Cryptanalysis of the "Kindle" Cipher

Alex Biryukov, Gaëtan Leurent, Arnab Roy

University of Luxembourg

SAC 2012

Cryptography: theory and practice

In theory

- Random Oracle
- Ideal Cipher
- Perfect source of randomness

In practice

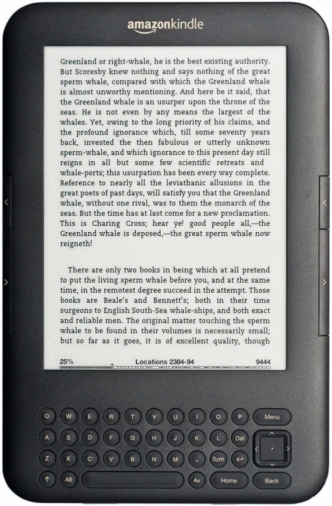
- Algorithms
  - AES
  - SHA-2
  - RSA
- Modes of operation
  - CBC
  - OAEP
  - ...
- Random Number Generators
  - Hardware RNG
  - PRNG

Cryptography in the real world

Several examples of flaws in industrial cryptography:

- Bad random source
  - SLL with 16-bit entropy (Debian)
  - ECDSA with fixed k (Sony)
- Bad key size
  - RSA-512 (TI)
  - Export restrictions...
- Bad mode of operation
  - CBC-MAC with the RC4 stream-cipher (Microsoft)
  - TEA with Davies-Meyer (Microsoft)
- Bad (proprietary) algorithm
  - A5/1 (GSM)
  - Crypto-1 (MIFARE/NXP)
  - CSS (DVD forum)
  - KeeLoq (Microchip)

Amazon Kindle

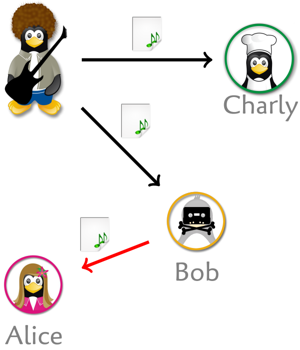


- E-book reader by Amazon
- Most popular e-book reader ($\approx$ 50% share)

- 4 generations, 7 devices
- Software reader for 7 OS, plus *cloud* reader
- Several million devices sold
- Amazon sells more e-books than paper books

- Uses crypto for DRM (Digital Rights Management)

Digital Rights Management

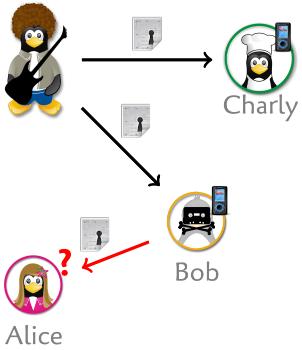


- Company sells media (music, video, ebook, game, ...)
- Wants to prevent sharing
  - Customer should read but not copy

DRM scheme

- Encipher media
- Give player to users
  - Hardware or software
- Player contains the key

Digital Rights Management



- Company sells media (music, video, ebook, game, ...)
- Wants to prevent sharing
  - Customer should read but not copy

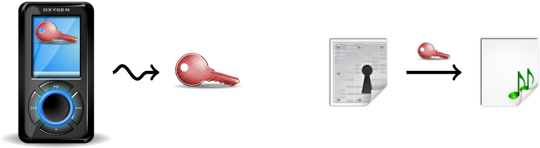
DRM scheme

- Encipher media
- Give player to users
  - Hardware or software
- Player contains the key

Breaking DRM

- Copy the media while being played

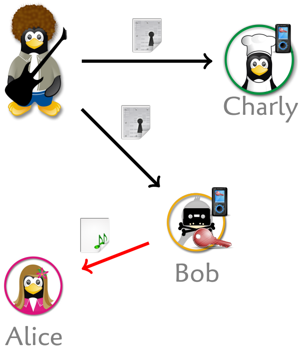
- Extract the key from the player, decipher media



Tamper-proof hardware? Obfuscation? White-box crypto?

- No need to break the crypto!
- Pirates break once, copy...

Digital Rights Management



Legal User

- Can only use authorized player
  - Collection locked-in
- DRM can restrict user rights
  - Lending, reselling, ...
- No format shifting:
  - play DVD on tablet
  - read ebook w/ speech synth.

