

Using the Collision Attack for Breaking Cryptographic Modes

Albert Carlson*, Indira Dutta[†], and Bhaskar Ghosh[‡].

*Computer Science Department, Austin Community College, Austin, TX, USA.

[†]Dept. of Computer and Information Sci., Arkansas Tech University, Russellville, AR, USA.

[‡]Dept. of Computer and Information Sci., Arkansas Tech University, Russellville, AR, USA.

Email: *albert.carlson@austincc.edu, [†]idutta@atu.edu, [‡]bghosh@atu.edu.

Abstract—Presently the cipher system that is considered to be the industry gold standard is the Advanced Encryption Standard paired with the Cipher Block Chaining cryptographic mode (AES with CBC). The key space and randomization create such a large problem that even a Brute Force attack is thought to be mathematically infeasible. However, separate research by McGrew and Carlson has resulted in an attack known as either the Improbability or Collision attack that shows that the system can be broken with a reasonable amount of input data. In this paper, we present the attack, along with background mathematics, an example, and the experimental results reached by a team led by Carlson. Extensions to other modes are also presented, with the suggestion that the side channel attack renders modes futile. In fact, applying modes reveals so much information that employing them may even compromise the security of the transmitted message.

Index Terms—Encryption, Advanced Encryption Standard, Set Theory, Cryptography, Cybersecurity, Advanced Encryption Standard, Cryptographic Modes, CBC, AES

I. INTRODUCTION

A. Independent Random Variables

Independent random variables (IRVs) [1] are those variables that have the property that for every time that they are invoked there is the same probability of occurrence. There is no Bayesian dependence on some other outside property or action. Many examples of such a variable exist in nature and in practice. Classic examples of IRVs include the number of heads (or tails) resulting from repeated fair coin tosses and the number resulting from playing roulette with a fair ball and machine, IRVs are characterized and governed by the Birthday paradox [2], [3].

B. Collisions

A “collision” in encryption occurs when two blocks of encrypted data are identical. Those blocks must comprise of entire bytes of the size of the cipher key ($|k|$) and begin at an integer multiple of $|k|$ from the beginning of the message. The encrypted data (plain text) does not have to be identical for the collision. Only the cipher text need be identical [4], [5].

C. Cipher Reduction

Cipher reduction [6], [7] is a technique introduced by Carlson that allows substituting one cipher algorithm for another, given an appropriate key, such that for all input blocks

$$E_{c_1, k_1}(M) = E_{c_2, k_2}(M) \quad (1)$$

where the ciphers (c_1, c_2) and not identical (i.e. $c_1 \neq c_2$). While the keys (k_1, k_2) are typically not identical, they may be, depending on the input data and ciphers. This cipher reduction was based on work and results published by Feistel [8] in the early 1970’s. In this article, Feistel wrote that at their hearts, all ciphers are substitution (S) ciphers. Carlson took this literally and showed that all ciphers can be replaced by an S cipher, or as in the case of a serial cipher, a series of S ciphers. S ciphers have a property known as “idempotence” [9]. This property states that for product S ciphers, those where a message is repeatedly encrypted by S ciphers, that there exists ($\exists$) a single S cipher with a new key that is equivalent to the result of applying that sequence of S ciphers. That is

$$E_{S_1, k_1}(E_{S_2, k_2}(\dots(E_{S_n, k_n}(M))\dots)) = E_{S, k_{equivalent}}(M) \quad (2)$$

or in a less rigorous shorthand

$$S_1 S_2 \dots S_n(M) = S_{equivalent}(M) \quad (3)$$

or

$$S_1 S_2 \dots S_n = S' \quad (4)$$

As a result of idempotence and the ability to map ciphers to S ciphers, this means that it is possible to reduce a product cipher to a single S cipher for cryptanalysis and decryption. Therefore

$$E_{c_{product}, k}(M) = E_{S', k'}(M) \quad (5)$$

Another characteristic, and problem, with S ciphers, is that the patterns in the original text of the message are not changed by substitution. Only the mappings of symbols change. After encryption the same patterns are present. Therefore, if a cipher can be replaced by an equivalent S cipher, this implies that for any such cipher the patterns are still present. This means that idempotence and cipher reduction are viable and applicable for product ciphers. As a result of this property, practitioners respond to the problem by both increasing key size and by adding randomization to make the encryption process safer.

