# Impossible plaintext cryptanalysis and probable-plaintext collision attacks of 64-bit block cipher modes

David McGrew
mcgrew@cisco.com

Cisco Systems, Inc.

**Abstract.** The block cipher modes of operation that are widely used (CBC, CTR, CFB) are secure up to the birthday bound; that is, if $w2^w$ or fewer bits of data are encrypted with a $w$-bit block cipher. However, the detailed security properties close to this bound are not widely appreciated, despite the fact that 64-bit block ciphers are sometimes used in that domain. This work addresses the issue by analyzing plaintext-recovery attacks that are effective close to that bound. We describe possible-plaintext attacks, which can learn unknown plaintext values that are encrypted with CBC, CFB, or OFB. We also introduce *impossible plaintext* cryptanalysis, which can recover information encrypted with CTR, and can improve attacks against the aforementioned modes as well. These attacks work at the birthday bound, or even slightly below that bound, when the target plaintext values are encrypted under a succession of keys.

## 1 Introduction

A $w$-bit block cipher with a $\kappa$-bit key is an invertible function that maps $w$-bit inputs to $w$-bit outputs, in which the mapping is pseudorandom and different for each of the $2^\kappa$ possible values of the key. Users of cryptography appreciate the need for appropriately large keys, and in practice $\kappa = 128$ is common and is recommended. However, there is less appreciation of the need for large block width $w$, and a decade after the adoption of 128-bit block ciphers, the use of 64-bit block ciphers is still ongoing.

A block cipher encryption mode of operation is an algorithm for encrypting arbitrary-length plaintexts using a block cipher, which logically breaks the plaintext into $w$-bit blocks, and processes these blocks using the block cipher (and other operations such as bitwise exclusive-or).The Cipher Block Chaining (CBC), Ciphertext Feedback (CFB), and Counter (CTR) modes [5] are used in practice with 64-bit block ciphers such as Triple-DES [27, 20], GOST 28147-89 [22], and KASUMI [26].

Theory advises against using a $w$-bit block cipher to encrypt more than $2^{w/2}$ blocks with a single key; this is known as the *birthday bound*. Triple-DES has parameters $\kappa = 168$ and $w = 64$, and GOST 28147-89 has $\kappa = 256$ and $w = 64$. The key sizes of these ciphers imply strong security goals, e.g. that an attacker cannot learn anything about the plaintext without $\mathcal{O}(2^{112})$ or $\mathcal{O}(2^{256})$ operations. However, any 64-bit block cipher in a conventional encryption mode of operation that processes gigabytes of plaintext will give up some secret information; in short, a 64-bit block cipher cannot meet its security goals when it is used at modern data rates. We demonstrate this fact by describing attacks that recover information about unknown plaintexts that take as few as $\mathcal{O}(2^{35})$ operations.

Throughout this work we assume that the attacker can observe all of the ciphertext, and assume that the attacker has some (incomplete) knowledge about the plaintext. For instance, an attacker might know that a particular plaintext consists of english-language HTML4 carried over HTTP/1.1 [8], TCP [24], and IPv4 [23]. Through a careful analysis of the protocols being encrypted, and observations of the lengths and order of packets, an attacker can make many inferences about some parts of the plaintext. The goal of the encryption system is to prevent the attacker from leveraging this partial knowledge to learn other more sensitive parts of the plaintext; for example, passwords like those used for HTTP Basic Authentication [9]. Similar assumptions are valid for the encryption of data at rest, i.e. disks, partitions, filesystems, files, file elements, and database entries. The examples we draw on are from network security, but our analysis applies to other uses of encryption as well.

## 1.1 Previous work

The fact that CBC, CFB, and CTR give away some information about their plaintext after about $2^{w/2}$ blocks are encrypted is well established, though to our knowledge no systematic way to exploit this vulnerability has been described in the open literature. Collision attacks have long been known, but in applied cryptography circles, they are considered more in the context of hash functions, where they first appeared in the literature [19]. Interestingly, there is no discussion of the birthday bound for CBC, CFB, and CTR in the standard that specifies them [5] nor in the Handbook of Applied Cryptography [18].

Bellare et. al. introduced the concrete security analysis of block cipher encryption modes with an investigation of CBC and CTR [1]. This seminal work shows that the security of these modes reduces to the indistinguishability of the block cipher from a random permutation, and that CBC and CTR are secure below the birthday bound; it did not explore how an attacker can exploit those modes when that bound is not respected, or how practical security degrades close to that bound. Other works have shown how cryptanalyts can exploit probabilistic information about plaintexts. Probable plaintext in the Internet protocol family has been studied [2] and it has been shown that some generic key-recovery attacks can be effective on CTR even when the attacker's knowledge of the plaintext is probabilistic [16]. Mason et. al. [15] developed attacks that recover unknown plaintext from misused one-time-pad, based on a statistical model of the plaintext and a dynamic programming approach that identifies the most likely value of the plaintext by finding the highest probability path through a hidden markov model. None of these last three works consider the birthday bound.