Illegal User

- Can still find illegal copies
- Can do anything with the media

DRM on the Kindle

- Kindle e-books use DRM
- Like any DRM system, it is bound to fail
- In practice, it is easy to extract the key (Google for details...)

Overview

- In this talk, we study the cipher used in this DRM system
  We don't study the DRM system itself
- The DRM system uses a cipher called PC1
- It's a really weak cipher...

Outline

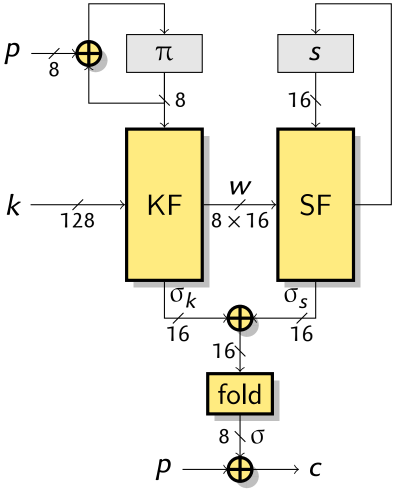
Introduction
- Cryptography in the real world
- Digital Rights Management

The PC1 Cipher
- Description
- Weaknesses

Known-plaintext key-recovery
- Collision detection
- Key recovery

Ciphertext only key-recovery
- Bias with independent keys
- Recovering the plaintext

The PC1 Cipher

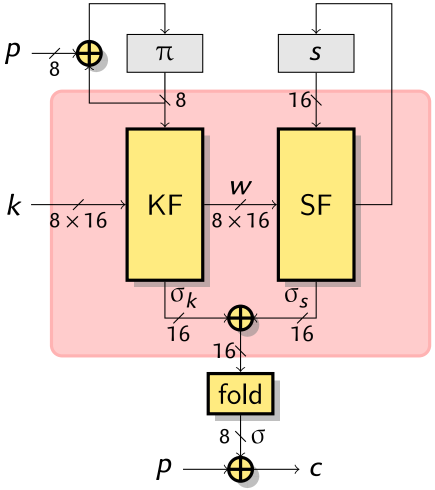


- Designed by Pukall in 1991
- Posted on Usenet
- Kindle DRM based on PC1
- Self-synchronizing stream cipher
  No IV!
- 16-bit arithmetic: add, mult, xor

Main loop (KF and SF)

for $0 \leq i < 8$ **do**
- $w \leftarrow w \oplus k_i \oplus (\pi \times 257)$
- $x \leftarrow 346 \times w$
- $w \leftarrow 20021 \times w + 1$
- $s \leftarrow s + x$
- $\sigma \leftarrow \sigma \oplus w \oplus s$
- $s \leftarrow 20021 \times (s + (i{+}1 \bmod 8)) + x$

Weakness 1: T-functions



Weakness

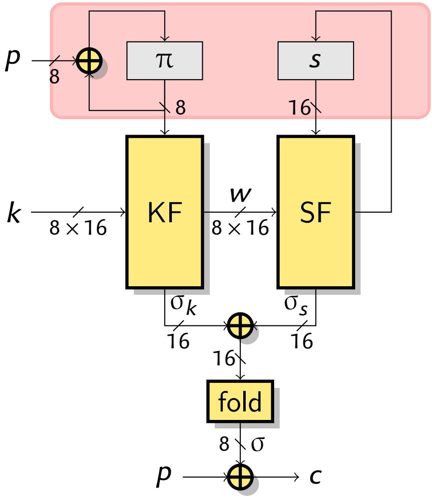
This is a T-function

- *Low bits of the output depend only on the low bits of the input*
- Add, mult, xor

- Guess 8×9 bits of the key
- Get 9 bits before the fold
- Get 1 bit after the fold
- Verify with known plaintext

- Complexity: 2^{72}
  some bytes of known plaintext

Weakness 2: small state



Weakness

The state is very small

- s 16-bit
- π 8-bit, key-independent

- Build a set of plaintexts $x_i \| y$, x_i's with fixed xor-sum
- With high probability the state collides after x_i and x_j
- Same encryption of y

- Complexity: 2^8 CP (distinguisher)

Outline

Collision detection

Can we use state collisions in a known-plaintext attack?

```
How much wood could a woodchuck chuck
```
$gfecuhaupmaqcdlvtognfgdhisqghugbrfqvc$
```
if a woodchuck could chuck wood?
```
$ghxadiaphjjxicwpidkasqghugbqsjbf$

- In a natural language text, some words will be repeated.
- With some probability ($p \approx 2^{-24}$), two instances of a repeated word begin with the same state.
- This gives a repetition in the ciphertext.