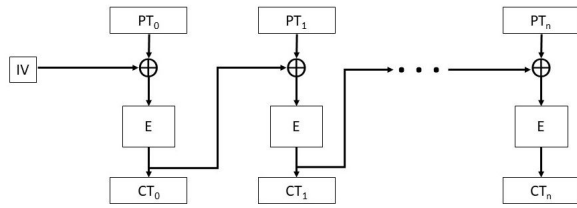


Fig. 1. Structure of Cipher Block Chaining Mode

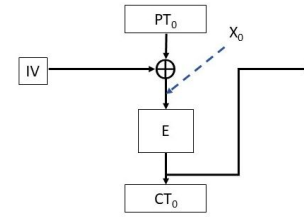


Fig. 2. Location of X in CBC

D. Cryptographic Modes

Randomization of the data is accomplished by the use of cryptographic modes [10]. The role of the mode is to try and change the input into the encryption function to be a different plain text character than is that of the message plain text. Even though the input into the encryption step is different from the message, there is a deterministic and reversible process to map the two blocks of characters. In this way, identical message plain text blocks can result in different cipher text outputs. However, this is only partially correct. The difference in outputs is only due to the use of the preceding block's cipher text creating a new block presented to the encryption function. Several accepted mode algorithms have been presented [10]. These are Electronic Code Book (ECB), Cipher Block Chaining (CBC), Propagating Cipher Block Chaining (PCBC), Output Feedback (OFB), Counter (CTR), XEX-based tweaked-codebook mode with ciphertext stealing (XTS), and format preserving encryptions (FF1 and FF3). These modes are applied on top of the encryption and are thought to be more safe for data transmission. However, studies by McGrew [11] and Carlson [4], [5] have shown that at least some modes are susceptible to side channel attacks on the modes that return enough information for the attacker to recover the original plain text message from the randomization data.

E. Side Channel Attacks

Side channel attacks were first introduced by Kocher in 1996 [12]. They are defined as security exploits that use indirect measurements of a system hardware or software algorithms in order to gain information about information being processed. They are attacks directed against systems, such as cryptography, that focus on information that can impute the key or decrypted text of an encrypted message. Among the forms of side channel data used in attacks are the timing of instructions in a processor [12], power used by the hardware [13] (such as Tempest attacks [14]), electromagnetic analysis [15], and more recently attacking the software algorithms that provide operational data to the encryption algorithm. Each of these attacks attempts to gain insight into the cryptographic process by different means. Of specific interest is CBC mode, whose structure is shown in Figure 1.

The time attack does precise measurements of the time it takes for each processor instruction to execute. Variations of the instruction operations due to inputs and different choices

made in execution due to the input data can be used to work backwards until the plain text inputs are identified.

Power side channel attacks are based on the fact that different instructions use a characteristic combination of gates inside the processor. Manufacturing variations make it possible for these variations to exist and be identifiable. By measuring the power used by known instructions during operations it is possible to use those measurements to infer what instructions and input combinations are being run on the machine. With sufficient characterization of the processor and good measurements, it is possible to reconstruct the exact steps taken in encryption and decrypt the message.

Inferring inputs can also be done to various programs based on their outputs. It is not possible for all programs. Such inferences should never be possible for encryption algorithms where the inference allows for the attacker to read the original input text. Any means for such inference that does not use the internal information of the program is a side channel attack.

