Cryptanalysis of TWOPRIME

Don Coppersmith
IBM Research
copper@watson.ibm.com

David Wagner
University of California, Berkeley
daw@cs.berkeley.edu

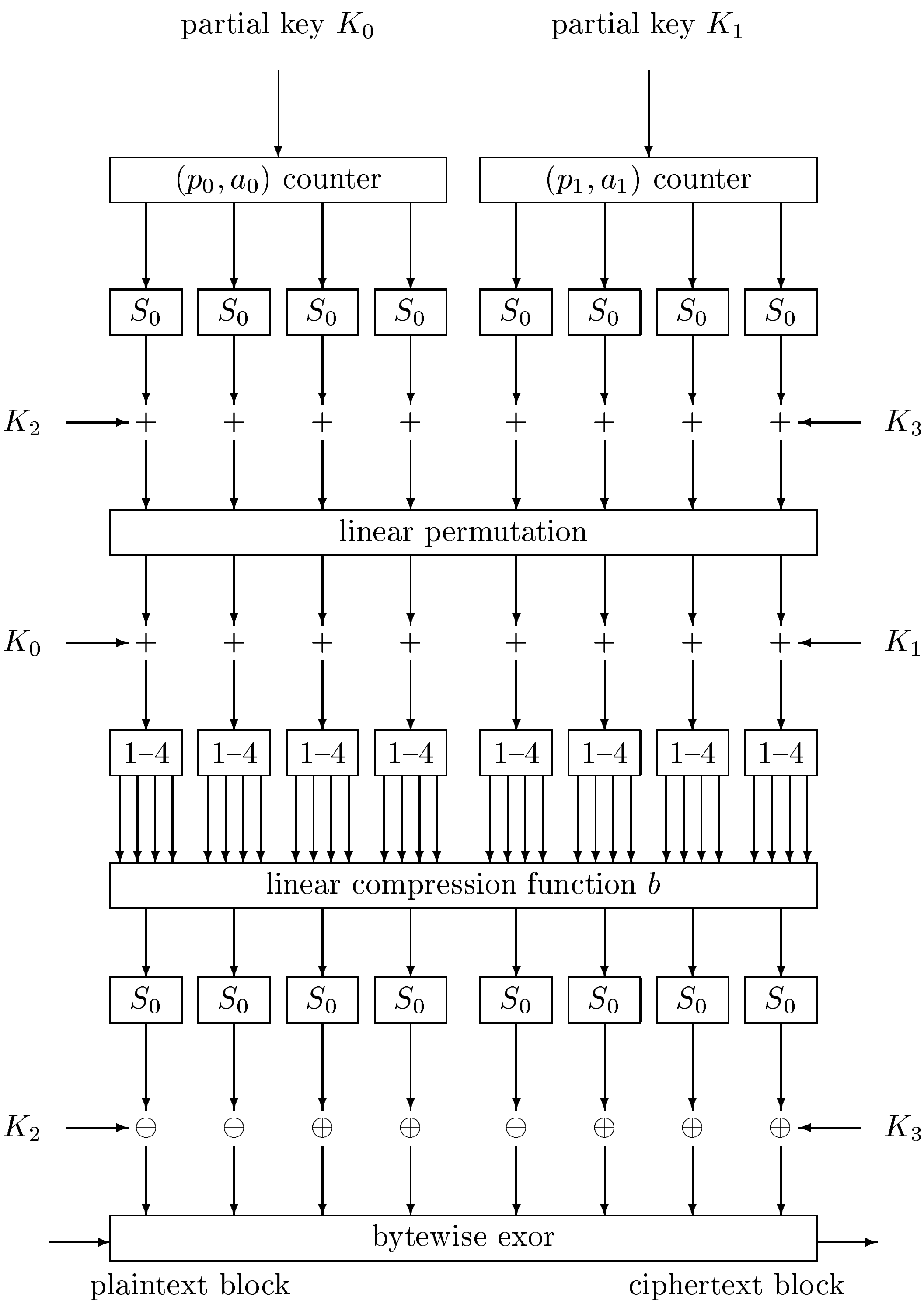
Bruce Schneier
Counterpane Systems
schneier@counterpane.com