In this note, we describe plaintext-recovery attacks against CBC, CFB, and CTR modes that are effective close to the birthday bound. Our attack on CBC and CFB is an application of the well known collision attack, while our impossible plaintext attacks against CTR have not previously appeared. We also show how these attacks can be applied at, or even slightly *below* the birthday bound, by continuing the attack across a succession of distinct keys. We provide motivating examples from communications security to show that the attacks can be effective, discuss the use of 64-bit block ciphers at modern data rates, and conclude that 128-bit block ciphers have significant security advantages.

### 1.2 Notation

We let $w$ denote the number of bits in the plaintexts and ciphertexts of the block cipher. The plaintext and ciphertext consist of a sequence of $w$-bit blocks $P_i$ and $C_i$, respectively, for $i = 1, 2, \ldots, q$. We also use the convention that $C_0$ is an initialization vector.

For for any set $S \subset \{0,1\}^w$ and any element $x \in \{0,1\}^w$, we use $S \oplus x$ to denote the set $\{x \oplus s : s \in S\}$. For any sets $S, T$, $\#S$ denotes the number of elements in $S$, and $S/T$ denotes the set difference; $x \in S/T$ iff $x \in S, x \notin T$. The direct product of two sets $S, T$ is denoted as $S \times T = \{(s,t) : s \in S, t \in T\}$.

### 1.3 Modes

A block cipher mode of operation is a ways in which a fixed-width block cipher can be used to encrypt arbitrary length data. In this note, we consider the CFB, CBC, and CTR modes [5]. For $i = 1, 2, \dots, q$, the plaintext and ciphertext are related as

$$P_i = \begin{cases} E^{-1}(C_i) \oplus C_{i-1} & \text{in CBC mode} \\ E(C_{i-1}) \oplus C_i & \text{in CFB mode} \\ E(i) \oplus C_i & \text{in CTR mode.} \end{cases} \tag{1}$$

Without essential loss of generality, we assume that the number of bits in the plaintext is a multiple of $w$, for clarity of exposition. We also neglect the packetization of data and initialization vectors (other than $C_0$) since it is irrelevant to the analysis. We neglect details of how $i$ is represented as a $w$-bit string, which are also not relevant.

In each mode, there is a straightforward relation between a particular plaintext block and a single invocation of the block cipher, which we use in our attacks:

$$E_x = \begin{cases} C_i \text{ where } x = P_i \oplus C_{i-1} & \text{in CBC mode} \\ C_x \oplus P_x \text{ where } x = C_{i-1} & \text{in CFB mode} \\ C_x \oplus P_x & \text{in CTR mode.} \end{cases} \tag{2}$$

## 2 Collision attacks on CBC and CFB

The CBC and CFB modes of operation leak information about plaintext when a collision in the $w$-bit block cipher input occurs. For these modes, we denote as $I_i$ the input to the block cipher encryption operation that is associated with the $i^{th}$ plaintext:

$$I_i = \begin{cases} C_i & \text{in CBC mode} \\ C_{i-1} & \text{in CFB mode.} \end{cases} \tag{3}$$

A *collision* is the event that $I_i = I_j$ for some $i \neq j$; this event is easy to observe, based only on the value of the ciphertext blocks. After a collision is observed, the attacker learns a relationship between the values of $P_i$ and $P_j$:

$$P_i \oplus P_j = \Delta_{ij}, \text{ where } \Delta_{ij} = \begin{cases} C_{j-1} \oplus C_{i-1} & \text{in CBC mode} \\ C_j \oplus C_i & \text{in CFB mode.} \end{cases} \tag{4}$$

If the attacker has some knowledge of the plaintext, as is typical, then this relationship provides them with additional information. If the attacker knows all or part of $P_i$, the relationship allows them to infer all or part of $P_j$. Likewise, $P_i$ can be learned if $P_j$ is known. If the attacker has partial knowledge about both $P_i$ and $P_j$, then Equation 4 increases the knowledge about both values.