F. Motivation for the Work

AES with CBC has long been considered the standard for encryption worldwide. Until the work of McGrew and Carlson, there was no strong attack on the cipher. The attack showed the mathematics of the attack on the randomizing function was effective. The proof of the mathematics was demonstrated by Carlson, but a peer-reviewed report of the experimental results has not been made to this point. This paper discusses the approaches to the attack and the results of implementing the theory.

II. THE COLLISION ATTACK

McGrew [11] and Carlson [4], [5] independently arrived at a side channel attack for AES with CBC. McGrew was the first to publish his algorithm for the attack and called it the "improbability" attack. Carlson followed afterward, calling his version of the attack the "collision" attack, referencing the collision of cipher text that forms the basis of the attack. Both attacks follow the same basic math pattern up to the application of the data applied to find the final plain text output. A major difference between the two implementations is that McGrew focuses on solving the decryption one bit at a time while Carlson et al. solve the problem at the block level. Both approaches concentrate on a point internal to the CBC structure at point X. That point is shown in Figure 2.

The mathematics of the attack depend on collisions of the cipher text and proceeds as follows. If $CT_i = CT_j$ then

$$X_i = X_j \quad (6)$$

then by the 1:1 and onto (bi-junctive) law [16] principal for cryptography

$$X_i = CT_{i-1} \oplus PT_i \quad (7)$$

and

$$X_j = CT_{j-1} \oplus PT_j \quad (8)$$

Therefore

$$CT_{i-1} \oplus PT_i = CT_{j-1} \oplus PT_j \quad (9)$$

and after a bit of rearrangement the relationship becomes

$$CT_{i-1} \oplus CT_{j-1} = PT_i \oplus PT_j \quad (10)$$

Now, assume that a series of comparisons is made with the collisions at various points in the encrypted message that collide. However, the first instance of the collision occurs at i . Each comparison is made with the same i and different collision pairs at the corresponding sequence of locations represented by j . That is, the block i does not change. Each pair of values may differ. Therefore

$$\text{block}_i \in \bigcap_{m=1}^n CT_i \circ CT_j \quad (11)$$

applied until the result is $\exists! \text{block}_i$ mapping from cipher text to plain text.

Once the value of PT_i is known, then it is possible to uniquely and unambiguously resolve the values for all of the other plain text values for every block whose cipher text collides with the known plain text block. It does not matter what the original plain text was, because

$$PT_j = PT_i \oplus CT_{i-1} \oplus CT_{j-1} \quad (12)$$

But this is not the extent of the revealed information related to this particular collision. Because the other cipher text blocks do not collide with this known cipher text (i.e. $CT_i \neq CT_j$), it must be true that

$$PT_j \neq PT_i \oplus CT_{i-1} \oplus CT_{j-1} \quad (13)$$

Thus, for all blocks that do not collide with the known block the value calculated for PT_j can be eliminated from consideration. The list of possible plain text values for those blocks throughout the message is then reduced by one value. However, if the collision occurs on both pairs of cipher text characters at locations i and j , as well as $i-1$ and $j-1$, then the plain text at locations i and j are identical.

An important observation can be made at this point. That observation is that nowhere in any of the equations that govern the information retrieval does a term for the encryption algorithm appear. This means that the attack is encryption agnostic. It can be used with any encryption. The encryption can, and is, treated as if it is a substitution (S) cipher [6], [7], [8]. Therefore, the encryption is not really important in

the decryption. There is so much information contained in the randomization function that the input can be directly imputed from that function.

The math for this algorithm are fairly simple and conceptually easy. Creating the data in the difference database for $CT_i \oplus CT_j$ is similarly easy to calculate. Some effort is required to create the information in the database and to organize it for easy access. While the algorithm is extensible for any size block, the amount of memory and time that is required for this extension can be memory and time intensive. Requirements for memory are exponential in the size of the block being processed. The time to process the data also increases quickly corresponding to the block size. However, linear ciphers can be processed much more quickly than non-linear ciphers. The treatment for all ciphers is identical, making the approach uniform and easy using the results of isomorphic cipher reduction. Further, variations of this attack are applicable to at least two other modes, making it effective on at least three of the known cryptographic modes (specifically CBC, CTR, and PCBC). There are ways to defeat this attack, but the methods needed require the application of new techniques not yet in widespread use, such as applying different sized blocks with irregular key sizes.

