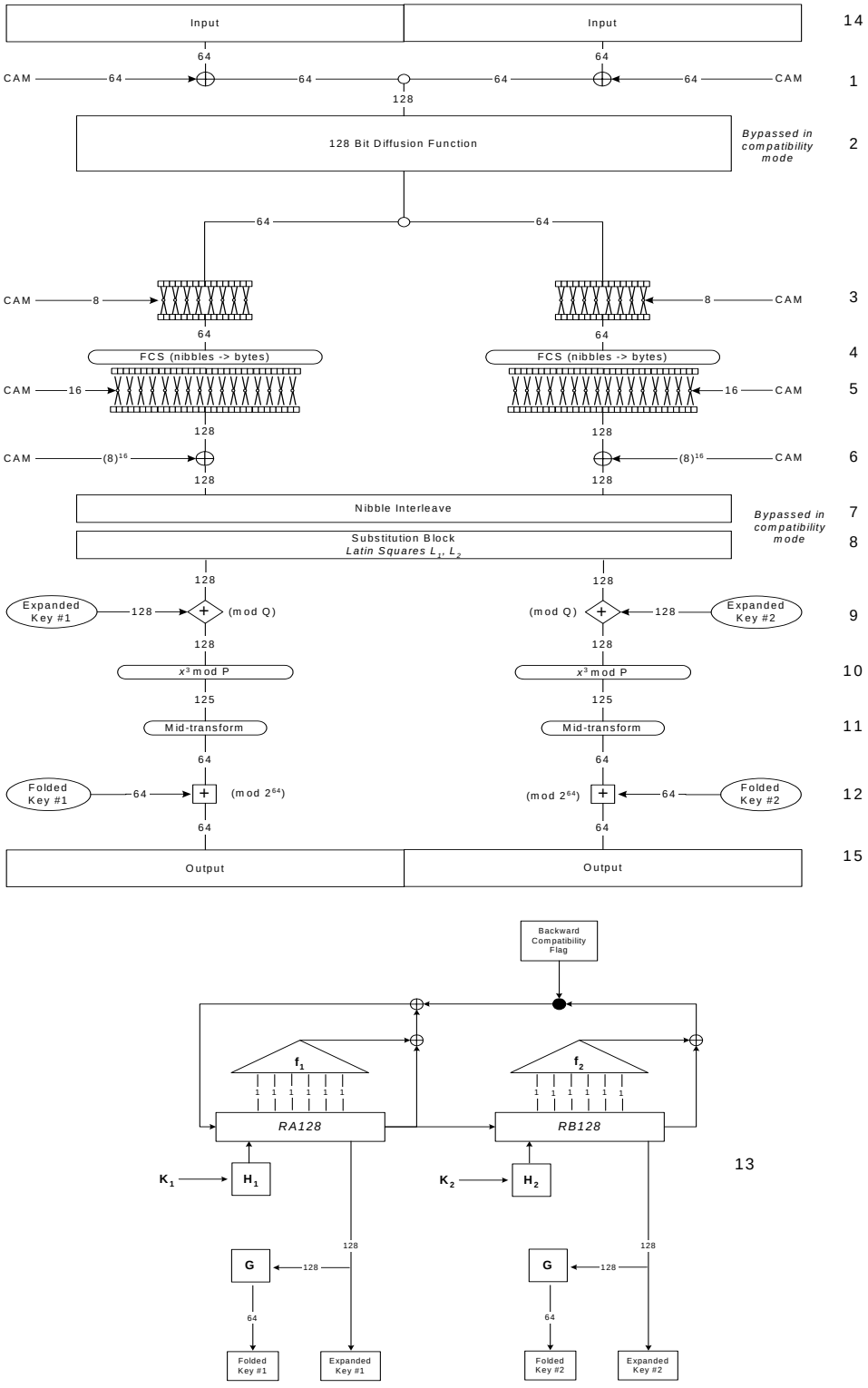

Figure 1. Detailed Block Diagram

Each encryption stage has at least a 128-bit I/O space. Encryption stages (1), (6), (9), (10) and (12) consist of pairs of modular arithmetic logic. Stages (1), (6), (9) and (12) perform modular addition pairs and stage (10) performs a pair of modular cubing operations. Each stage employs a different modulus. Stage (11) is a non-invertible "Mid Transform". In addition, stages (1), (9), (10), (11) and (12) cannot be segmented into a set of lower level operations. These design features make the Block Cipher Device secure against popular cryptanalysis techniques.

Stages (3) and (5) are "nibble swapping" pair blocks. These blocks allow the structure of the algorithm to be changed based on an externally applied input called a Customer Algorithm Modification (CAM) vector. Stage (4) is a pair of customizable Substitution/Expansion blocks. This allows custom operations to be inserted into the algorithm structure and unique algorithm variants to be implemented. The structure of block (4) is not arbitrary and must meet the requirements of secure cipher design. Custom Substitution/Expansion block designs are disclosed only to the intended end user.

