An Overview of Cryptanalysis Research for the Advanced Encryption Standard

Alan Kaminsky[1], Michael Kurdziel[2], Stanisław Radziszowski[1]
[1]Rochester Institute of Technology, Rochester, NY
[2]Harris Corp., RF Communications Div., Rochester, NY
ark@cs.rit.edu, mkurdzie@harris.com, spr@cs.rit.edu

***Abstract* - Since its release in November 2001, the Advanced Encryption Standard (NIST FIPS-197) has been the subject of extensive cryptanalysis research. The importance of this research has intensified since AES was named, in 2003, by NSA as a Type-1 Suite B Encryption Algorithm (CNSSP-15). As such, AES is now authorized to protect classified and unclassified national security systems and information. This paper provides an overview of current cryptanalysis research on the AES cryptographic algorithm. Discussion is provided on the impact by each technique to the strength of the algorithm in national security applications. The paper is concluded with an attempt at a forecast of the usable life of AES in these applications.**

Keywords-Advanced Encryption Standard; AES; Cryptanalysis; Side Channel Attacks

1.0 INTRODUCTION

In 2003, the National Security Agency took the unprecedented step of approving a public-domain encryption algorithm, AES, for classified information processing. Prior to this milestone, all encryption algorithms approved by the NSA for classified processing were, themselves, classified. The strength of any good encryption algorithm is not enhanced by holding the design as secret. In fact, a public domain encryption standard is subject to continuous, vigilant, expert cryptanalysis. Any breakthroughs will very likely be available to users as well as their adversaries at the same time.

In consumer applications, this isn't as much of a problem, but in military communication applications it can be disastrous. Here, the adversary can have national intelligence agency level resources and can exploit vulnerabilities as soon as they are identified. If practical vulnerabilities are found, there will be a period of reduced confidence until a new algorithm can be installed.

It is prudent for users and providers of military communications equipment to stay abreast of the progress and trends on cryptanalysis of AES. Facilitating this process is the objective of this paper.

Section 2 presents a summary of the past and current areas of research on cryptanalysis of the AES. This section is divided into 5 subsections. The first discusses attacks that pre-existed AES and were addressed as part of its design. The second discusses progress in the new area of algebraic attacks. The third discusses progress on SAT solver and hybrid attacks. Subsection 4 discusses the progress made in side-channel cryptanalysis. Subsection 5 presents a summary of related-key vulnerabilities and distinguishing attacks on AES. These are particularly relevant when AES is used in applications other than traffic encryption (such as hash functions). Section 3 provides discussion of the current strength of AES in national security applications. A forecast of the usable life of AES in these applications is attempted. The paper is concluded in Section 4.

2.0 CURRENT AREAS OF RESEARCH

2.1 Pre-Existing Attacks

2.1.1 Linear Cryptanalysis

Linear cryptanalysis exploits approximate linear relationships that exist between inputs and outputs of a function block [1]. In the case of a block cipher, linear combinations of plaintext patterns and linear combinations of ciphertext patterns are compared to linear combinations of key bits. The goal is to discover a relationship that is valid either significantly more or less than 50% of the time. This will constitute a "biased" approximation which can then be used to determine key bits. A linear attack would consist of, first, identifying a biased linear approximation to the algorithm. Then apply plaintext patterns, retrieve the resulting ciphertext patterns and linearly combine them (in a mod-2 sense) according to the approximation. The result of this operation will be, with some probability, a linear combination of key bits. Enough trials are run such that good guesses can be made of some key bit values. More trials and more accurate linear approximations will increase the success of this attack. The remaining key bits are found by exhaustive enumeration.

2.1.2 Differential Cryptanalysis

Differential cryptanalysis exploits relationships that exist between differences in the input and output of a function block [2]. In the case of an encryption algorithm, plaintext patterns with fixed differences are examined. The goal is to discover "characteristics". Characteristics are specific differences in pairs of plaintext patterns that, for a given key, have a high probability of causing specific differences in the ciphertext pairs. A differential attack would consist of applying pairs of plaintext with fixed differences, observing the differences in the ciphertext pairs and assigning probabilities to different candidate subkeys. The probabilities will be based on the cryptanalyst's knowledge of the algorithm's characteristics.

Enough trials are run such that the correct key can be determined.

2.1.3 The Boomerang Attack

The boomerang attack devised by Wagner [3] can be seen as an upgrade of classical differential cryptanalysis operating on quadruples of data instead of pairs with fixed difference. Quadruples (or quartets) of plaintexts are properly chosen, and observed together with corresponding quadruples of ciphertexts and intermediate states. Wagner showed how to apply this attack to some of the lesser known block ciphers. In 2005, Biryukov [4] claimed that boomerang attacks on 5 and 6 rounds of AES are much faster than the exhaustive key search and twice as fast as original Square attack by the designers of the AES. We could not find any more recent work on boomerang attacks on AES, until recent 2009 related-key attacks in [49] summarized in section 3.