This method is meant to be used in conjunction with other statistical based attacks. While the algorithm can be used independently and will still be effective, if used with other property sets, will decrease the amount of time and effort expended in decryption. It will also reduce the memory resource burden on the computing resources. If the message is large enough, sections of the message (shards) can also be evaluated in parallel to increase the effectiveness of the attack. Graphics Processing Units (GPUs) can then be employed to allow for massively parallel treatment of the data.

III. BREAKING CBC

There are two approaches to attacking CBC: predict the bits (as shown by McGrew [11] and directly attacking the entire encrypted block (as shown by Carlson, et al [4], [5]). While both are effective, in this paper the focus will be on the algorithm and methodology to break a block of symbols at a time. The reason for this focus is to demonstrate that there is so much information revealed by the mode, which is a reversible randomization algorithm, that it does not matter what encryption is used with the mode. Just attacking the mode allows an attacker to be able to retrieve the underlying plain text. The general, high level algorithm for running the collision attack is shown in Figure 3.

One of the common criticisms of this algorithm is that the numbers of possible block combinations are so large that collisions will not occur in a timely manner and that this technique is time infeasible [4], [5]. That criticism is based on the size of the sample space and the assumption that the randomization is an IRV. IRVs are mistakenly thought to occur only once in a cycle whose size is $|\text{blocks}|$. This is untrue, although the IRV case can be used to bound the amount of effort, on the average. IRVs are ruled by the Birthday Paradox,

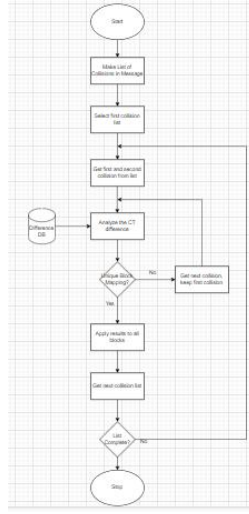


Fig. 3. High Level Algorithm

which paradoxically is not a paradox, but rather a surprise to users. Collisions in data can occur repeatedly and often appear much more often than a user would expect. For example, if there are 26 characters in a block, the point where a collision can be expected (50% chance of collision) is between 6 and 7 characters (46.34% for 6 characters, 58.7% for 7 characters). Assuming that the plain text data is in a natural language, such as English, the probability of collision is even greater. Shannon placed that probability experimentally at approximately every 4 characters ($R_\lambda \approx .75$) [17]. But, in general, the probability for a syntax/rule based language corpus will be less than the bounding IRV case. Since collisions happen far more readily than one would expect, there are generally sufficient numbers of collisions in even relatively small messages (under 10,000 blocks) for this method to be useful. Thus, this criticism is unfounded.

Block ciphers that work on each character in a block, like an XOR cipher, present a special case for encryption collisions. The collision occurs on the byte in the encrypted block as opposed to having to look for the collision on the entire block. Because the character is the bases of the collision, those collisions will occur more often and fewer characters are needed to accomplish the decryption. Using a linear cipher as the encryption can actually decrease the security of encryption paired with the mode.

As an example of the power of this technique, consider the case of AES 8 applied to an English text. The small key size makes it possible to show the operation in this paper, but the technique is extensible to larger key sizes with no loss of generality. The list of sets of possible letters that can be eliminated for the combination of cipher texts preceding the collision event ($CT_{i-1} \oplus CT_{j-1}$) is shown in Table I.

