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

David McGrew
mcgrew@cisco.com

Fast Software Encryption Workshop 2013
March 11-13, 2013
Outline

1 Background

2 Collision attack on CBC and CFB
   - How it works
   - Recovering plaintext
   - Efficacy
   - Rekeying

3 Impossible plaintext cryptanalysis of CTR
   - Algorithms

4 Conclusions
Block ciphers

$\mathcal{w}$-bit block cipher with a $\kappa$-bit key

\[ E : \{0, 1\}^{\mathcal{w}} \times \{0, 1\}^{\kappa} \rightarrow \{0, 1\}^{\mathcal{w}}, \]
\[ E^{-1} : \{0, 1\}^{\mathcal{w}} \times \{0, 1\}^{\kappa} \rightarrow \{0, 1\}^{\mathcal{w}} \text{ such that } \]
\[ E(E^{-1}(x)) = E^{-1}(E(x)) = x \text{ for all } x \in \{0, 1\}. \]
Block ciphers

**w-bit block cipher with a \( \kappa \)-bit key**

\[
E : \{0, 1\}^w \times \{0, 1\}^\kappa \rightarrow \{0, 1\}^w, \\
E^{-1} : \{0, 1\}^w \times \{0, 1\}^\kappa \rightarrow \{0, 1\}^w \text{ such that} \\
E(E^{-1}(x)) = E^{-1}(E(x)) = x \text{ for all } x \in \{0, 1\}.
\]

**Examples**

<table>
<thead>
<tr>
<th>Block Cipher</th>
<th>( w )</th>
<th>( \kappa )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MISTY</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>KASUMI</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>Triple-DES</td>
<td>64</td>
<td>168</td>
</tr>
<tr>
<td>GOST 28147-89</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>AES</td>
<td>128</td>
<td>128, 192, 256</td>
</tr>
</tbody>
</table>
Modes of operation

\[ P \]

\[ C \]
Modes of operation

\[ P_i = \begin{cases} 
E^{-1}(C_i) \oplus C_{i-1} & \text{in CBC mode} \\
E(C_{i-1}) \oplus C_i & \text{in CFB mode} \\
E(i) \oplus C_i & \text{in CTR mode.}
\end{cases} \]
Plaintext model
Indicator

\[ I_i = \begin{cases} C_i & \text{in CBC mode} \\ C_{i-1} & \text{in CFB mode.} \end{cases} \]
How it works

Indicator collisions reveal information

When \( I_i = I_j \) for some \( i \neq j \) then \( P_i \oplus P_j = \Delta_{ij} \), where

\[
\Delta_{ij} = \begin{cases} 
C_{j-1} \oplus C_{i-1} & \text{in CBC mode} \\
C_j \oplus C_i & \text{in CFB mode}.
\end{cases}
\]
Exploiting collisions in theory

Attacker’s knowledge about $P_j \rightarrow$ knowledge about $P_i$
Exploiting collisions in theory

Attacker’s knowledge about $P_j \rightarrow$ knowledge about $P_i$

$$\mathbb{P}[P_i = x \mid P_i \oplus P_j = \Delta] = \frac{\mathbb{P}[P_j = x \oplus \Delta] \mathbb{P}[P_i = x]}{\sum_y \mathbb{P}[P_j = y \oplus \Delta] \mathbb{P}[P_i = y]}$$
## Exploiting collisions in practice

| $P_i$          | 00001010000000000000 | 10.0.*.*          |
|               | 10101100000100000000 | 172.16.*.*        |
|               | 11000000101010000000 | 192.168.*.*       |
Exploiting collisions in practice

<table>
<thead>
<tr>
<th>$P_i$</th>
<th>00001010000000000000</th>
<th>10.0.<em>.</em></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10101100000100000000</td>
<td>172.16.<em>.</em></td>
</tr>
<tr>
<td></td>
<td>11000000101010000000</td>
<td>192.168.<em>.</em></td>
</tr>
<tr>
<td>$P_j$</td>
<td>1*****<strong><strong>1</strong></strong>****</td>
<td>ASCII</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Exploiting collisions in practice

| $P_i$ | 00001010000000000000 | 10.0.*.*  
|      | 101011000000100000 | 172.16.*.*  
|      | 110000001010100000 | 192.168.*.*  

| $P_j$ | 1********1******** | ASCII  

| Δ$_{ij}$ | 1********1******** | $P_i = 10.0.*.*$  
|         | 0********1******** | $P_i = 172.16.*.*$  
|         | 0********0******** | $P_i = 192.168.*.*$  