2.1.4 Truncated Differentials, the Square Attack and Interpolation Attacks

Truncated differentials are a generalization of differential cryptanalysis where partially determined differentials are considered [5]. These partial differentials often cluster into pools of difference pairs. This property can yield statistics that significantly reduce the complexity for a successful attack.

The Square attack is a generalization of an attack originally proposed against the Square Block Cipher [6]. For this attack, a "multiset" of plaintexts is carefully chosen to have certain properties. This multiset is applied to the algorithm and the propagation of these multisets is then examined through the various rounds. The persistence of these properties gives insight to the statistical behavior of the algorithm which can be used to reveal bits of key.

For interpolation attacks, the cipher is modeled using a high-order polynomial [7]. Then the polynomial is solved for the key-dependent coefficients. The technique is very effective when a compact expression of low degree describing the cipher is possible.

2.1.5 Security Summary

The tenets of differential cryptanalysis, linear cryptanalysis, truncated differentials, the Square attack and interpolation attacks matured prior to the design of AES. In [8], the authors of AES establish the conditions that for a cipher to be secure against differential cryptanalysis that there are no differential trails with a predicted propagation ratio higher than 2^{1-n} and to be secure against linear cryptanalysis there are no linear trails with a correlation coefficient higher than $2^{n/2}$. They then proceed to show that AES meets these conditions with 8 rounds or greater and is, therefore, provably secure against both of these techniques. Further, AES is secure against truncated differentials with 6 rounds or more, is secure against the Square attack for 7 rounds or more and is secure, by design, against interpolation attacks.

2.2 Algebraic Attacks

Algebraic attacks were first introduced in 2002 in [9]. For these attacks, AES is expressed as a system of multivariate polynomial equations over a single Galois field. Efficiently solving this system of equations to recover the key variable is the objective of the attack. A very attractive feature of most algebraic attacks is that they require only a single, or a very small number of plaintext/ciphertext pairs, where encryption used the unknown key. This is in stark contrast to, say, classical linear attacks on DES, which perhaps are computationally manageable, but unfortunately they require a very unrealistic number of such pairs, namely about 2^{40}. On the other hand, the algebraic attack would be dangerous only if the set of equations defined by the cipher and unknown key is realistically solvable for sizes of several thousand variables and equations. There is no convincing evidence that such computations are feasible, while the difficulty of handling much smaller cases is notorious.

2.2.1 XL and XSL

In 1999, Kipnis and Shamir [13] were perhaps the first to attract attention of several researchers to the following general strategy: given a system of multivariate polynomials describing relationships between variables, i/o and keys of some cryptographic function, first try to express it as a single univariate polynomial of a special form over an extension field, and then use it to reduce the original cryptanalytic problem to a system of quadratic equations over the extension field. Such systems might be attacked using relinearization methods which are easier to handle, but require a larger number of variables.

This was extended in 2000 by Courtois, Klimov, Patarin and Shamir [12] to an approach potentially usable in the attacks on AES, which was called the XL (eXtended Linearization) algorithm. It is a method of solving systems of multivariate quadratic equations via linearization. This has been followed by an improvement of the XL algorithm called XSL (eXtended Sparse Linearization) by Courtois and Pieprzyk in 2002 [9]. The authors of XSL aimed at exploiting two properties of large systems of equations obtained from cryptanalysis: the systems are very sparse and they are overdefined. There were several further papers proposing more improvements to these algorithms, but also many papers and theses essentially implying that these attacks, as intended, are unworkable.

2.2.2 Cube Attacks

Cube attacks rely on the ability to determine a low-order polynomial description of the output of the cipher. Then a clever iterative approach is used to solve the expression to find bits of key. This attack is most effective on stream ciphers with an LFSR structure [10]. AES and DES are believed to be immune to the attack primarily because an algebraic polynomial that could describe any good block cipher would be of too high a degree to allow this attack to be any more practical than a brute force search of the key space [11].

2.2.3 Security Summary

In general, AES seems to be overdesigned with respect to linear and differential cryptanalysis. This is not the case with respect to the algebraic attacks. Nobody really knows what is the ultimate answer in this case since a general design of XSL only vaguely refers to specific algorithms which need to be used, and leaves much to decide to the implementer. So, every failed implementation can be deemed as one which did not properly exploit the special structure of available linear and quadratic identities.

In 2006, the CTC cipher (Courtois Toy Cipher) and its upgrade to CTC2 were described by Courtois, and then nicely attacked by the author [17] using simple algebraic method. Together with another paper [16], these seemed to promise much stronger and more dangerous results in attacking block ciphers with algebraic methods. Even more, the author planned to delay the publication of the method in order to minimize potential damage from fast unexpected attacks. These anticipations did not materialize; on the contrary, a general feeling of impracticality of algebraic attacks was building up. Recent work by Dunkelman and Keller [18] seems to indicate some success of algebraic attacks on some versions of CTC. It points to special cipher design features, not the generic power of such attacks. AES is very structured and algebraically elegant, thus, it is tempting to envision that algebraic methods should be especially effective against it. Still, one must remain skeptical of how realistic it is to expect any significant further progress of such methods any time soon.