This information can be quantified as follows. Let $\mathbf{P}[P_i = x]$ and $\mathbf{P}[P_j = y]$ denote the 'prior' probabilities known to the attacker, where $x, y \in \{0, 1\}^w$. These probability distributions for $P_i$ and $P_j$ contain the attacker's knowledge about those plaintext blocks before the attack. We let $P_i \oplus P_j = \Delta$ denote the event that a collision is observed between blocks $i$ and $j$, and thus that equality holds. The probability estimate of those plaintext values are improved via Bayes' equation, that is, the distribution of $P_i$ conditioned on the collision event is

$$\mathbf{P}[P_i = x | P_i \oplus P_j = \Delta] = \frac{\mathbf{P}[P_j = x \oplus \Delta]\mathbf{P}[P_i = x]}{\sum_y \mathbf{P}[P_j = y \oplus \Delta]\mathbf{P}[P_i = y]}. \tag{5}$$

In most real-world scenarios, an attacker knows enough about the plaintext to be able to exploit these relationships very effectively, though doing so may require the use of specialized statistical methods. These techniques are well understood and have been successfully applied to cryptanalysis. The work of Mason et. al. [15] assumes plaintext relations in the form of Equation 4, and is successful at automatically uncovering the most likely plaintext values based only on a statistical language model of the plaintext. To give an idea of how these methods work, we present a motivating example. We suppose that the attacker knows that the first two bytes of $P_i$ are in the set { 0A00, AC10, C0A8 }, as would be the case if that block starts with an IPv4 private address [25], and the first two bytes of $P_j$ contain ASCII data [4], and thus each of the first two bytes of $P_j$ is less than 80 (all numbers in hexadecimal). By comparing the first two bytes of $\Delta_{ij}$ to 80, the attacker can find the first two bytes of $P_i$, and from that value, can find the first two bytes of $P_j$.

The following lemma shows the effectiveness of the collision attack, using a simple plaintext model; the proof is in Appendix A.

**Lemma 1.** *The expected number of bits of unknown plaintext that are revealed in a collision attack with $k$ blocks of known plaintext and $u$ blocks of unknown plaintext is $\frac{wku}{2^w} \leq n^2 \frac{w}{2^{w+2}}$, where $n = k + u$.*

This model underestimates the data leakage, because it neglects the fact that when the attacker learns the value of one plaintext block, she may gain partial information about other plaintext blocks. Nonetheless, it is useful for understanding the attack.

An efficient method of attack is detailed in Appendix B, which requires $\mathcal{O}(n)$ operations on $2^w$-bit words, and $\mathcal{O}(n)$ storage. This attack is easy to prosecute, and in fact the attacker's computational cost is roughly the same as that of the victim. In practice, storage costs will likely dominate the cost of the attack, though a one terabyte disk costs less than $100 and with $2^{43}$ bits, is sufficient when $w = 64$. An attacker faced with a situation in which $n > 2^{\frac{w}{2}+1}$ but only $m < n$ ciphertext blocks can be stored can maximize their results with the strategy of discarding unknown-plaintext blocks if $u > k$ and discarding known-plaintext blocks otherwise.

### 2.1 Rekeying

Lemma 1 assumes that the block cipher key is unchanged, so that all blocks are encrypted with the same key. The best mitigation against these attacks (other than using a cipher with a larger value of $w$) is to rekey the block cipher; that is, to use multiple keys, so as to limit the total amount of data encrypted with any single key. Assume that the total number of blocks to be encrypted is $t$, and that the number encrypted under a single key is $n$. Then expected number of bits of information leakage is $tn\frac{w}{2^{w+2}}$; see Section 4 for a concrete example. This equation highlights the important relationship between the total amount of data encrypted and the information leakage. If there is a practical lower limit on the rate at which new keys can be established, that determines the amount of information that a patient attacker can learn from observing traffic across all keys. Note that since the computational costs of the attack are less than the costs of the storage required, there is not much additional incremental cost to the attacker.

## 3 Impossible plaintext cryptanalysis of CTR

Collision attacks do not work against CTR when it is used with a deterministic nonce (as is conventional), because the there are no collisions in the block cipher inputs. These attacks are also inapplicable to CTR-based modes such as CCM [7] and GCM [6]. However, CTR does reveal information about the plaintext when it is used beyond the birthday bound. In this section, we introduce a plaintext-recovery attack against CTR (and CTR-based modes) that uses *impossible plaintexts*; it tracks which values of a particular plaintext block are possible, given the attacker's knowledge of other parts of the plaintext, and winnows out the correct one. This method can also be used to improve the effectiveness of attacks against other modes.

An attacker can make inferences about CTR-encrypted data as follows. From Equation 1 and because of the invertibility of $E$, the attacker knows that $E(i) \neq E(j)$ for any $i \neq j$, and thus

$$P_i \neq P_j \oplus C_i \oplus C_j. \tag{6}$$

An attacker who knows $P_j$, and observes $C_i$ and $C_j$, learns that a particular value of the plaintext is impossible.