- When we detect a repetition in the plaintext and ciphertext, we can assume that the state is colliding.

Collision detection

Can we use state collisions in a known-plaintext attack?

```
How much wood could a woodchuck chuck
gfecuhaupmaqcdlvtognfgdhisqghugbrfqvc
if a woodchuck could chuck wood?
ghxadiaphjjxicwpidkasqghugbqsjbf
```

- In a natural language text, some words will be repeated.
- With some probability ($p \approx 2^{-24}$), two instances of a repeated word begin with the same state.
- This gives a repetition in the ciphertext.
- When we detect a repetition in the plaintext and ciphertext, we can assume that the state is colliding.

Collision detection

Can we use state collisions in a known-plaintext attack?

```
How much wood could a woodchuck chuck
gfecuhaupmaqcdlvtognfgdhisqghugbrfqvc
if a woodchuck could chuck wood?
ghxadiaphjjxicwpidkasqghugbqsjbf
```

- In a natural language text, some words will be repeated.
- With some probability ($p \approx 2^{-24}$),
  two instances of a repeated word begin with the same state.
- This gives a repetition in the ciphertext.
- When we detect a repetition in the plaintext and ciphertext,
  we can assume that the state is colliding.

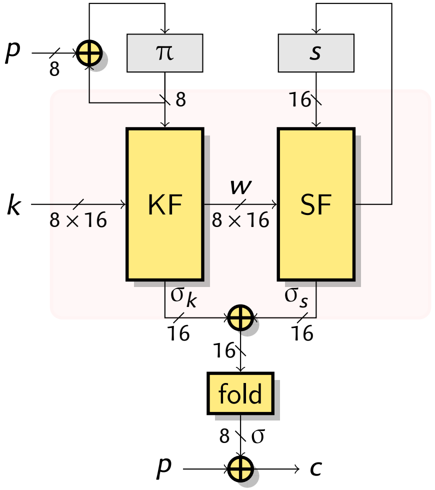
Collision detection

Can we use state collisions in a known-plaintext attack?

```
How much wood could a woodchuck chuck
gfecuhaupmaqcdlvtognfgdhisqghugbrfqvc
if a woodchuck could chuck wood?
ghxadiaphjjxicwpidkasqghugbqsjbf
```

- In a natural language text, some words will be repeated.
- With some probability ($p \approx 2^{-24}$), two instances of a repeated word begin with the same state.
- This gives a repetition in the ciphertext.
- When we detect a repetition in the plaintext and ciphertext, we can assume that the state is colliding.

Collision Based Key-recovery



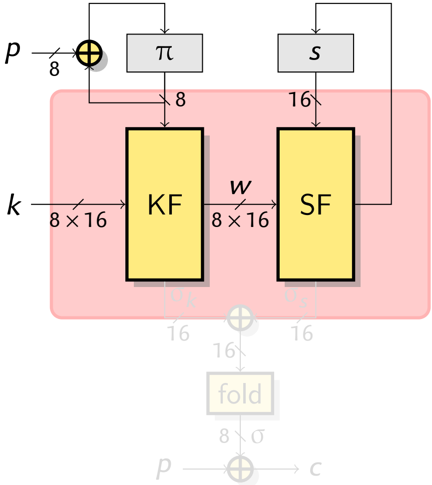
- Use state collisions to test key guess
- Skip output part

Weakness

This is a T-function

- Guess 8 × 1 bits of the key
- Compute 1 bit of *s*, check collisions in *s*
- Repeat with 2nd bit, ...

