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

Block ciphers

w-bit block cipher with a κ-bit key

$E : \{0,1\}^w \times \{0,1\}^\kappa \to \{0,1\}^w$,
$E^{-1} : \{0,1\}^w \times \{0,1\}^\kappa \to \{0,1\}^w$ *such that*
$E(E^{-1}(x)) = E^{-1}(E(x)) = x$ *for all* $x \in \{0,1\}$.

Block ciphers

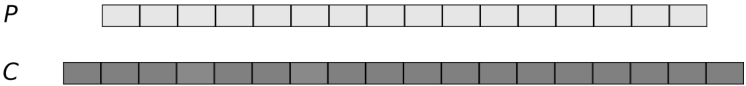
w-bit block cipher with a κ-bit key

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

Examples

MISTY	$w = 64$	$\kappa = 128$
KASUMI	$w = 64$	$\kappa = 128$
Triple-DES	$w = 64$	$\kappa = 168$
GOST 28147-89	$w = 64$	$\kappa = 256$
AES	$w = 128$	$\kappa = 128, 192, 256$

Modes of operation

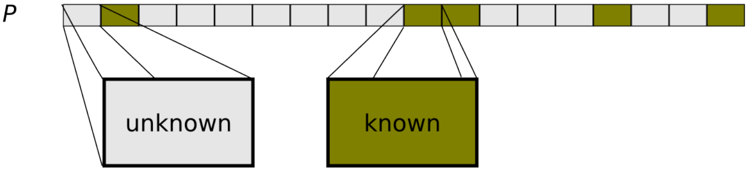


Modes of operation

Modes

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

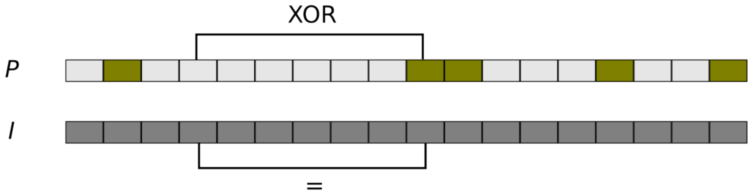
Plaintext model



Indicator

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

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

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

Exploiting collisions in practice

P_i	0000101000000000	10.0.*.*
	1010110000010000	172.16.*.*
	1100000010101000	192.168.*.*

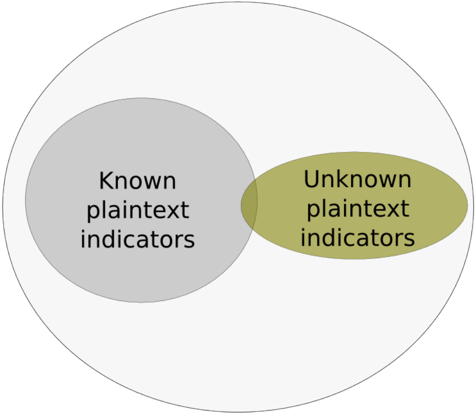
Exploiting collisions in practice

P_i	0000101000000000	10.0.*.*
	1010110000010000	172.16.*.*
	1100000010101000	192.168.*.*
P_j	1*******1*******	ASCII

Exploiting collisions in practice

P_i	0000101000000000	10.0.*.*
	1010110000010000	172.16.*.*
	1100000010101000	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



$\mathcal{O}(n)$ work and storage

Lemma

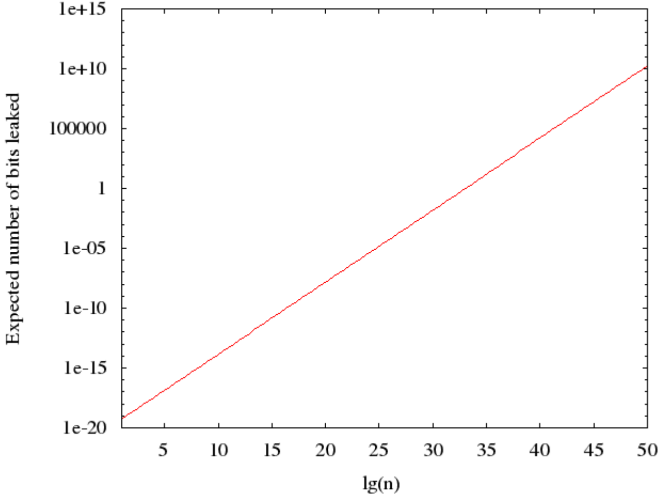
Lemma

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

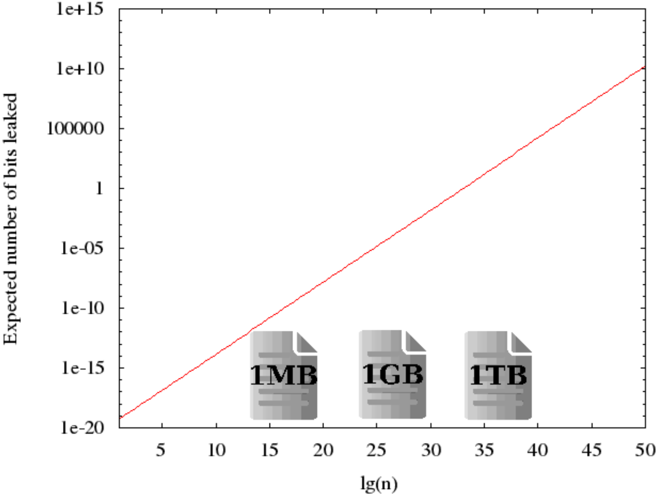
$$\frac{wku}{2^w} \leq n^2 \frac{w}{2^{w+2}},$$

where $n = k + u$.