## 3.1 Plaintext model

Before detailing attack techniques, we first establish a model for the plaintext source. We let $\mathcal{K}$ denote a set of known plaintext blocks; for all $i \in \mathcal{K}$, the attacker knows the value of $P_i$. We assume that the attacker understands (at least part of) the format of the plaintext, and we characterize the attacker's knowledge about the plaintext as follows. An *unknown repeated value* occurs when $P_i = \phi$ for all $i \in \mathcal{R}$ for some unknown value $\phi$ and some set $\mathcal{R}$ of blocks. We say that the unknown value has *repetition* $r$ when $\#\mathcal{R} = r$. When an unknown plaintext value is information that the attacker considers valuable and wants to discover, we call that value a *target*. When it is data that is of no interest to the attacker, we call it an *incidental value*. An incidental value can be used as a stepping stone to learning target values. We denote as $\Phi = \{\phi_1, \phi_2, \dots, \phi_s\}$ the set of possible plaintext values; that is, the attacker knows that $P_i = \phi \in \Phi$ for all $i \in \mathcal{R}$, but does not know the actual value of $\phi$.

Unknown repeated values occur more often in real-world systems than one might initially expect. For instance, every 1500 byte IPv4 packet contains two four-byte addresses, and every IPv6 packet with the same size contains two sixteen-byte addresses; as these values are constant for a particular data flow, they present repeated values that comprise 0.5% and 2.1% of those flows. When HTTP Basic Authentication [9] is used, a username/password value is repeated in each HTTP request.

## 3.2 Inferences

Equation 6 provides a small amount of information about $P_i$, based on knowledge of $P_j$. Collecting this knowledge across all of the known plaintexts leads to the first part of the following lemma; collecting information across each ciphertext $C_i : i \in \mathcal{R}$, gives the second part.

**Lemma 2.** *For any ciphertext block $C_i : i \notin \mathcal{K}$ the corresponding plaintext block $P_i \notin (\mathcal{E} \oplus C_i)$, where $\mathcal{E} = \{E(j) : j \in \mathcal{K}\} = \{P_j \oplus C_j : j \in \mathcal{K}\}$.*

*An unknown repeated value $\phi$ corresponding to the set $\mathcal{R}$ satisfies $\phi \notin \mathcal{E} \oplus \Gamma$, where $\Gamma = \{C_j : j \in \mathcal{R}\}$.*

We estimate that an impossible plaintext attack against an unknown repeated value with repetition $r$, a possible plaintext set of size $\#\Phi = s$, and $k = \#\mathcal{E}$ known plaintext blocks succeeds when $kr \geq (\ln(s) + 1)2^w$. This estimate is derived as follows. We make the heuristic assumption that the elements of the set $\mathcal{E} \times \Gamma$ are uniformly random and independent, and then model the attack as the coupon collector's problem [10], where attacker seeks to find $s - 1$ coupons (possible plaintexts) by drawing $kr$ coupons uniformly at random (from $\mathcal{E} \times \Gamma$). Finding a coupon is equivalent to eliminating a possible plaintext. After $j$ possible plaintexts have been collected, the number of draws required to collect the next one follows a geometric distribution with $p = 1 - j2^{-w}$, so the expected number of draws needed to find the $j + 1^{th}$ possible plaintext is $\frac{2^w}{2^w - j}$. When $s < 2^w$, it is as if $2^w - s$ coupons have already been found before the start of the attack, so the total number of draws needed to eliminate all $s - 1$ possible plaintexts is

$$2^w \sum_{j=2^w-s}^{2^w-1} \frac{1}{2^w - j} = 2^w \left[ \frac{1}{s} + \frac{1}{s-1} + \ldots + \frac{1}{2} + 1 \right] = 2^w \sum_{i=1,}^{s} \frac{1}{i}, \qquad (7)$$

from the linearity of expectations. The sum in the rightmost term is the $s^{th}$ harmonic number; bounding this value [13] gives that the expected number of draws needed to find all possible plaintexts is in between $(\ln(s) + \frac{1}{2})2^w$ and $(\ln(s) + 1)2^w$, and the estimate follows directly. See Appendix C for further discussion.