2.3 SAT Solver and Hybrid Attacks

A block cipher such as DES or AES can be expressed as a very complicated Boolean expression involving a number of variables. These variables are the plaintext input bits, the key input bits, and the ciphertext output bits. The Boolean expression is constructed to be true if and only if the ciphertext bits are equal to the encryption of the plaintext bits using the key bits. One way to attack a block cipher is to set the plaintext and ciphertext variables in the Boolean expression to the values corresponding to a known plaintext-ciphertext pair, and then to find values for the key variables that make the Boolean expression true. This is an example of the *Boolean satisfiability (SAT)* problem. A computer program that automatically finds the solution to a SAT problem is known as a *SAT solver*; zChaff, MiniSat, and SAT4J are examples of modern open-source SAT solvers. Rather than trying every combination of values for the unknown variables (a brute force search), a SAT solver tentatively assigns values to the variables one at a time until a conflict is encountered and the Boolean expression becomes false. The SAT solver then backtracks and assigns different values to the variables to avoid the conflict. This conflict-driven backtracking can potentially eliminate large portions of the search space, allowing the SAT solver to find the solution in less time than a brute force search.

Some studies attempted to attack DES using the just-described approach. Massaci [19] and Massaci and Marraro [20] were able to find the key for 2-round and 3-round DES, respectively. While SAT solvers are theoretically capable of recovering the key for any number of rounds, all the way up to the full 16 rounds of DES, the SAT solvers take too long to find the solution. The reason seems to be that, for the full number of rounds, the Boolean formula for DES does not start to experience conflicts until almost all of the unknown key variables are assigned values. Thus, the conflict-driven backtracking does not eliminate very much of the search space, and the SAT solver takes not much less time than a brute force search. Although we are not aware of any studies that simply dumped the Boolean expression for AES into a SAT solver, it seems unlikely that AES can be effectively attacked this way.

A more effective approach is to combine a SAT solver with another technique for a *hybrid attack*. Potlapally et al. [21] reported a combined side-channel and SAT-solver attack on DES, 3DES, and AES. They showed that if a side-channel attack can provide values for the input and output bits of any one of the ten rounds of AES, a SAT solver can then find the full 128-bit key. However, they did not actually carry out the side-channel attack, nor did they assess the difficulty of finding all the inputs and outputs of a round using side-channel techniques, so whether this hybrid attack would work in practice is still unknown.

Courtois and Bard [22] reported another hybrid attack on DES, combining algebraic techniques with a SAT solver. They expressed the DES S-boxes as large, sparse, nearly-linear systems of equations in GF(2). ("Nearly-linear" means each equation had at most one nonlinear term.) These were extended to form equations describing the whole cipher for some number of rounds. The equations were converted to a Boolean expression. A SAT solver was then used to find a subset of 36 key bits, the remaining 20 key bits being fixed. (Alternatively, the key bits not found by the SAT solver could have been found by brute force search.) Using this approach, they were able to find the key for DES reduced to 6 rounds.

Although we are not aware of any studies that tried the algebraic/SAT-solver attack on AES, it is unlikely that such an attack can break AES at the present state of the art. However, this approach is worth watching. While algebraic techniques by themselves cannot break AES now, algebraic techniques will continue to improve; SAT solver programs will also continue to improve; and the combination may eventually pose a threat to AES. Furthermore, as Courtois and Bard point out, the hybrid algebraic/SAT-solver attack is able to find the key from *just one* known plaintext-ciphertext pair. In contrast, linear and differential cryptanalysis require exponentially many plaintext-ciphertext pairs. Thus, the hybrid algebraic/SAT-solver attack is much more likely to become practical, since in the real world it is difficult or impossible for an adversary to collect enough plaintext/ciphertext pairs to mount a linear or differential attack.

2.4 Side Channel Attacks

A side-channel attack exploits information leaked from a cryptosystem due to vulnerabilities in its physical

implementation rather than any cryptographic vulnerabilities of the algorithm. Information gained from observable parameters such as variations in timing, power consumption, electromagnetic radiation, thermal emanations or acoustic emanations can sometimes be used to determine sensitive data, such as bits of plaintext or a key variable.

Some examples of these methods are: timing attacks, differential power analysis attacks, simple power analysis attacks and fault injection based attacks. Timing analysis exploits relationships between the run-time of functions within a cryptographic device and sensitive data elements that are being processed. Changes in execution times of these functions are used together with a model of the system to determine bits of sensitive data. Although they are sometimes limited by the need for precise measurements, timing attacks can be particularly powerful because they are non-invasive and can be applied remotely [23]. Differential Power Analysis (DPA) enables the security of cryptographic devices to be compromised by analyzing their power consumption. Simple Power Analysis (SPA) is a simpler form of the attack that does not require statistical analysis [24] [25]. Fault injection based attacks exploit computational errors to find cryptographic keys [26] [27]. Computational errors are introduced into a cryptographic device by exposing the device to some physical effect such as electromagnetic radiation, excessive temperature or by applying inputs that exceed the device's specifications (clock rate, input levels, input timing, etc.). Miscomputed results, together with a fault model, are used to extract secret data. Some other examples of side-channel attacks include acoustic attacks and electromagnetic emanation analysis [28] [29].