Birthday bound for indicator collisions

\( O(n) \) work and storage
Efficacy

Lemma

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 \( n = k + u \).
expected number of bits leaked due to collisions
expected number of bits leaked due to collisions

![Graph showing expected number of bits leaked vs. log(n)]
Network traffic with one-day rekeying

<table>
<thead>
<tr>
<th>Bits leaked per day</th>
<th>1 Mbit/s</th>
<th>1 Gbit/s</th>
<th>1 Tbit/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w$</td>
<td>6.3 bits</td>
<td>$6.3 \times 10^6$ bits</td>
<td>$6.3 \times 10^{12}$ bits</td>
</tr>
<tr>
<td>64</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>$1.7 \times 10^{-19}$ bits</td>
<td>$1.7 \times 10^{-13}$ bits</td>
<td>$1.7 \times 10^{-7}$ bits</td>
</tr>
</tbody>
</table>
Rekeying to limit leakage

- Idea: limit number of blocks encrypted under each distinct key

**Corollary**

The expected number of bits of unknown plaintext that are leaked when a total $t$ blocks are encrypted, changing keys every $c$ blocks, is less than or equal to

$$tcw^{2-w-2}$$
Rekeying to limit leakage

- Idea: limit number of blocks encrypted under each distinct key

Corollary

The expected number of bits of unknown plaintext that are leaked when a total $t$ blocks are encrypted, changing keys every $c$ blocks, is less than or equal to

$$ tcw2^{-w-2} $$

Example: $n = 2^{20}$, $t \leq 2^{w-18-\lg(w)} = 2^{40}$
Plaintext inferences

Given

\[ P_i = E(i) \oplus C_i \]
Plaintext inferences

Given

\[ P_i = E(i) \oplus C_i \]
\[ P_j = E(j) \oplus C_j \]
Plaintext inferences

Given

\[ P_i = E(i) \oplus C_i \]
\[ P_j = E(j) \oplus C_j \]
\[ E(i) \neq E(j) \text{ for } i \neq j \]
Plaintext inferences

Given

\[ P_i = E(i) \oplus C_i \]
\[ P_j = E(j) \oplus C_j \]
\[ E(i) \neq E(j) \text{ for } i \neq j \]

We know

\[ P_i \neq P_j \oplus C_i \oplus C_j \]
Extending across multiple known plaintexts
Extending across multiple known plaintexts

Lemma part 1

For any ciphertext block $C_i : i \notin \mathcal{K}$ the corresponding plaintext block $P_i \notin (\mathcal{E} \oplus C_i)$, where

\[ \mathcal{E} = \{ E(j) : j \in \mathcal{K} \} = \{ P_j \oplus C_j : j \in \mathcal{K} \}. \]
Plaintext model

To: bob@example.com
From: alice@example.com
Hello Bob, I need you to move the meeting to 9AM. Our visitors will be early. Thanks, Alice.

To: bob@example.com
From: alice@example.com
Hello Bob, make that 8AM. Alice

To: bob@example.com
From: mailmaster@example.com
Your new password is 1h8PSwds.

To: bob@example.com
From: alice@example.com
Hello Bob, our new minimum bid is $3.2M.
Plaintext model

To: bob@example.com
From: alice@example.com
Hello Bob, I need you to move the meeting to 9AM. Our visitors will be early. Thanks, Alice.

To: bob@example.com
From: alice@example.com
Hello Bob, make that 8AM. Alice

To: bob@example.com
From: mailmaster@example.com
Your new password is 1h8FSwds.

To: bob@example.com
From: alice@example.com
Hello Bob, our new minimum bid is $3.2M.