## 3.3 Implementation

Two different attack approaches are shown in Algorithms 1 and 2. Algorithm 1 uses a sieving method to eliminate the elements of $\Phi$ that are impossible solutions. It loops over all of the elements of $\mathcal{E} \times R$, and thus takes $rk$ operations on the set $\Phi$. If a simple storage method is used to hold $\Phi$, then at least $s$ bits of storage are needed. Algorithm 2 uses a searching method, and loops over each of the elements of $\Phi \times \mathcal{R}$, thus taking $rs$ operations and requiring only $r + k$ storage. Both algorithms are easily parallelizable, and can be used in situations in which the block cipher key changes, as long as the repeated value is constant, since the set $\Phi$ of possible plaintexts can be maintained across those key changes. Of course, $\mathcal{E}$ needs to be recomputed for each new key.

**Algorithm 1** (sieving, left) and **Algorithm 2** (searching, right)

```
for $\epsilon \in \mathcal{E}$ do
    for $i \in \mathcal{R}$ do
        remove $C_i \oplus \epsilon$ from $\Phi$
    end for
end for
return $\Phi$
```

```
for $\phi \in \Phi$ do
    for $i \in \mathcal{R}$ do
        if $C_i \oplus \phi \in \mathcal{E}$ then
            remove $\phi$ from $\Phi$
        end if
    end for
end for
return $\Phi$
```

Faster algorithms are possible when more than the minimum amount of known plaintext is available; see Appendix D.

### 3.4 Hybrid algorithm

The seiving algorithm takes fewer computations whenever $k < s$, while the searching algorithm takes fewer computations otherwise. The first few passes of the seiving algorithm greatly reduce the size of the possible plaintext set. Thus, a better attack strategy for when $k < s$ is to use the seiving algorithm until $\#\Phi$ has been reduced in size enough so that $\#\Phi < k$, then switch to the latter algorithm. An efficient hybrid algorithm can be constructed by noting that $\mathcal{E}$ can be divided into two distinct sets $\mathcal{E} = \mathcal{E}^1 \cup \mathcal{E}^2$, and $\mathcal{E}^1$ can be used in the seiving stage, while $\mathcal{E}^2$ is used in the searching stage, without affecting the reliability of the algorithm at finding the correct plaintext. In practice, the exact size of the set $\mathcal{E}^1$ need not be known in advance, the attacker can count the number of remaining possible plaintexts during the sieving stage, and adaptively choose when to move the attack to the searching stage.

## 4 Discussion

Network data speeds grow exponentially (though not as fast as Moore's law) [21], and modern networks achieve speeds that were not in consideration when the 64-bit block ciphers Triple-DES and GOST 28147-89 were conceived (in 1979 and 1989, respectively). One day is 86,400 seconds. At high data rates, it is easy for a network encryption device to exceed the birthday bound for 64-bit block ciphers in that length of time. At one megabit per second, a data stream generates 15,625 64-bit blocks per second, or $1.35 \times 10^9 \simeq 2^{30.3}$ blocks per day. At one gigabit per second, a data stream generates about $2^{40.3}$ blocks per day. At these data rates, and a one-day key lifetime, the birthday bound is exceeded for 64-bit block ciphers. Figure 1 illustrates this expected plaintext leakage, and Table 1 shows how dramatically better the security of 128-bit block ciphers is.

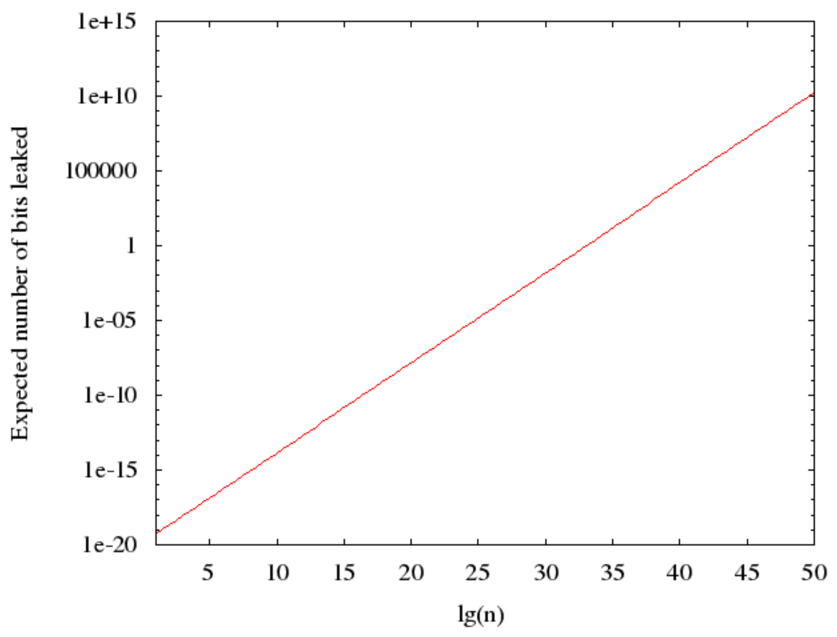

Fig. 1: The expected number of bits leaked due to collisions in as a function of the logarithm of the number $n$ of ciphertext blocks, as per Equation 1, with a $w = 64$-bit block cipher.