2.4.1 Timing Attacks

Research on timing attacks is making steady progress against implementations of the AES algorithm. It is important to understand these methods, as countermeasures are often straightforward to implement during the design phase. The most promising developments in timing attacks on software implementations of AES focus on "micro-architectural" features of the hosting platform. Cache-based attacks take advantage of the correlation between the secret key and the cache usage. This is performed either through direct timing analysis in the case of single threaded implementations [30] or through analysis via a parasitic co-resident process on multi-process platforms. A "cache-collision" attack is reported in [31] which claims to be able to recover a full key with only 213 timing samples. A "cache-usage" attack using an unprivileged process is described in [32] which retrieves 45.7 bits of key using only about 1 minute of timing data. Another method exploits timing dependencies between the branch prediction capability common to all high performance micros and bits of the secret key [33], [34]. Compromise of the RSA algorithm was successfully demonstrated using this attack and generalization to symmetric ciphers is recommended for future work in [35]. These attacks are also applicable to hardware implementations that utilize these same processing elements.

2.4.2 Power Analysis

Military encryption systems usually employ physical intrusion protection mechanisms. One might conclude that this would make them secure against power analysis attacks. However, poorly designed equipment may allow other parameters that correlate with current draw to be monitored remotely (e.g. electromagnetic leakage or transmission envelope power). An attacker could also gain access to the power consumption profile of a target machine by inserting a monitoring device covertly during the design phase or later in an unprotected area of the equipment (e.g. within the battery pack). Power analysis attacks take advantage of many of the same vulnerabilities with AES implementations as timing attacks. Power consumption profiles can reveal secret key information leaked by micro-architectural mechanisms such as cache usage. A "cache-trace attack" on the final round of AES is reported in [36]. Here, optimized constraint methods are used to recover the full key using power traces of between 5 to 50 encryptions. A less complex cache attack is presented in [37] that requires 256 traces to derive the entire key. A version of this attack can derive a key from architectures that are hardened to DPA using 480 traces. In [38], Boracchi and Breveglieri investigate the application of DPA against hardware implementations of the AES S-Boxes. This work indicates that even hardened hardware implementations of AES may be vulnerable to DPA. Another interesting research course combines analytical methods with power analysis techniques. In [39], SPA is used to detect cryptographic collisions. The attack offers the advantages of avoiding a complex statistical approach and requiring a very small number of samples. The authors report a chosen-plaintext attack which allows a 128-bit key to be derived with only 40 power measurements.

2.4.3 Fault Injection Analysis

The next major area of side-channel attack research on AES implementations is fault injection analysis [26]. Although AES has proven to be sensitive to fault analysis, an attacker must be in physical possession of the cryptosystem to execute this attack and may even need access to the actual encrypting device [40]. In addition, the attack requires use of a "fault model" of the device and a means to reliably induce faults without permanently damaging the unit under attack. The fault model must be available before an attack is planned and can require detailed knowledge of the design of the system. Even though fault injection analysis doesn't currently represent a practical threat to military tactical communications applications, research in this area is brisk and practical applications have already emerged outside the tactical environment. In [41], predictable fault injection is demonstrated by under-powering an AES-base smart card to induce setup time violations. This work showed that faults can be induced reliably in accordance with an AES fault model and, more importantly, without permanently damaging the unit under attack. In [42], a practical application of the concepts presented in [26] and [44] is presented which allowed fault

injection analysis to retrieve a full AES-128 key by analyzing less than 50 ciphertexts.

2.4.4 Countermeasures

A number of countermeasures to the timing attacks are summarized in [32]. For AES, the timing of memory accesses to look-up tables is strongly correlated with secret key data. A number of implementation recommendations seek to reduce or eliminate this correlation. If possible, the embedder should avoid look-up tables altogether and use the logical implementations of AES instead. Alternatively, look-up tables can be stored in registers to eliminate memory accesses and associated timing. AES implementations using a smaller set or multiple copies of tables are also available which changes the access statistics, making timing more difficult to predict.
Other recommendations are made to "obfuscate" memory access timing as follows:

- Implement memory accesses by reading all entries of the relevant table, in fixed order, and use just the one needed.
- Read one representative element from each memory block.
- Shuffle memory content whenever it is accessed or occasionally permuting the memory and keeping the cache locked between permutations.
- Add noise to the memory access pattern by adding spurious accesses.
- Hide timing be adding random latency or by adding delays to normalize access time.
- Disable interrupts and simultaneous threads during memory access to prevent leakage.
- Disable cache capability.