In this example, the list of collisions for the first byte in the encrypted message of 10k bytes begins with the locations 1, 43, 67, 105, 152, 190, 221, 230 ... The first byte is not considered for immediate treatment, since the IV in the CBC

TABLE I
LETTERS REMOVED BY CT PAIRS

$CT_{i-1} \oplus CT_{j-1}$	Letters	$CT_{i-1} \oplus CT_{j-1}$	Letters
1	a,z	17	j,l,m,n,o,q
2	b,y	18	i,l,m,n,o,r
3	c,x	19	h,i,m,n,o,s
4	d,x,y,z	20	h,i,j,k,o,t
5	e,x,y,z	21	h,i,j,k,u
6	f,x,y,z	22	h,i,j,k,m,v
7	g,x,y,z	23	h,i,j,k,l
8	h,s,t,u,v,w	24	c,d,e,f,g,x
9	i,t,u,v,w	25	b,d,e,f,g
10	j,t,u,v,w	26	a,d,e,f,g,z
11	k,t,u,v,w	27	d,e,f,g
12	l,p,q,r,s	28	a,b,c,g
13	m,p,q,r,s	29	a,b,c,f
14	p,q,r,s,u	30	a,b,c,e
15	o,p,q,r,s,t	31	a,b,c,d
16	k,l,m,n,o,p		

function is as yet unknown. $CT_0 = 23_H$ and $CT_1 = 78_H$. The values for CT_{i-1} , in hexadecimal, for the same list of locations is 23, 27, 2D, 2B, 3C, 33, 2F, 32, ...

Starting with location 1, all letters are possible. At this point, no analysis has taken place. The first step in analysis begins by using the first number on the list (1) and the second (43). The XOR of the two preceding cipher test values (the “CT difference”) is 4_H . For that difference, the letters d, x, y, and z can be eliminated. The remaining set of possible values for the plain text at location 1 is $PT_1 \in \{a, b, c, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w\}$. This set becomes the new base set of alphabet possibilities.

The second step in the analysis is to run the same steps for the first number on the list (location 1) and the third (location 67). Notice that one location, the first, is used in both pairs of locations where the cipher text collides. The result of the comparisons is a list of the only pairs of plain text that could have resulted in the collision. Since the XOR operation does not give a clue as to which of the pair is associated with a location (“location insensitivity”). Therefore, the plain text at the first location on the list can be any of the characters in either of the pairs. By keeping one location static in the pairs that character must be in all of the lists. Using the sets of lists possible for each difference, letters are gradually removed from the solution set because they do not appear in all of the pairs. The last character in the solution set must be the mapping for the plain text at the first location on the list. For the second pairing the CT difference is 8 (difference numbers are all hexadecimal), corresponding to the elimination of the letters h, s, t, u, v, and w. Removing these letters drops the number of possibilities for mapping to the first character (or block) to 16, with the solution set constricted to $PT_1 \in \{a, b, c, e, f, g, i, j, k, l, m, n, o, p, q, r\}$.

The next pair considered is locations 1 and 105. In this case the difference is 24 and the letters eliminated are c, d, e, f, g, and x. Of this list, the letters c and d have already been eliminated by an earlier pair. Since the reduction is accomplished

by means of intersection it does not matter if more than one list eliminates the same letter/block. Duplication has no effect on the result. After this application, the result list of possible block mappings is now $PT_1 \in \{a, b, i, j, k, l, m, n, o, p, q, r\}$.

After this pair, the next pair is location 1 and 152. Here the difference is 31 and the deleting the letters a, b, c, and d. The resulting solution list is pared down to $PT_1 \in \{i, j, k, l, m, n, o, p, q, r\}$. Following this step, the next difference is 16 at location 190, which normally removes the blocks k, l, m, n, o, and p. After intersection the new solution set has been modified to be $PT_1 \in \{i, j, q, r\}$. Evaluating location 221 the difference is 12 and the elimination list consists of the blocks l, p, q, r, and s. Of these, only the blocks q and r contribute to making the set smaller. Only $PT_1 \in \{i, j\}$ comprise the solution set. One more collision, that at location 230 is required to solve the problem of plain text for location 1. Having a difference of 17, the list of possible reduction candidates is the list of blocks comprised of j, l, m, n, o, and q. Applying the intersection leaves a solution set with exactly a single block, $PT_1 = \{i\}$.