| | 1 Mbit/s for one day | 1 Gbit/s for one day | 1 Tbit/s for one day |
|---|---|---|---|
| $w = 64$ | 6.3 bits | $6.3 \times 10^{6}$ bits | $6.3 \times 10^{12}$ bits |
| $w = 128$ | $1.7 \times 10^{-19}$ bits | $1.7 \times 10^{-13}$ bits | $1.7 \times 10^{-7}$ bits |

Table 1: The expected number of bits leaked due to collisions, as per Lemma 1.

At very high data rates, it is difficult to keep leakage low, even with frequent rekeying, because the total leakage is summed over all of the many rekeying periods. For instance, a 64-bit block cipher that is rekeyed every ten seconds at 10 Gbit/s will leak about two bits every ten seconds, or 2 kilobytes per day, 15 kilobytes per week, 60 kilobytes per month, 788 kilobytes per year. Clearly, $w = 64$ is undesirable at these data rates and should be avoided when possible.

An implementation that uses manually installed keys in a 64-bit block cipher is especially problematic, since it cannot avail itself of the rekeying strategy of Section 2.1. When automated key management is used, there may still be a risk that an attacker can prevent rekeying from taking place by dropping the key establishment messages, and thus prolong the use of a particular key. As a general rule, 64-bit block ciphers are inappropriate for encrypting at high data rates, and 128-bit block ciphers are much more suitable in such scenarios.

Arguably, the vulnerabilities exploited in this work are more relevant to security practice than differential, linear, integral, algebraic, and biclique cryptanalysis. The biclique attack on AES with 128-bit keys, for instance, is an important recent advance in the cryptanalysis of block ciphers [3]. Prosecuting that attack requires $2^{88}$ chosen plaintexts/ciphertexts and takes $\mathcal{O}(2^{126})$ operations, which is a million times as much ciphertext as that needed for collision attacks, and $2^{62}$ times as much computation. Furthermore, it suffices to observe the victim's traffic, without interacting with it, in order to prosecute the attacks described in this note. In contrast, attacks based on differential, linear, integral, algebraic, and biclique cryptanalysis typically require that the victim compute the decryption of many ciphertexts, and/or the encryption of many plaintexts, which have been chosen by the attacker. They are thus much harder to perpetrate against real-world systems, most of which use modes of operation like CBC, CFB, and CTR, in which the block cipher inputs cannot be controlled by an attacker in a chosen-plaintext or chosen-ciphertext attack.

Interestingly, the modern block cipher modes CCM and GCM are specified only for use with 128-bit block ciphers, and consciously omit smaller block sizes. They also define a maximum on the number of blocks that can be encrypted with a single key. The specifications for the traditional modes accept any value of $w$, and do not include similar upper bounds. This contrast no doubt reflects the heightened expectations that the modern cryptography community has put onto new techniques, and the newer standards are better because of it.

There are block cipher encryption modes that are secure beyond the birthday barrier, such as Iwata's CENC/CHM [11] and AE1 [12]. These modes use slightly more than $n$ block cipher invocations to encrypt $n$ blocks of plaintext, and are indistinguishable from random above the birthday bound. For applications that are constrained to use 64-bit block ciphers for some reason, these modes may be appealing; of course, the best security would be provided by using one of these modes with a 128-bit block cipher!

CTR is more difficult to attack than CBC or CFB; in real-world scenarios, it appears that CTR leaks information more slowly than the other modes. This fact is evident from a comparison of Equations 4 and 6; an attacker learns much more information from the equality $P_i \oplus P_j = \Delta$ than the inequality $P_i \oplus P_j \neq \Delta$ .

## 5 Conclusions

We show attacks against $w$-bit block ciphers in CBC, CFB, and CTR modes that can recover an unknown plaintext values when the birthday bound is not respected. These attacks are practical against $w = 64$-bit block ciphers that are used at, or even slightly below, the birthday bound. The collision-based attacks against CBC and CFB are straightforward and relatively inexpensive to carry out against 64-bit block ciphers; the impossible plaintext attacks against CTR are more involved, but are still feasible. The only protection against these attacks is to stay well below the birthday bound. 128-bit block ciphers provide substantially better security than 64-bit ones, with the exception of scenarios in which all encryption takes place at very low data rates and keys are changed often.