Algorithm masking techniques (see Figure 1) are also available. Here, the AES algorithm is modified in a way that allows a "mask" to be mixed with the plaintext data prior to encryption/decryption and removed afterwards to yield the correct result. This method can remove the correlation between timing measurements and sensitive data. Masks can be random, calculated or fixed value. Mixing can be additive, multiplicative or both.

All of these methods will impact performance even in hardware implementations. In high performance applications, impact can be minimized by applying countermeasures only to rounds that can be targeted for attack.

Lastly, Intel has announced plans to include 7 new AES instructions in their new Westmere™ family of processors. These will calculate the AES encryption round function, decryption round function, and key expansion function in constant time hardware implementations. "Since the instructions run in data-independent time and do not use lookup tables, they help in eliminating the major timing and cache-based attacks that threaten table-based software implementations of AES." [48]

Power analysis countermeasures attempt to eliminate the correlation between power fluctuations and sensitive data. DPA is a powerful scheme and it is difficult to imagine a method that will provide perfect assurance especially with software implementations. Proposals such as adding noise generating circuits to the cryptographic device may seem to be a successful countermeasure, but in practice, DPA easily overcomes the technique with just a few more captured power traces.

Some advances have been made in the development of specialized ASIC standard cell libraries which do not exhibit data-dependent power consumption [46]. Another area of research is in the application of hardware-based masking schemes (Figure 1). The promise of this approach is that a properly designed technique could simultaneously protect against DPA and timing attacks. In [45], a successful random masking technique is applied to an AES hardware implementation with a performance penalty of only 40-50%.

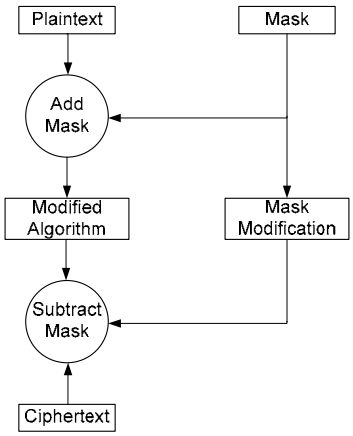

Figure 1: Generalized Masking Mechanism

Many countermeasures to fault injection analysis exist. All attempt to increase a cryptographic device's security by making error detection rates low or not determinable at all. The most successful techniques employ some sort of error detection mechanism. The intent is for the cryptographic device to deactivate itself after a certain number of detected faults, thus preventing an adversary from collecting enough information to mount a meaningful attack. Research focuses on the development of these error detection mechanisms as a tradeoff between level of assurance and impact to performance. For example, [43] and [47] focus on the application of error detection codes to the linear and non-linear algorithm elements so that as much coverage as possible is afforded. These methods offer a high level of assurance but inevitably impact performance significantly. In [44] the authors propose adding redundancy to the implementation and using a simple comparison to detect faults. The premise is that practical fault injection will not be able to identically affect both redundant elements and that the differences will be detectable at the function's outputs through comparison. This method is very effective. Hardware implementations can be rendered with little impact to performance at the expense of additional hardware. Other implementation guidance includes recommendations to prevent an adversary from bypassing the error detection functionality.

2.5 Related-Key and Distinguishing Attacks

A related-key attack on a block cipher is a variant of a chosen-plaintext differential attack. The attacker chooses multiple pairs of plaintexts, where the difference between the plaintexts in each pair is specified. Using the cipher as a black box oracle, the attacker encrypts each plaintext with two keys, where the difference between the keys is specified (but the keys themselves are unknown); these are the "related" keys for which the attack is named. From the information derived, the attacker recovers the unknown keys. Although related keys are unlikely when a block cipher is used for encryption, related keys are common when a block cipher is used as part of a cryptographic hash function. A successful related-key attack may then break the hash function.

In 2009, Biryukov et al. [49] published related-key attacks on *full-strength* AES-192 and AES-256. The attacks recover the key with 2^{176} work for AES-192 and 2^{119} work for AES-256. Since these attacks take less time than brute force, AES-192 and AES-256 are theoretically broken; but the attacks take too long to be practical. However, Biryukov et al. [50] also published related-key attacks on *reduced-round* variants of AES-256 that *are* practical -- 2^{39} work for 9-round AES-256, 2^{45} work for 10-round AES-256. Ironically, these attacks do not succeed for AES-128, which with its shorter key is supposedly weaker than AES-192 and AES-256.

A distinguishing attack allows the attacker to detect nonrandomness in the block cipher technically; the attacker can distinguish the block cipher's behavior from that of an ideal random cipher. Since the security of cryptographic constructions, notably hash functions, built from block ciphers is typically proven assuming the block cipher is an ideal random cipher, a distinguishing attack on the block cipher calls into question the security of the construction.