Collision Based Key-recovery



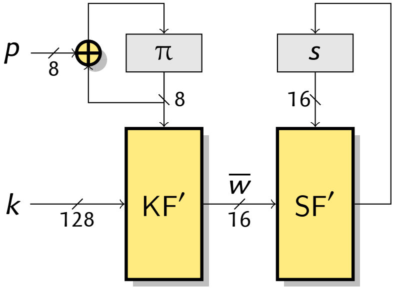
- Use state collisions to test key guess
- Skip output part

Weakness

This is a T-function

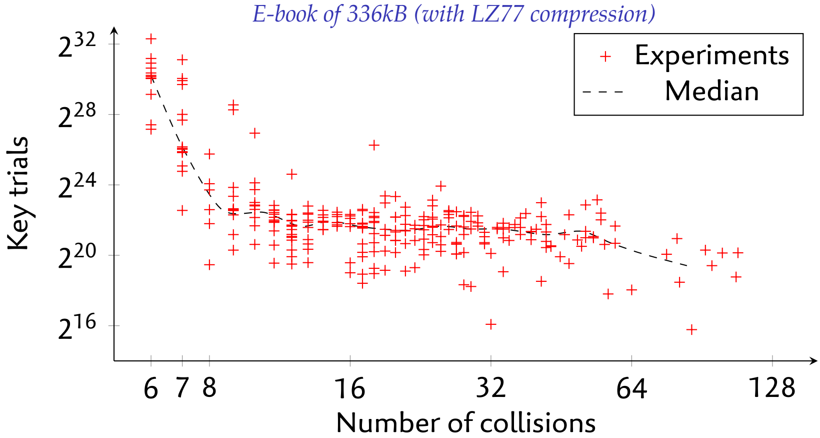
- Guess 8×1 bits of the key
- Compute 1 bit of s, check collisions in s
- Repeat with 2^{nd} bit, ...

Improving the Complexity



- Simplified state update:
  $s^{t+1} = \overline{w}^t + b \times s^t + c$
  - $\overline{w} \triangleq \sum_{i=0}^{7} (a_i \times w_i)$
  - key-dep. S-box $\mathrm{KF}' : \pi \to \overline{w}_\pi$
- Iterate the state update:
  $s^t = \mathrm{R}^t(\overline{w}_0, \ldots, \overline{w}_{255})$ linear
  Explicit with known π^t
- State collisions give linear relations of $\overline{w}_x$: $\mathrm{R}^t = \mathrm{R}^u$
  - Look for sparse relations
- For each (partial) key guess, compute $\overline{w}_x$ & check relations
  - Faster than computing s

Experiments



Practical key-recovery attack

Complexity $\approx 2^{31}$ with $\approx 2^{20}$ bytes of (low entropy) known plaintext

Key trial costs less than 256 instead of full encryption

Outline

Introduction

- Cryptography in the real world
- Digital Rights Management

The PC1 Cipher

- Description
- Weaknesses

Known-plaintext key-recovery

- Collision detection
- Key recovery

Ciphertext only key-recovery

- Bias with independent keys
- Recovering the plaintext

Ciphertext Only Attack

Main idea

If the state (s, π) collides, then the output stream σ is the same.
Note that s depend on the key, but $\pi = \bigoplus p^i$
Consider two positions t, u and a random key:

$$\Pr_K[\sigma^t = \sigma^u \quad] \approx \begin{cases} 2^{-8} & \text{if } \pi^t \neq \pi^u \\ 2^{-8} + \Pr\left[s^t = s^{t'}\right] & \text{if } \pi^t = \pi^u \end{cases}$$

$$c^t \oplus c^u = \sigma^t \oplus p^t \oplus \sigma^u \oplus p^u$$

- Consider several copies of a given text, encrypted with different, unrelated keys (collusion).
- Look at the distribution of $c^t \oplus c^u$:
  - If flat, $\pi^t \neq \pi^u$
  - If one peak, then $\pi^t = \pi^u$, and get $p^t \oplus p^u$

Ciphertext Only Attack

Main idea

If the state (s, π) collides, then the output stream σ is the same.
Note that s depend on the key, but $\pi = \bigoplus p^i$
Consider two positions t, u and a random key:

$$\Pr_K [c^t \oplus c^u = X] \approx \begin{cases} 2^{-8} & \text{if } \pi^t \neq \pi^u \\ 2^{-8} + \Pr\left[s^t = s^{t'}\right] & \text{if } \pi^t = \pi^u, X = p^t \oplus p^u \end{cases}$$