## References

1. Mihir Bellare, Anand Desai, Eric Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science*, pages 394–403, Miami Beach, Florida, October 19–22, 1997. IEEE Computer Society Press.
2. Steven M. Bellovin. Probable plaintext cryptanalysis of the IP security protocols. In *ISOC Network and Distributed System Security Symposium – NDSS'97*, San Diego, California, USA, February 10–11, 1997. The Internet Society.
3. Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full AES. In *Advances in Cryptology – ASIACRYPT 2011*, Lecture Notes in Computer Science, pages 344–371. Springer, Berlin, Germany, December 2011.
4. V.G. Cerf. ASCII format for network interchange. RFC 20, October 1969.
5. Morris Dworkin. Recommendation for Block Cipher Modes of Operation: Methods and Techniques. National Institute of Standards and Technology (NIST) Special Publication SP 800-38A, 2001.
6. Morris Dworkin. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. National Institute of Standards and Technology (NIST) Special Publication SP 800-38C, 2004.

7. Morris Dworkin. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality. National Institute of Standards and Technology (NIST) Special Publication SP 800-38C, 2004.
8. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Updated by RFCs 2817, 5785.
9. J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, and L. Stewart. HTTP Authentication: Basic and Digest Access Authentication. RFC 2617 (Draft Standard), June 1999.
10. Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes.* Oxford University Press, 1992.
11. Tetsu Iwata. New blockcipher modes of operation with beyond the birthday bound security. In Matthew J. B. Robshaw, editor, *Fast Software Encryption – FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 310–327, Graz, Austria, March 15–17, 2006. Springer, Berlin, Germany.
12. Tetsu Iwata. Authenticated encryption mode for beyond the birthday bound security. In Serge Vaudenay, editor, *AFRICACRYPT 08: 1st International Conference on Cryptology in Africa*, volume 5023 of *Lecture Notes in Computer Science*, pages 125–142, Casablanca, Morocco, June 11–14, 2008. Springer, Berlin, Germany.
13. G. Klambauer. Problems and propositions in analysis, 1979.
14. Donald E. Knuth. The art of computer programming, volume iii: Sorting and searching. In *The Art of Computer Programming, Volume III: Sorting and Searching*, 1973.
15. Joshua Mason, Kathryn Watkins, Jason Eisner, and Adam Stubblefield. A natural language approach to automated cryptanalysis of two-time pads. In Ari Juels, Rebecca N. Wright, and Sabrina De Capitani di Vimercati, editors, *ACM CCS 06: 13th Conference on Computer and Communications Security*, pages 235–244, Alexandria, Virginia, USA, October 30 – November 3, 2006. ACM Press.
16. David A. McGrew and Scott R. Fluhrer. Attacks on additive encryption of redundant plaintext and implications on internet security. In Douglas R. Stinson and Stafford E. Tavares, editors, *SAC 2000: 7th Annual International Workshop on Selected Areas in Cryptography*, volume 2012 of *Lecture Notes in Computer Science*, pages 14–28, Waterloo, Ontario, Canada, August 14–15, 2001. Springer, Berlin, Germany.
17. Willi Meier and Othmar Staffelbach. Fast correltaion attacks on stream ciphers (extended abstract). In C. G. Günther, editor, *Advances in Cryptology – EUROCRYPT'88*, volume 330 of *Lecture Notes in Computer Science*, pages 301–314, Davos, Switzerland, May 25–27, 1988. Springer, Berlin, Germany.
18. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* The CRC Press series on discrete mathematics and its applications. CRC Press, 2000 N.W. Corporate Blvd., Boca Raton, FL 33431-9868, USA, 1997.
19. Ralph C. Merkle. A certified digital signature. In Gilles Brassard, editor, *Advances in Cryptology – CRYPTO'89*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238, Santa Barbara, CA, USA, August 20–24, 1990. Springer, Berlin, Germany.
20. National Institute of Standards and Technology. FIPS PUB 46-3: Data Encryption Standard (DES), October 1999.
21. Jakob Nielsen. Nielsen's law of internet bandwidth (web page). http://www.useit.com/alertbox/980405.html, 2010.

22. V. Popov, I. Kurepkin, and S. Leontiev. Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms. RFC 4357 (Informational), January 2006.
23. J. Postel. Internet Protocol. RFC 791 (Standard), September 1981. Updated by RFC 1349.
24. J. Postel. Transmission Control Protocol. RFC 793 (Standard), September 1981. Updated by RFCs 1122, 3168.
25. Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. Address Allocation for Private Internets. RFC 1918 (Best Current Practice), February 1996.
26. Technical Specification Group Services and 3G Security System Aspects. Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification, V3.1.1 . 3rd Generation Partnership Project,, 2001.
27. W. Tuchman. Hellman presents no shortcut solutions to DES. *IEEE Spectrum*, 16(7):40–41, July 1979.