Biryukov et al. [51], [52] have published a related-key distinguishing attack on AES-256 requiring 2^{120} time. They parlayed the distinguishing attack into a key recovery attack requiring 2^{65} memory and 2^{131} time. Like their previous attacks, this attack theoretically breaks full-strength AES-256 but is not practical. Gilbert and Peyrin [53] have published a known-key distinguishing attack on AES-128 reduced from 10 rounds to 8 rounds; the attack requires 2^{32} memory and 2^{48} time. This attack is practical and breaks a nearly-full-strength variant of AES.

3.0 SUMMARY OF SECURITY IN TACTICAL MILITARY APPLICATIONS

For encryption algorithms used in the military/government domain, any cryptanalysis progress is cause for concern, especially when breakthroughs appear at near real-time in public literature. At issue is the level of sophistication of the adversary. In the military/government threat model, the adversary is a national intelligence agency. These agencies have access to world-class expertise, funding and resources. Targeted information is any information that provides a military, political or diplomatic advantage over an opponent. The value of this information cannot be measured in monetary terms so classic cost tradeoffs cannot be applied. It must be assumed that no expense will be spared by the adversary in an attempt to compromise the security of the target system [54].

Military/government encryption solutions must be secure against all known cryptanalysis techniques. AES was designed to be secure against differential and linear cryptanalysis and their variants. Therefore, any threat from these attacks is minimal. Despite impressive initial results, algebraic attacks have not made enough progress to be practical. Hybrid algebraic/SAT solver attacks might yield results, but these have not yet been extensively studied. A breakthrough is doubtful, but caution is still advised. AES is vulnerable to a "related key" attack when used in a hash function structure and is not recommended for these applications.

Side channel attacks represent a very real threat in the military/government communications domain. Research on side-channels attacks of AES embedments has made enough progress to warrant serious consideration by implementers. Software implementations of AES running on general purpose processors appear to be inappropriate for most military communications applications. Although countermeasures have been proposed which are able to reduce the threat, it is doubtful that code for a general purpose processor could ever be designed for constant execution and uniformly distributed power consumption [21]. Wherever possible, a hardware implementation using constant execution/uniform power functions is recommended. The system designer must take care to control the extraneous leakage of information in the physical implementation of not only the encryption solution but throughout the equipment. For fielded systems, physical access to the equipment and its peripherals (batteries, headsets, etc.) should be controlled. Any of these could be used as a clandestine entry point by the adversary for monitoring a range of parameters.

AES is now over ten years old (the Rijndael cipher, which became AES, was published in 1999). During that time significant cryptanalysis of AES has accumulated, although without breaking AES. The next ten years of cryptanalysis will probably not break AES, but may weaken AES's security enough that a new standard block cipher will have to be developed. (Note that while the SHA-1 and SHA-2 hash functions have not been broken either, enough hash function cryptanalysis has been published that NIST decided to develop a new SHA-3 hash function anyway.) Look for a new AES-2 block cipher development effort to begin no later than 2020.

If a breakthrough appears in the literature while the AES is still in service, the impact of the specific method needs to be examined. For example, it still may not represent a practical vulnerability to tactical applications where the value of information is short. The period of vulnerability will be the time between the publication of a practical breakthrough and the completion of a replacement effort. Interim solutions such as enhanced round or a multiple encryption versions of AES can also be considered. Besides identifying a suitable

replacement, a major challenge is one of logistics. The only risk mitigation for either of these is to plan ahead as if a breakthrough is certain.

4.0 CONCLUSION

This paper presented the results of a study on the current progress of cryptanalysis research on the Advanced Encryption Standard (AES). The objective was to specifically identify threats and vulnerability trends in secure military communication applications. A military threat model represents a much more severe exposure to a much more capable adversary than for any commercial application.

It was determined that cryptanalysis research is making progress against AES. Further, caution is recommended because that progress is happening in the public domain. Results show that AES is currently vulnerable to various side channel attacks. However, appropriate countermeasures are available which, when properly implemented, can eliminate these vulnerabilities at the equipment level. Other methods such as algebraic attacks, hybrid attacks, etc., are making steady progress, but no breakthroughs have been reported. With these, the trends indicate that AES won't have the life expectancy of the traditional algorithm suite approved for classified applications. This makes AES an inappropriate option for classified strategic applications. However, modern secure tactical communications equipment employs programmable cryptography. In the event of a public domain breakthrough, a new algorithm could be fielded relatively quickly. The period of vulnerability will be more defined by practical logistic issues rather than technical issues. Advance planning is required to prepare for this inevitable event.

5.0 REFERENCES

[1] M. Matsui, "Linear Cryptanalysis Method for DES Cipher", EUROCRYPT, LNCS 765, pp.386-397, Springer, 1994.

[2] I. Ben-Aroya, E. Biham, "Differential Cryptanalysis of Lucifer", CRYPTO, Journal of Cryptology, pp.187-199, Springer, 1994.

[3] D. Wagner, "The Boomerang Attack, Fast Software Encryption", 6th International Workshop on Fast Software Encryption, LNCS 1636, Springer, 1999.