Once any one of the plain text blocks is known in the list, the rest of the blocks whose cipher text collides with that block can be easily calculated. Give the cipher texts of the list used above (the cipher text outputs at each location are $4A_H$), the plain text can be directly calculated using the formula

$$PT_j = PT_i \oplus CT_{i-1} \oplus CT_{j-1} \quad (14)$$

Since PT_i is now known and the appropriate cipher texts are known, each PT_j is immediately recoverable. The ASCII for the list (starting at location 1) is 69, 6d, 61, 61, 7A, 79, 66, and 78. This corresponds to the letters i, m, a, a, z, y, f, and x, respectively. Note that the mappings are the same for different blocks, in general, although identical plain text blocks also occur.

This data is considerable and important. However, more information can still be extracted. To this point in the analysis, the focus has been on collisions. But when the blocks do not collide the following formula holds

$$PT_j \neq PT_i \oplus CT_{i-1} \oplus CT_{j-1} \quad (15)$$

Therefore, if the cipher text blocks do not collide then the combination of the input data did not meet the equality. The block cannot be that which would have caused the collision and can be eliminated from the possible plain text inputs. In other words, the PT_j calculated by the right hand side of the equation cannot be the correct plain text. To take advantage of this information, the impossible PT_j is calculated and removed from the list of plain text values for that block.

At the completion of these two actions the next list of collisions is analyzed in the same manner. All blocks with sufficient redundancy will be broken [18]. Those without sufficient redundancy are likely to have reduced possible mappings at the end of the application of the attack algorithm. Other heuristic techniques, such as m -gram analysis [18], can

then be applied to recover more information and possibly recover the encrypted plain text.

IV. EXPERIMENTAL RESULTS

Verification of the efficacy of the methodology was done in two parts: testing using AES as the encryption and testing with other ciphers as the encryption.

In the case where the encryption block is other than AES, the same file was used with multiple IVs and passwords that were randomly selected for use. The message was comprised of a file from Project Gutenberg [19] that was a fragment of an accepted work of English literature that was 224,832 bytes in length. A number of basic ciphers were tested. They were the S, P, XOR, Affine, Multiplication, and Vignaire ciphers. These ciphers range from relatively simple to more complex. In order to achieve statistical significance, 3000 tests were run for block sizes that range from 1 - 10 bytes (8 to 1024 bits keys and IVs). Results were categorized by the percentage of recovered data in the file in increments of 10%. Practically, a cipher was considered broken if more than 50% of the text was correctly recovered. Almost universally, this standard was met with 32 bit blocks, although at least 10% of the file was recovered for almost all of the tests up to 512 bit blocks. At the low end of the block scale, 100% recovery was achieved for 8 bit blocks 99.996% of the time. The disparity of results is almost certainly due to the number of blocks available to create collisions. The number of blocks available was approximately 2809 for 8 bit blocks down to 219 blocks for 1024 bit keys.

An interesting effect was observed during data analysis. A comparison of the maximum, minimum, and average amount of data returned from the attack resulted in a performance graph as shown in Figure 5. In these curves, the gray dashed line, or upper line, is the maximum percentage of correctly decrypted plain text in the file for an IV/key pair. On the bottom, the red dotted line represents the opposite - the minimum percentage of correctly returned plain text in a file for any IV/key pair. In between the solid blue line represents the average percentage of correctly decrypted plain text for the file given all of the IV/key pairs. The difference between the maximum and minimum curves represents the reproducibility of the results for different IV/key pairs. The further apart the curves the greater the difference in security for some IV/key pairs using that cipher and key size, since the source plain text does not change. The location of the average in the graph is also significant. If the average is close to the minimum then the weak IV/key pairs are less common, if closer to the maximum, they are more common (see Figure 5). Of the ciphers tested the most consistent were the XOR and Affine ciphers.