## A Proof of Lemma 1

*Proof.* Assume that the set of all $k$ ciphertexts corresponding to known plaintexts is fixed; the probability that a single unknown ciphertext will collide with a known plaintext is $\frac{k}{2^w}$, under the reasonable assumption that the ciphertexts are random[1]. Let $X$ denote the random variable that counts the number of collisions after $r$ unknown ciphertexts have been generated. There are exactly $\binom{r}{x}$ distinct ways that $x$ collisions can occur. Thus $X$ has a binomial distribution with probability $\frac{k}{2^w}$ and parameter $r$. The expectation of $X$ is therefore $\frac{rk}{2^w}$ (see for instance [10]) and the main result follows from the linearity of the expectation operator.

## B Collision attack

An efficient algorithm for finding all of the exploitable collisions present in CBC and CFB modes of operation is presented in Algorithm 3, in which $\mathbf{K} = \{I_j : j \in \mathcal{K}\}$ is the set of block cipher inputs corresponding to known plaintexts, and $\mathbf{U} = \{I_i : i \notin \mathcal{K}\}$ is the set of block cipher inputs corresponding to unknown plaintexts. If a radix sort [14] is used, a set of size $u$ can be sorted in $\mathcal{O}(u)$ operations on $w$-bit words; then the algorithm takes $\mathcal{O}(n)$ operations, and requires storage on the same order.

[1] If the ciphertexts are not random, the cipher has an exploitable statistical defect.

**Algorithm 3** Collision attack

```
sort $\mathbf{K}$ and $\mathbf{U}$ into ascending order
$i \leftarrow 1, j \leftarrow 1$
while $i < u, j < k$ do
    if $\mathbf{K}_j > \mathbf{U}_i$ then
        $i \leftarrow i + 1$
    else if $\mathbf{K}_j < \mathbf{U}_i$ then
        $j \leftarrow j + 1$
    else
        $P_i = P_j \oplus \Delta_{ij}$
        $i \leftarrow i + 1$, $j \leftarrow j + 1$
    end if
end while
```

## C Estimating the success probability of impossible plaintext cryptanalysis

The elements of $\mathcal{E} \times \Gamma$ are not, of course, uniformly and independently random, but there is some evidence supporting our assumption of treating them as though they are. The most important consideration is the size of that set; our model implicitly assumes that it is at least as large as a set of $kr$ elements chosen uniformly at random. We consider the $k \times r$ matrix $M : M_{ij} = \epsilon_i \oplus \gamma_j$. The elements of $\mathcal{E}$, and the elements of $\Gamma$, are distinct because of the invertibiltiy of the block cipher, and because of this, there cannot be any collisions among the elements within each row, and each column of $M$. Additionally, there can be at most one collision between each pair of rows, and between each pair of columns. Therefore, in some ways the structure of $\mathcal{E} \times \Gamma$ avoids collisions that would reduce its size. Interestingly, $M$ is a rectangular sub-array of a latin square.

## D Fast attacks using cosets

If the attacker has more known plaintext than the minimum required, it may be possible to structure the attack in such a way that its computational cost is lower than described in Section 3.3. This situation is similar to that of fast correlation attacks on stream ciphers [17]. If $\mathcal{E}' \subset \mathcal{E}$ is a linear subspace of $\{0,1\}^w$, then $\mathcal{E}' \oplus \gamma$ is a coset of $\mathcal{E}'$. Each coset is distinct, that is, for any distinct values $\gamma_i, \gamma_j \in \Gamma$, $\mathcal{E}' \oplus \gamma_i$ and $\mathcal{E}' \oplus \gamma_j$ are either identical or distinct. Thus $\{0,1\}^w = (\mathcal{E}' \oplus \gamma_1) \cup (\mathcal{E}' \oplus \gamma_2) \cup \cdots \cup (\mathcal{E}' \oplus \gamma_r)$, and the correct plaintext is in the set $\cup_{\gamma \notin \Gamma} \mathcal{E}' \oplus \gamma$, which has size $2^w - kr'$, where $r'$ is the number of distinct cosets. It is easy to enumerate the elements of this set with $r$ linear operations on $w$-bit words. If there is an abundance of known plaintext, then the attacker can exploit this fact by finding a linear subspace $\mathcal{E}'$ of $\mathcal{E}$.