Stages (2), (7) and (8) are mixing operations. Each of these operations accepts two separate 64-bit inputs and mixes them such that their information content is diffused across a single 128-bit output. Operation (2) has a customizable structure. Operation (7) is a nibble-interleave function and operation (8) is a substitution function consisting of a basic scaled "Latin Squares" mapping.

The Key Scheduling Unit (13) accepts externally applied input variables, K_1 and K_2, and deterministically generates two sets of pseudo random bit patterns, Folded Key 1 and 2 and Expanded Key 1 and 2. These patterns are used as operands for stage (9) and stage (12) respectively. The Key Scheduling Unit contains two customizable substitution functions, f1 and f2, consisting of nx1 look-up tables.

Note that this cipher can be configured to interoperate with its predecessor [4]. In this "Backward Compatibility Mode", stages (2), (7) and (8) are bypassed and the key scheduler (13) is configured to produce the same pseudo random bit patterns, Folded Key 1 and Expanded Key 1, as the predecessor.

The input unit (14) assembles and buffers input data to the algorithm and the output unit (15) assembles and buffers output data from the algorithm.

CRYPTOGRAPHIC PERFORMANCE

The modern adversary has a number of analytical techniques at their disposal to attack a cryptographic algorithm. These attacks vary from those that only require knowledge of the key length and data protocol to attacks that utilize access to the chip boundary and full knowledge of the algorithm design. The Harris Citadel II™ algorithm has been designed to be secure against each of these types of attacks. This section will discuss the common attacks against cryptographic algorithms.

Brute Force Attack

The most well-known attack against any cryptographic algorithm is the "Brute Force" key space search. For this attack, the adversary simply cycles through all possible key patterns until they find a pattern that enables them to decrypt the protected data. The Harris Citadel II™ algorithm includes many design elements that reduce the likelihood of a successful brute force attack [5]. It contains structures that execute slowly in software but quickly in hardware to limit the speed of a software-based brute force attack. The Key Scheduler Unit contains processing latencies to intentionally reduce the speed at which keys can be loaded without hindering the encryption throughput. The Citadel II™ device also utilizes a 256-bit key. This is twice as long as its successful predecessor, the Citadel™, which, in 2002, was determined to be secure for at least 50 years.

Timing and Power Analysis

Timing and power analysis attacks are physical attempts to gain insight to an algorithm's construction and secret key variable [2] [3]. The Harris Citadel II™ algorithm does not vary the length of processing based on the structure of the key variable. It implements functions that require a fixed number of clock cycles, and registers all input and output data. The device was also designed to have a nearly uniform probability distribution at each of the intermediate stages while utilizing highly paralleled integrated circuits. These precautions were taken to make sure a Timing Analysis attack or a Differential Power Analysis attack will not yield information regarding the secret key or Customer Algorithm Modification blocks.

Differential and Linear Cryptanalysis

The Citadel II™ algorithm is secure, by design, against "Differential" and "Linear" Cryptanalytic techniques. It is a 128 bit block cipher. The algorithm is not iterative and can not be segmented into "rounds" for analysis. It consists of a series of operations or sub-blocks having at least 64 bit I/O spaces. Most, of which, can not be segmented into lower level operations or smaller I/O space. The

operations, which comprise the algorithm, consist of non-linear mapping functions, mixed-mode arithmetic, non-linear mixing functions and non-invertible transforms. These operations prevent differential characteristics or a significantly biased linear approximation from existing between the algorithm's input, output and key spaces.

Analytical Modeling

In an effort to uncover vulnerabilities in the new Advanced Encryption Standard (AES), researchers have begun exploring whether a low order analytical model over a single mode of arithmetic can be developed to attack an algorithm [1] [7]. Many of the features described in the previous section prevent analytical modeling from being applied to the Citadel II™ algorithm. The algorithm is comprised of non-linear mapping functions, random permutation mappings, four modes of arithmetic, non-linear mixing functions and non-invertible transforms. No one can predict the future, but it is difficult to imagine that a single mode description of this algorithm could ever be developed.

FUNCTIONAL PERFORMANCE

A number of different traffic modes are necessary to operate in different transmission environments. Although the Citadel II™ algorithm can be operated using any block cipher traffic mode, the following three modes cover a wide range of channel conditions. For example, many wireless applications require a traffic mode that supports late net entry or can recover from channel fading, while others are intolerant of error extension. A set of modes that provide this flexibility include Cipher Feedback mode (CFB), Counter Mode and Self Synchronizing Cipher Feedback Mode (SSCFB). These three modes are discussed below, including the benefits and drawbacks of each. The reader is referred to [9] for detailed discussion.