AES cipher examples were full texts taken from Project Gutenberg that were limited to keys of up to 24 bits. This limitation was primarily imposed due to the time it takes to process these texts and run the tests. These tests were run on circa 2010 laptops due to their cost and availability. Text size ran from 1 to 2.5 million bytes. A total of 8000 tests were conducted with a success rate of 92%. Another 5000 tests were run on a dedicated laptop. Single byte (8 bit) IV/key

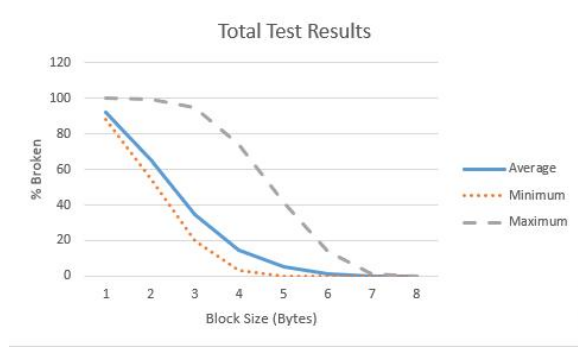


Fig. 4. Non AES Test Results

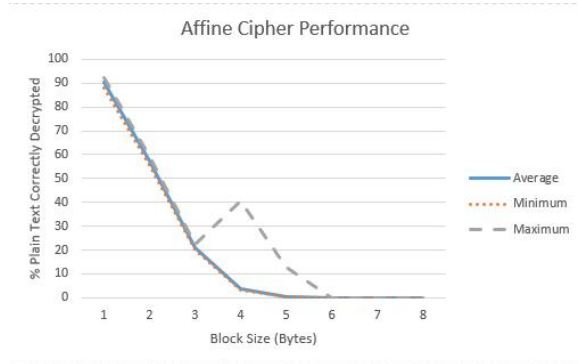


Fig. 5. Affine CipherTest Results

pairs averaged about 12 minutes per file while 3 byte (24 bits) IV/key pairs averaged approximately 72 hours to be completed with a 96% success rate.

V. CONCLUSION AND FUTURE WORK

This paper and study have shown that it is possible to successfully and easily attack the CBC mode used in security. In this algorithm, collisions of the cipher text are key to a side channel attack that is encryption agnostic. Since the architecture of CBC is similar to that of the PCBC and CTR modes it is possible that a similar approach will work on those modes as well. The basic mathematics and results of the attack are also reported.

The Collision Attack algorithm presented here has been proven for small key sizes and is extensible to potentially any size key. The limit for key size rests with the memory capabilities of the computer on which the program is run. While this algorithm has been tested for key sizes of up to 24 bits on AES with CBC and up to 4096 bits for linear ciphers using the same mode, that extensibility must still be tested on larger key size ciphers. Ideally, that testing effort would include 256 bit keys, and possibly be extended to 4096 bits for AES. At the same time, recording and characterizing the time it takes to solve the problem is required in order to understand the complexity and resource used for the attack.

Solving AES with CBC is a major accomplishment, but other modes closely resemble CBC in structure and information content. In particular, the modes closest to CBC are the

Propagating CBC mode (PCBC) [10] and Counter (CTR) [10] modes. PCBC requires coordinating four plain text values, but should not greatly complicate processing. CTR mode presents the same input to combinations to the encryption block as is done in CBC. However, the randomization number used in calculations is regular and the constraints placed on it are stringent enough to make analysis relatively easy. The remaining modes, such as Output Feedback Mode (OFB) [10] and the related feedback modes share characteristics that indicate an algorithm successful in breaking one of them can be extended and quickly modified to be effective on the rest of the similarly structured modes.

Breaking just the PCBC and CTR modes is accomplished and this will show that half of the NIST approved cryptographic modes are susceptible to side channel attacks. This would be a strong indication that modes are not as safe as has been suspected to this point in time. If one of the feedback modes could also be broken it would give further evidence that the concept of modes may be outdated and does not enhance the security of an encrypted message.