$c^t \oplus c^u = \sigma^t \oplus p^t \oplus \sigma^u \oplus p^u$

- Consider several copies of a given text, encrypted with different, unrelated keys (collusion).
- Look at the distribution of $c^t \oplus c^u$:
  - If flat, $\pi^t \neq \pi^u$
  - If one peak, then $\pi^t = \pi^u$, and get $p^t \oplus p^u$

Ciphertext Only Attack

Main idea

If the state (s, π) collides, then the output stream σ is the same.
Note that s depend on the key, but $\pi = \bigoplus p^i$
Consider two positions t, u and a random key:

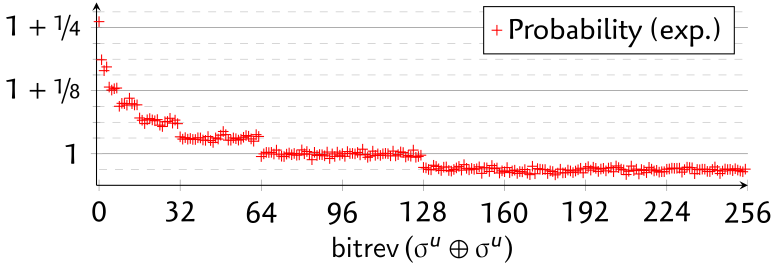
$$\Pr_K [c^t \oplus c^u = X] \approx \begin{cases} 2^{-8} & \text{if } \pi^t \neq \pi^u \\ 2^{-8} + \Pr\left[s^t = s^{t'}\right] & \text{if } \pi^t = \pi^u, X = p^t \oplus p^u \end{cases}$$

$c^t \oplus c^u = \sigma^t \oplus p^t \oplus \sigma^u \oplus p^u$

- Consider several copies of a given text, encrypted with different, unrelated keys (collusion).
- Look at the distribution of $c^t \oplus c^u$:
  - If flat, $\pi^t \neq \pi^u$
  - If one peak, then $\pi^t = \pi^u$, and get $p^t \oplus p^u$

Tricks to Improve the Bias

1. There are similar bias with the low bits of σ:



Use bias in low bit of $c^t \oplus c^u$: if $\pi^t = \pi^u$ and $X \equiv p^t \oplus p^u \bmod 2$, then

$$\Pr_K\left[c^t \oplus c^u \equiv X \bmod 2\right] \approx 2^{-1} + \Pr\left[s^t \equiv s^{t'} \bmod 2^9\right]$$

2. Use positions with $t \equiv u \bmod 8$
   - This gives a bias of 2^{-6} to 2^{-4} (cancellations in the state update)

Clustering algorithm

Finding relations

- Look at the distribution of $c^t \oplus c^u$ mod 2:
  - If flat, then $\pi^t \neq \pi^u$
  - If one peak, then $\pi^t = \pi^u$

- Use a clustering algorithm to recover π^t:
  - Initially, all positions are assigned a different *color*.
  - When $\pi^t = \pi^u$ is detected, merge colors.
- Easier to detect bias with larger clusters
  - Combine the biases $c^{t_i} \oplus c^{t_j}$

- At the end, 256 colors correspond to the 256 values of π^t
  - Recover the value of π^t using some known plaintext.
  - Recover p.

- Practical with 2^{10} keys, and 2^{17} data

Conclusion

Don't use an untested cipher!

Attacks on PC1		Complexity	Data	Ref.
Dist.	Chosen plaintext	2^{16}	2^{16}	Usenet
Key rec.	Known plaintext	2^{72}	2^{4}	Usenet
Key rec.	Known plaintext	2^{31}	2^{20}	New
Key rec.	Ciphertext only, 2^{10} unrelated keys	2^{35}	2^{17} per key	New
Attacks on PSCHF		**Complexity**		**Ref.**
2^{nd} pre.	with meaningful messages	2^{24}		New

Impact for the Kindle?

Pirates can just extract the key…
They don't need to break the cipher to break the DRM scheme.