[4] A. Biryukov, "The Boomerang Attack on 5 and 6-Round Reduced AES", LNCS 3373, pp.11-15, Springer, 2005.

[5] L. Knudsen, "Truncated and Higher Order Differentials", 2nd International Workshop on Fast Software Encryption, LNCS 1008, pp.196–211, Springer, 1994.

[6] J. Daemen, L. Knudsen, V. Rijmen, "The Block Cipher Square", 4th International Workshop on Fast Software Encryption, LNCS 1267, pp. 149–165, Springer, 1997.

[7] T. Jakobsen, L. Knudsen "The Interpolation Attack on Block Ciphers", 4th International Workshop on Fast Software Encryption, LNCS 1267, pp.28–40, Springer, 1997.

[8] J. Daemen, V. Rijmen, "AES Proposal: Rijndael, Version 2", http://www.esat.kuleuven..ac.be/vijmen/rijndael, 1999.

[9] N. Courtois, J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations", ASIACRYPT, LNCS 2501, pp.267-287, Springer, 2002.

[10] I. Dinur, A. Shamir, "Cube Attacks on Tweakable Black Box Polynomials", EUROCRYPT, LNCS 5479, pp. 278-299, Springer, 2009.

[11] B. Schneier, "Adi Shamir's Cube Attacks". http://www.schneier.com/blog/archives/2008/08/adi_shamirs_cub.html, August 19, 2008.

[12] N. Courtois, A. Klimov, J. Patarin, A. Shamir, "Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations", EUROCRYPT, LNCS 1807, pp.392-407, Springer, 2000.

[13] A. Kipnis, A. Shamir, "Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization", CRYPTO, LNCS 1666, pp.19-30, Springer, 1999.

[14] E. Filiol, "Plaintext-dependent Repetition Codes Cryptanalysis of Block Ciphers - The AES Case", http://eprint.iacr.org/2003/003, 2003.

[15] N. Courtois, R. Johnson, P. Junod, T. Pornin, M. Scott, "Did Filiol Break AES?", http://eprint.iacr.org/2003/022, 2003.

[16] N. Courtois, "How Fast can be Algebraic Attacks on Block Ciphers?", http://eprint.iacr.org/2006/168, 2006.

[17] N. Courtois, "CTC2 and Fast Algebraic Attacks on Block Ciphers Revisited", http://eprint.iacr.org/2007/152, 2007.

[18] O. Dunkelman, N. Keller, "Cryptanalysis of CTC2", CT-RSA, LNCS 5473, pp.226-239, Springer, 2009.

[19] F. Massacci, "Using Walk-SAT and Rel-SAT for Cryptographic Key Search", International Joint Conference on Artificial Intelligence, pp.290-295, Kaufmann, 1999.

[20] F. Massacci, L. Marraro, "Logical Cryptanalysis as a SAT-Problem: the Encoding of the Data Encryption Standard", Journal of Automated Reasoning, 24, pp.165-203, 2000.

[21] N. Potlapally, A. Raghunathan, S. Ravi, N. Jha,, R. Lee, "Aiding Side-Channel Attacks on Cryptographic Software with Satisfiability-Based Analysis", IEEE Transactions on VLSI Systems, 15(4), pp.465-470, April 2007.

[22] N. Courtois, G. Bard, "Algebraic Cryptanalysis of the Data Encryption Standard", IMA Int. Conf. Proceedings, LNCS 4887, pp.152-169, Springer, 2007.

[23] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems", CRYPTO, LNCS 1109, pp.104-113, Springer, 1996.

[24] P. Kocher, J. Jaffe, B. Jun, "Introduction to Differential Power Analysis and Related Attacks", Tech. Rep., Cryptography Research Inc, 1998.

[25] P. Kocher, J. Jaffe, B. Jun, "Differential Power Analysis", CRYPTO, LNCS 1666, pp.388-397, Springer, 1999.

[26] D. Boneh, R. A. DeMillo, R. J. Lipton, "On the Importance of Checking Computations", EUROCRYPT, LNCS 1233, pp.37-51, Springer, 1997.

[27] E. Biham, A.Shamir, "Differential Fault Analysis of Secret Key Cryptosystems", CS 0910, CRYPTO, LNCS 1294, pp. 513 – 525, Springer, 1997.

[28] D. Asonov, R. Agrawal, "Keyboard Acoustic Emanations", IEEE Symposium on Security and Privacy, Oakland, CA, pp.3-11, 2004.

[29] J.J. Quisquater, D. Samyde, "ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards", Proceedings of the International Conference on Research in Smart Cards: Smart Card Programming and Security, pp.200–210, 2001.

[30] D. J. Bernstein, "Cache-Timing Attacks on AES", http://cr.yp.to/antiforgery/cachetiming-20050414.pdf, April 2005.

[31] J Bonneau, "Cache-Collision Timing Attacks Against AES", Workshop on Cryptographic Hardware and Embedded Systems, Yokohama, Japan, Oct. 2006.