Linear ciphers are susceptible to an even more effective attack. This is because the collisions that occur are applicable to individual bits in the block rather than requiring a collision of the entire block. Thus, data can be collected more rapidly than in the case of a full block collision. This implies that the use of a linear cipher as the encryption algorithm protected by the mode may even be weakened by using the mode. More testing is required in this effort needs to be investigated to verify the assertion.

A limitation of this attack is that the attack is only applicable to randomization routines built on the XOR of the present and preceding data. It also assumes that the same cipher/key pair is used throughout the message. Polymorphic encryptions cannot be solved with this attack because the 1:1 assumption does not hold for the entire message. In addition, the method requires a sufficient corpus to contain at least $\lg|A|$ collisions in accordance with the Birthday paradox. The larger the corpus, the better.

Further investigation into the remaining modes recommended by NIST [20] is also required. To date, the FF1 and FF3 modes have not been studied in the manner [21]. The FFX modes are a form of format preserving encryptions (FPEs) that encrypt data into the same format and characters as the original plain text. Each of the currently recommended modes must also be investigated and tested in order to verify the assumption that modes should be deprecated and avoided.

REFERENCES

- [1] Sheldon Ross. *A First Course in Probability*. MacMillan Publishing, Inc, New York, 1976.
- [2] E. H. McKinney. Generalized birthday problem. *The American Mathematical Monthly*, 73(4):385–387, 1966.
- [3] Kyle Siegrist. The birthday problem, <https://www.randomservices.org/random/urn/birthday.html>, 2018.
- [4] Albert Carlson, Patrick Doherty, Isaiah Eichen, and James Gall. Breaking cbc, or randomness never was happiness. Internet Video, August 2015.
- [5] Albert Carlson, Patrick Doherty, Isaiah Eichen, and James Gall. Using collisions to break cbc, 2016.

- [6] Albert Carlson. *Set Theoretic Estimation Applied to the Information Content of Ciphers and Decryption*. PhD thesis, University of Idaho, 2012.
- [7] Bhaskar Ghosh, Indira Dutta, Shivanjali Khare, Albert Carlson, and Michael Totaro. Isomorphic cipher reduction. 2021.
- [8] Horst Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15 – 20, 1973.
- [9] Richard Wells. *Applied Coding and Information Theory*. Prentice Hall, Upper Saddle River, 1999.
- [10] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons Inc., New York, 2nd edition, 1996.
- [11] David McGrew. Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes. In *Proceedings of the Fast Software Encryption Workshop*, 2013.
- [12] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO 1996: Advances in Cryptology CRYPTO 96*, pages 104 – 113, 1996.
- [13] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael Wiener, editor, *Advances in Cryptology — CRYPTO’ 99*, pages 388–397, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- [14] National Security Agency. Tempest: A signal problem the story of the discovery of various compromising radiations from communications and comsec equipment. *Cryptologic Spectrum*, 2(3):26 – 30, 1972.
- [15] Karine Gandolfi, Christophe Mourtet, and Francis Olivier. Electro-magnetic analysis: Concrete results. In *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 251–261. Springer, 2001.
- [16] John B. Fraleigh. *A First Course in Abstract Algebra*. Addison-Wesley, 7th edition, 2003.
- [17] Claude Shannon. Prediction and entropy of printed english. *Bell System Technical Journal*, 30:50 – 64, 1951.
- [18] Claude Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28:656 – 715, 1949.
- [19] The Gutenberg Project. Main page. Internet, 2008.
- [20] M. Dworkin. Nist sp 800-38a, recommendation for block cipher modes of operation. Technical report, NIST.
- [21] Mihir Bellare, Phillip Rogaway, and Terence Spies. The ffx mode of operation for format-preserving encryption, draft 1.1. Technical report, National Institute of Standards and Technology.