expected number of bits leaked due to collisions



expected number of bits leaked due to collisions



Network traffic with one-day rekeying

Bits leaked per day

w	1 Mbit/s	1 Gbit/s	1 Tbit/s
64	6.3 bits	6.3×10^{6} bits	6.3×10^{12} bits
128	1.7×10^{-19} bits	1.7×10^{-13} bits	1.7×10^{-7} bits

Rekying 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

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

Rekying 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

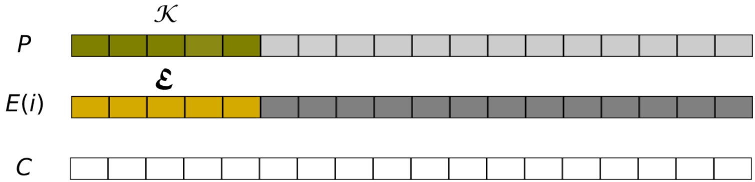
Given

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

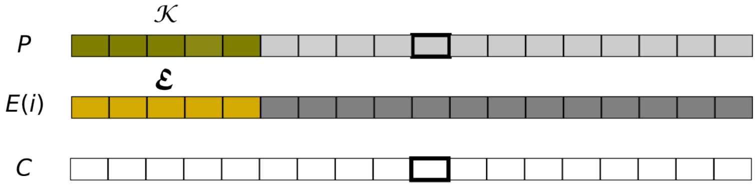
We know

$$P_i \neq P_j \oplus C_i \oplus C_j$$

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

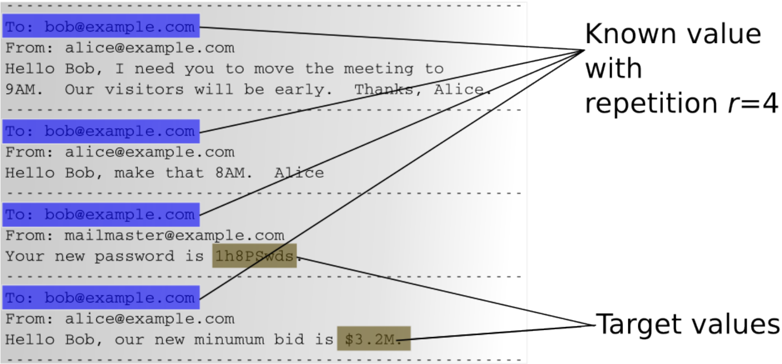
```
-----------------------------------------------------
To: bob@example.com
From: alice@example.com
Hello Bob, I need you to move the meeting to
9AM.  Our visitors will be early.  Thanks, Alice.
-----------------------------------------------------
To: bob@example.com
From: alice@example.com
Hello 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 minumum bid is $3.2M.
-----------------------------------------------------
```

Plaintext model

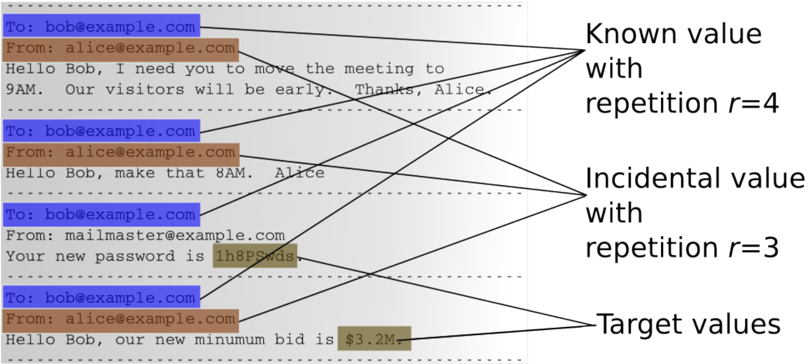
```
----------------------------------------------------
To: bob@example.com
From: alice@example.com
Hello Bob, I need you to move the meeting to
9AM.  Our visitors will be early.  Thanks, Alice.
----------------------------------------------------
To: bob@example.com
From: alice@example.com
Hello Bob, make that 8AM.  Alice
----------------------------------------------------
To: bob@example.com
From: mailmaster@example.com
Your new password is 1h8PSwds.
----------------------------------------------------
To: bob@example.com
From: alice@example.com
Hello Bob, our new minumum bid is $3.2M.
----------------------------------------------------
```

Target values

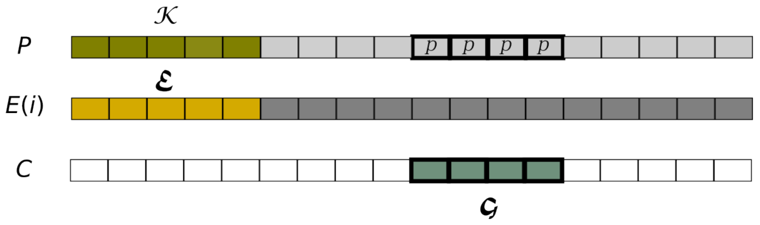
Plaintext model



Plaintext model



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 $\mathcal{G} = \{C_j : j \in \mathcal{R}\}$.

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

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

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

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

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

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

Algorithms for finding p

Sieving

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

Algorithms for finding p

Sieving

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

$\mathcal{O}(kr)$ operations, $\mathcal{O}(s)$ storage

Algorithms for finding p

Searching

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

Algorithms for finding p

Searching

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

$\mathcal{O}(rs)$ operations, $\mathcal{O}(r+k)$ 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.

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.

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

$$\text{CBC, CFB: } P_i \oplus P_j = \delta$$
$$\text{CTR: } P_i \oplus P_j \neq \delta$$

Thank You

mcgrew@cisco.com