John Kelsey
Counterpane Systems
kelsey@counterpane.com

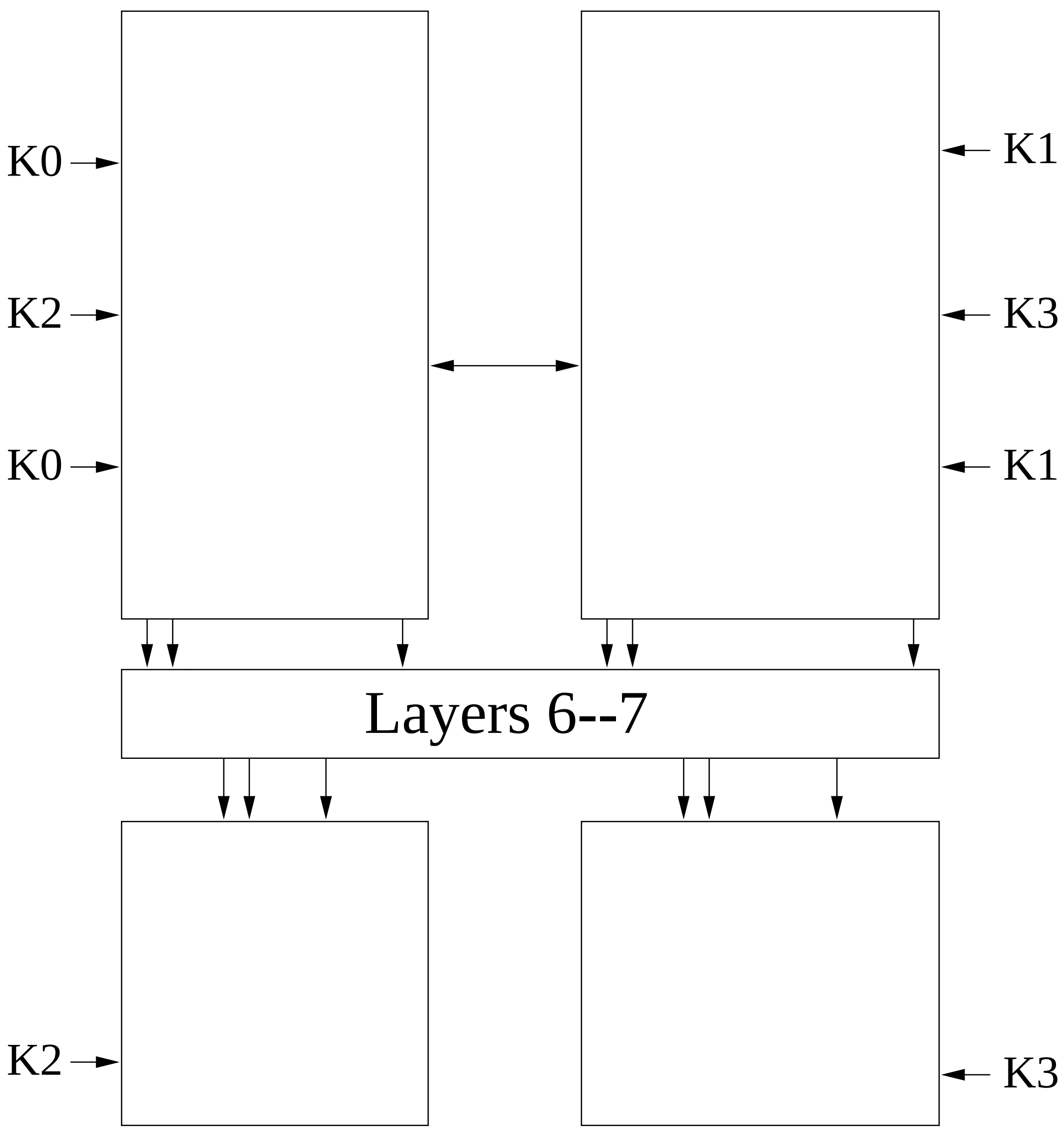
Organization

- The TWOPRIME cipher.
- Three ways to get K_2, K_3.
- Two ways to get K_0, K_1.
- Putting them together into full attacks.
- Variants of TWOPRIME.

TWOPRIME


partial key K_0
partial key K_1
(p_0, a_0) counter
(p_1, a_1) counter
S_0
K_2
K_3
linear permutation
K_0
K_1
1–4
linear compression function b
bytewise exor
plaintext block
ciphertext block

High-level view



Non-surjectivity

The seventh round transforms $(X_0, \ldots, X_{31})$ to $(Y_0, \ldots, Y_7) = b(X_0, \ldots, X_{31})$ by

$$\left\{\begin{array}{lcl}
Y_0 & = & X_0 + X_5 + X_{10} + X_{15} + X_{16} + X_{22} + X_{24} + X_{30}, \\
Y_1 & = & X_1 + X_6 + X_{11} + X_{12} + X_{17} + X_{23} + X_{25} + X_{31}, \\
Y_2 & = & X_2 + X_7 + X_8 + X_{13} + X_{18} + X_{20} + X_{26} + X_{28}, \\
Y_3 & = & X_3 + X_4 + X_9 + X_{14} + X_{19} + X_{21} + X_{27} + X_{29}, \\
Y_4 & = & X_{16} + X_{21} + X_{26} + X_{31} + X_0 + X_6 + X_8 + X_{14}, \\
Y_5 & = & X_{17} + X_{22} + X_{27} + X_{28} + X_5 + X_{11} + X_{13} + X_3, \\
Y_6 & = & X_{18} + X_{23} + X_{24} + X_{29} + X_{10} + X_{12} + X_2 + X_4, \\
Y_7 & = & X_{19} + X_{20} + X_{25} + X_{30} + X_{15} + X_1 + X_7 + X_9.
\end{array}\right.$$

We have the linear relation

$$Y_0 + Y_1 + Y_2 + Y_3 = Y_4 + Y_5 + Y_6 + Y_7 \bmod 256.$$

In other words, the seventh round is non-surjective.