Target values
Plaintext model

<table>
<thead>
<tr>
<th>To: <a href="mailto:bob@example.com">bob@example.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>From: <a href="mailto:alice@example.com">alice@example.com</a></td>
</tr>
<tr>
<td>Hello Bob, I need you to move the meeting to 9AM. Our visitors will be early. Thanks, Alice.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: <a href="mailto:bob@example.com">bob@example.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>From: <a href="mailto:alice@example.com">alice@example.com</a></td>
</tr>
<tr>
<td>Hello Bob, make that 8AM. Alice</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: <a href="mailto:bob@example.com">bob@example.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>From: <a href="mailto:mailmaster@example.com">mailmaster@example.com</a></td>
</tr>
<tr>
<td>Your new password is 1h8F5wds.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To: <a href="mailto:bob@example.com">bob@example.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>From: <a href="mailto:alice@example.com">alice@example.com</a></td>
</tr>
<tr>
<td>Hello Bob, our new minimum bid is $3.2M.</td>
</tr>
</tbody>
</table>

Known value with repetition $r=4$

Target values
Plaintext model

Known value with repetition $r=4$

Incidental value with repetition $r=3$

Target values
Extending across repeated target values

Lemma part 2

An unknown repeated target value \( p \) corresponding to the set \( \mathcal{R} \) satisfies \( \phi \notin \mathcal{E} \oplus \mathcal{G} \), where \( \mathcal{G} = \{ C_j : j \in \mathcal{R} \} \).
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

$$kr \geq (\ln(s) + 1)2^w \geq (w + 1)2^w$$
Efficacy

Estimate
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

\[
kr \geq (\ln(s) + 1)2^w \geq (w + 1)2^w
\]

Heuristic

- \( \#(\mathcal{E} \oplus \mathcal{G}) = kr \)
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

$$kr \geq (\ln(s) + 1)2^w \geq (w + 1)2^w$$

**Heuristic**

- $\#(\mathcal{E} \oplus \mathcal{G}) = kr$
- Collecting $s$ coupons
Algorithms for finding $p$

**Sieving**

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

Sieving

\[
\begin{align*}
\text{for } \epsilon \in \mathcal{E} & \quad \text{do} \\
\quad \text{for } i \in \mathcal{R} & \quad \text{do} \\
\qquad \text{remove } C_i \oplus \epsilon \text{ from } \Phi \\
\quad \text{end for} \\
\text{end for} \\
\text{return } \Phi
\end{align*}
\]

$O(kr)$ operations, $O(s)$ storage
Algorithms for finding $p$

Searching

```plaintext
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 $\Phi$
```
Algorithms for finding $p$

Searching

\[
\begin{align*}
\text{for } & \phi \in \Phi \text{ do} \\
\text{for } & i \in \mathcal{R} \text{ do} \\
& \text{if } C_i \oplus \phi \in \mathcal{E} \text{ then} \\
& \quad \text{remove } \phi \text{ from } \Phi \\
& \text{end if} \\
& \text{end for} \\
& \text{end for} \\
& \text{return } \Phi
\end{align*}
\]

$O(rs)$ operations, $O(r + k)$ storage
Hybrid algorithm

Observations

- sieving algorithm takes less work when $k < s$
- searching algorithm takes less work when $k > s$
- The first few passes of the sieving algorithm greatly reduce the size of the possible plaintext set.
Observations

- sieving algorithm takes less work when $k < s$
- searching algorithm takes less work when $k > s$
- The first few passes of the sieving algorithm greatly reduce the size of the possible plaintext set.

Hybrid algorithm for $k < s$

1. Divide $\mathcal{E}$ into two distinct sets $\mathcal{E} = \mathcal{E}^1 \cup \mathcal{E}^2$, and
2. Run the sieving algorithm with $\mathcal{E}^1$ until $\# \Phi$ has been reduced in size enough so that $\# \Phi < k$
3. Switch to sorting algorithm using $\mathcal{E}^2$
Conclusions

- CBC, CFB, CTR leak information about plaintext at birthday bound
- Can be exploited by practical attacks for \( w = 64 \)
  - Security risk at high data rates
- CTR leaks information more slowly in known-plaintext model

CBC, CFB: \( P_i \oplus P_j = \delta \)
CTR: \( P_i \oplus P_j \neq \delta \)
Thank You

mcgrew@cisco.com