Block Cipher Feedback Mode (CFB)

CFB is a standard cipher block mode. The Initialization Vector (IV) or initial state is block encrypted to produce the first block of key stream. The key steam is combined, mod 2, with the first block of PT data and becomes the first block of CT. This CT block is transmitted and also fed back as the next state. Decryption is performed using the IV as the initial state. The resulting key stream is combined, mod 2, with the first block of CT data to produce the first block of PT. The first block of CT is also used as the next input state. This standard traffic mode resynchronizes once for every block of input data providing efficient use of processing resources and power. In addition to use as a traffic mode, CFB mode can also be used to calculate a Message Authentication Code (MAC) to verify integrity of data in a file. From an implementation perspective, CFB mode offers an advantage over Cipher Block Chaining Mode (CBC). CFB mode uses the same structure for encrypt and decrypt, where as CBC requires a separate decrypt structure.

Error Extension:

- 128 bit error extension (for one incorrect bit of CT in, 129 bits of error will result). Synchronization is maintained.
- One bit of CT dropped results in a loss of synchronization. In order to resynchronize, encryption and decryption have to be disabled, then enabled, after a new IV is reloaded.

Counter Mode

Counter Mode performs block encryption without feedback. The IV is used to initialize the input state register. The input state register is configured as a maximal length Linear Feedback Shift Register (LFSR) and is incremented after each input state is encrypted. The encrypted state is combined, mod 2, with PT to produce CT. The next input state is determined by incrementing the LFSR. Decryption is accomplished in the same way, combining CT, mod 2, with the key stream. (An equally valid implementation of this mode can use an up counter in place of the LFSR [10] [11]). Counter mode has no error extension, making it useful in low bit rate or noisy transmission environments.

Error Extension:

- No error extension (for one incorrect bit of CT in, one bit of error will result). Synchronization is maintained.
- One bit of CT dropped results in a loss of synchronization. In order to resynchronize, encryption and decryption have to be disabled, then re-enabled after a new IV is loaded.

Self Sync Mode (SSCFB)

SSCFB mode is a variation of CFB mode. This mode auto resynchronizes at the end of every block or when a predefined CT pattern is matched. The CT pattern length can be chosen to determine how often an auto resync occurs. When the CT pattern is matched, the last 128 bits of CT is feed back and used for the next input state. If the CT data is corrupted during transmission, including loss of frame synchronization, cryptographic synchronization will be reestablished when the CT pattern occurs. (Note, the CT resynchronization pattern does make SSCFB susceptible to a denial of service attack).

This mode provides resynchronization from the loss or removal of data during transmission. This attribute is

particularly useful for late net entry or fading channel conditions. Cryptographic synchronization can be acquired after a transmission has started and the IV has been missed. The pattern length can be selected to customize how often a resynchronization occurs. For example, Citadel I and II™ provide pattern length options of 8, 11, 13 or 16 bits for average auto resynchronization at 256, 2048, 8192 and 65536 bit intervals respectively [4].

Error Extension:

- 128 bit error extension.
- One bit slip results in a temporary loss of synchronization until an auto-resync occurs.

CONCLUSION

This paper describes the Harris Second Generation Customizable Cryptographic Architecture. This architecture meets the INFOSEC requirements for military and government applications. The architecture provides the user with security autonomy, cryptographic strength and a variety of functions traditionally available in Type 1 applications. The architecture is used in the Citadel II™ Cryptographic Device. Citadel I and II™ are Harris' prime export encryption solutions for all of its products requiring security.

REFERENCES

[1] N. Courtois, J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations," Proceedings of ASIACRYPT 2002, LNCS Springer, December 2002.

[2] P. Krocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS and Other Systems," Advances in Cryptology: Proceedings of CRYPTO '94, Springer-Verilag, 1994, pp.104-113.

[3] P. Krocher, J. Jaffe, and B. Jun, "Differential Power Analysis," Advances in Cryptology: Proceedings of CRYPTO '99, Springer-Verilag, 1999, pp.388-397.

[4] M. Kurdziel, R. Clements, "Harris Customizable Cryptographic Architecture," Proc. IEEE, Mil. Comm. Conf., Oct. 1998, pp. 1033-1037.

[5] M. Kurdziel, J. Fitton, "Baseline Requirements for Government & Military Encryption Algorithms," Proc. IEEE, Mil. Comm. Conf., Oct. 2002.

[6] X. Lai, J. Massey, S. Murphy, "Markov Ciphers and Differential Cryptanalysis," Advances in Cryptology-EUROCRYPT '91 Proceedings, Springer-Verilag, 1991, pp.17-38.

[7] S. Murphy, M. Robshaw, "Essential Algebraic Structure Within the AES," Advances in Cryptology: Proceedings of CRYPTO '02, Springer-Verilag, 2002.

[8] K. Nyberg, "Differentially Uniform Mappings for Cryptography," Advances in Cryptology-EUROCRYPT '93, Springer-Verilag, 1994, pp.55-64.

[9] B. Schneier, "Applied Cryptography 2nd ed.", John Wiley and Sons, 1996.

[10] FIPS Publication 81, "DES Modes of Operation." US.DoC/NIST, December 1980.

[11] NIST Special Publication 800-38A, "Recommendation of Block Cipher Modes of Operation.",US.DoC/NIST, July 2001.