We'll use this below to discover K_2, K_3 (in three different ways).

Getting K_2, K_3: Linear equations

Notation: $(k_0, \ldots, k_7) = (K_2, K_3)$. Keystream is $z_1, z_2, \ldots$.

For $i < 8, j < 256$, let

$$y_{i,j} = S0^{-1}(j \oplus k_i)$$

represent the result of decrypting the i-th keystream byte j up to the output of the seventh layer.

With 2048 known texts, we get 2048 linear equations

$$\sum_{i=0}^{3} y_{i,z_n} = \sum_{i=4}^{7} y_{i,z_n} \bmod 256.$$

Solve for the $y_{i,j}$'s. Recover k_i's.

Complexity: 2048 known texts, minimal work.

Getting K_2, K_3: Meet in the middle

Naive: Guess K_2, K_3. Decrypt each block of known keystream up to the output $Y_0, \ldots, Y_7$ of the seventh layer. Use unattainable values to discard incorrect guesses at K_2, K_3.

Better: For each K_2, decrypt keystream block up left half to obtain $Y_0, \ldots, Y_3$, and make a table with entries

$$(Y_0 + \ldots + Y_3), K_2.$$

For each K_3, decrypt up right side to obtain $Y_4, \ldots, Y_7$, and make a table with entries

$$(Y_4 + \ldots + Y_7), K_3.$$

Look for matches in the two tables.

Complexity: 8 known texts, 2^{32} work, 2^{32} space.

Getting K_2, K_3: Splitting the period

Same idea as classical attacks on two-loop Vigenere ciphers.

Look at four time steps:

$$a = t_0, \qquad b = t_0 + p_0$$
$$c = t_0 + p_1, \qquad d = t_0 + p_0 + p_1.$$

On left side, encryptions at time t and $t + p_0$ are the same. Right side: t and $t + p_1$.

The inputs X to the seventh layer satisfy

$$X_i[a] + X_i[d] = X_i[b] + X_i[c] \bmod 256.$$

Since the seventh layer is linear, its outputs satisfy

$$Y_i[a] + Y_i[d] = Y_i[b] + Y_i[c] \bmod 256.$$

Can find K_2, K_3 easily from this.

Getting K_2, K_3: Splitting the period

That doesn't quite work as stated, because there is diffusion from left to right (and right to left) in fourth layer.

But this diffusion channel is only 8 bits wide. Therefore, we can control it: attack will still work for 1 in 2^{16} choices of t_0.

Try $220,000$ values of t_0. Use a counting scheme.

Complexity: 2^{33} known texts, 2^{28} work. Works for any other (linear) seventh layer.

Getting K_0, K_1: Splitting the period, again

Given K_2, K_3, can identify when the diffusion channel in fourth layer was favorable.

Note that left-to-right diffusion channel depends only on K_0 and K_2. Given two time steps a, c where this channel collides, we can learn 8 bits of information on K_0.

Previous attack gives us several such collisions. Do exhaustive search for K_0.

Complexity: 2^{32} work.

Getting K_0, K_1: Meet in the middle, again

Notation. X: input to sixth layer. Y: output of seventh layer.

Recover X given Y in time 2^{24}:

> $Y_0 + Y_2 - Y_4 - Y_6$ depends only on half of X.
>
> $Y_0 + Y_2 - Y_5 - Y_7$ depends only on other half of X.
>
> Generate tables; then look for matching entries.

Find K_0 given X in time 2^{32}:

> Compute backwards from X to output of fourth layer.
>
> Compute forwards from K_0 to input of fourth layer.
>
> If incompatible, discard guess at K_0.

Get K_1 similarly.

Complexity: 2 known texts, 2^{32} work, 2^{24} space.

Summary of attacks

Attack techniques $K_2, K_3 + K_0, K_1$	Complexity Texts	Time	Space
MITM + MITM	8	2^{32}	2^{32}
Linear eqns + MITM	2048	2^{32}	$O(1)$
Split the period, twice	2^{33}	2^{28}	$O(1)$

Third attack will apply even if seventh layer is changed.

Variants

TWOPRIME-1:

- Uses different seventh layer.
  (It preserves halves.)
- Same attacks work; some are even easier.

ONEPRIME:

- Uses one 64-bit counter.
- Same attacks work. Not much harder.

Conclusion: the variants are no stronger.

Conclusions

- Can break TWOPRIME with 2^{32} complexity.
- Too few rounds for serious strength.
  - Weaknesses in only two layers enough to break the cipher.
- Attacks work by splitting cipher horizontally, vertically.
  - Avoid introducing non-surjectivity in the middle of a cipher.
- Lots of linearity aids the analyst, as always.