[32] D. Osvik, A. Shamir, E. Tromer, "Cache Attacks and Countermeasures: the Case of AES", CT-RSA, LNCS 3860, pp.1-20, Springer, 2006.

[33] O. Acicmez, S. Gueron, J. Seifert, "New Branch Prediction Vulnerabilities in OpenSSL and Necessary Software Countermeasures", IMA Int. Conf. Proceedings, pp.185-203, 2007.

[34] O. Acicmez, C. K. Koc, J. Seifert, "On the Power of Simple Branch Prediction Analysis", ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, Singapore, pp.312-320, 2007.

[35] O. Acicmez, C. Koc, J. Seifert, "Predicting Secret Keys via Branch Prediction", CT-RSA, LNCS 4377, pp.225-242, Springer, 2007.

[36] J. Bonneau, "Robust Final-Round Cache-Trace Attacks Against AES", IACR Cryptology ePrint Archive, Report # 374, 2006.

[37] J. Fournier, M. Tunstall, "Cache Based Power Analysis Attacks on AES", LNCS 4058, pp.17-28, Springer, 2006.

[38] G. Boracchi, L. Breveglieri, "A Study on the Efficiency of Differential Power Analysis on AES S-Box", Technical Report 2007-17, DEI Politecnico di Milano, 2007.

[39] K. Schramm, G. Leander, P. Felke, C. Paar, "A Collision-Attack on AES Combining Side Channel and Differential-Attack", Cryptographic Hardware and Embedded Systems - CHES 2004, 6th International Workshop, Cambridge, MA, USA, 2004.

[40] O. Faurax, T. Muntean, "Security Analysis and Fault Injection Experiment on AES", Proceedings of SAR-SSI 2007, 2007.

[41] N. Selmane, S. Guilley, J-L Danger, "Practical Setup Time Violation Attacks on AES", Dependable Computing Conference, pp.91-96, 2008.

[42] P. Dusart, G. Letourneux, O. Vivolo, "Differential Fault Analysis on AES", LNCS 2846, pp.293-306, Springer, 2003.

[43] M. Medwed, "A Continuous Fault Countermeasure for AES Providing a Constant Error Detection Rate", Cryptology ePrint Archive, Report 2009/119, http://eprint.iacr.org, 2009.

[44] M. Joye, P. Manet, J. Rigaud, "Strengthening Hardware AES Implementations Against Fault Attacks", IET Info Security 1(3), pp.106–110, 2007.

[45] N. Pramstaller, F. Gurkaynak, S. Haene, H. Kaeslin, N. Felber, W. Fichtner, "Towards an AES Crypto-Chip Resistant to Differential Power Analysis", 30th European Solid-State Circuits Conference - ESSCIRC, Leuven, Belgium, 2004.

[46] S. Mangard, "Hardware Countermeasures Against DPA, A Statistical Analysis of Their Effectiveness", CT-RSA, San Francisco, USA, 2004.

[47] M. Karpovsky, K. Kulikowski, A. Taubin, "Differential Fault Analysis Attack Resistant Architectures for the Advanced Encryption Standard", Sixth International Conference on Smart Card Research and Advanced Applications (CARDIS '04), Toulouse, France, Kluwer Academic Publishers, pp.177–192, 2004.

[48] S. Gueron, "Intel®'s Advanced Encryption Standard (AES) Instructions Set", Intel Corporation, White Paper, http://software.intel.com/en-us/articles/advanced-encryption-standard-aes-instructions-set, 2009.

[49] A. Biryukov, D. Khovratovich, "Related-key Cryptanalysis of the Full AES-192 and AES-256", ASIACRYPT, LNCS 5912, pp.1-18, Springer, 2009.

[50] A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, A. Shamir, "Key Recovery Attacks of Practical Complexity on AES Variants with up to 10 Rounds", EUROCRYPT, Springer, 2010. https://www.cryptolux.org/mediawiki/uploads/3/38/Fast_attack_on_reduced_AES-256.pdf, 2009.

[51] A. Biryukov, D. Khovratovich, I. Nikolić, "Distinguisher and Related-Key Attack on the Full AES-256", CRYPTO, LNCS 5677, pp.231-249, Springer, 2009.

[52] A. Biryukov, D. Khovratovich, I. Nikolić, "Examples of Differential Multicollisions for 13 and 14 Rounds of AES-256", https://www.cryptolux.org/mediawiki/uploads/f/f2/AES-256_nonrandomness_examples.pdf, 2009.

[53] H. Gilbert, T. Peyrin, "Super-Sbox Cryptanalysis, Improved Attacks for AES-like Permutations", Cryptology ePrint Archive Report 2009/531, November 2, 2009. http://eprint.iacr.org/2009/531.pdf.

[54] M. Kurdziel, J. Fitton, "Baseline Requirements for Government & Military Encryption Algorithms", Proc. IEEE, Mil. Comm. Conf., pp. 1491 – 1497, 2002.