Cryptanalysis of SPEED

(Extended Abstract)

Chris Hall[1], John Kelsey[1], Bruce Schneier[1], and David Wagner[2]

[1] Counterpane Systems
101 E. Minnehaha Pkwy
Minneapolis, MN 55419
(612) 823-1098
{hall,kelsey,schneier}@counterpane.com
[2] U.C. at Berkeley
Soda Hall
Berkeley, CA 94720-1776
daw@cs.berkeley.edu

Abstract. The cipher family SPEED (and an associated hashing mode) was recently proposed in *Financial Cryptography '97*. In cryptanalyzing the cipher we found several troubling potential weaknesses. Next, we were able to efficiently break the SPEED hashing mode using differential related-key techniques. Finally, we examined differential attacks against the 48-round version of SPEED. These results raise some significant questions about the security of the SPEED design.

1 Introduction

In *Financial Cryptography '97*, Zheng proposed a new family of block ciphers, called SPEED [1]. One specifies a particular SPEED cipher by choosing parameters such as the block size and number of rounds; the variations are otherwise alike in their key schedule and round structure. Under the hood, SPEED is built out of an unbalanced Feistel network. Zheng also proposed a hash function based on running a SPEED block cipher in a slightly modified Davies-Meyer mode.

One of the main contributions of the SPEED design is its prominent use of carefully chosen Boolean functions which can be shown to have very good non-linearity, as well as other desirable theoretical properties. One might therefore hope that SPEED rests on a solid theoretical foundation in cryptographic Boolean function theory. Nonetheless, we have found serious weaknesses in the cipher; many lead to practical attacks on SPEED.

In examining the cipher there appears to be an obvious 1-bit differential attack which works with probability 2^{-50} against the 48-round version of the cipher. However, our analysis indicates that this attack may in fact fail to work. A future paper will address the strength of SPEED against differential cryptanalysis in greater detail.

Despite our difficulties with the differential attack, we succeeded in finding collisions for the SPEED hash function. For the 128-bit hash with 32 rounds, we found the following collision (in base-16):

$$
\begin{array}{ll}
M = \texttt{21EA FE8E 1637 19F7 22D2 8CCB} & M\texttt{'} = \texttt{21EA FE8E 1637 19F7 22D2 8CCB} \\
\quad\; \texttt{3724 3437 B00F 7607 3C91 3710} & \quad\;\; \texttt{3724 3437 B00F 7607 3C91 3710} \\
\quad\; \texttt{2B69 C9C9 58FB 0823 AEC2 CD05} & \quad\;\; \texttt{2B69 C9C9 58FB 0823 AEC2 CD05} \\
\quad\; \underline{\texttt{FD80}}\ \texttt{14E6 B11E}\ \underline{\texttt{43C0}}\ \texttt{5767 76F7} & \quad\;\; \underline{\texttt{FDC0}}\ \texttt{14E6 B11E}\ \underline{\texttt{4380}}\ \texttt{5767 76F7} \\
\quad\; \texttt{FF07 17EC}\ \underline{\texttt{FCBA}}\ \texttt{224E 9627}\ \underline{\texttt{A16A}} & \quad\;\; \texttt{FF07 17EC}\ \underline{\texttt{7CBA}}\ \texttt{224E 9627}\ \underline{\texttt{216A}} \\
\quad\; \texttt{8D6E 83A9} & \quad\;\; \texttt{8D6E 83A9}
\end{array}
$$

This leads to the following values when hashing (in base-16):

$$
\begin{aligned}
D_0 &= \texttt{0000 0000 0000 0000 0000 0000 0000 0000} \\
D_1 &= \texttt{90DA 7F34 46FA A373 B048 11F7 F8D9 BB3D} \\
D_2 - D_1 &= \texttt{9781 9517 B5CC A046 D0F1 3719 ED9B A0B6}
\end{aligned}
$$

We also found the following collision for the 128-bit hash with 48 rounds (in base-16):

$$
\begin{array}{ll}
M = \texttt{3725 6571 48D5 CF52 DAE1 4065} & M\texttt{'} = \texttt{3725 6571 48D5 CF52 DAE1 4065} \\
\quad\; \texttt{7115 11A0 E3C5 9428 7BFD 18CB} & \quad\;\; \texttt{7115 11A0 E3C5 9428 7BFD 18CB} \\
\quad\; \texttt{EF79 82BB 1D7F 2F55}\ \underline{\texttt{36F2}}\ \underline{\texttt{CD58}} & \quad\;\; \texttt{EF79 82BB 1D7F 2F55}\ \underline{\texttt{38F2}}\ \underline{\texttt{CB58}} \\
\quad\; \texttt{9058}\ \underline{\texttt{FE57}}\ \underline{\texttt{D696}}\ \texttt{EA4C BD75 F7C9} & \quad\;\; \texttt{9058}\ \underline{\texttt{FC57}}\ \underline{\texttt{D896}}\ \texttt{EA4C BD75 F7C9} \\
\quad\; \underline{\texttt{1989}}\ \underline{\texttt{A048}}\ \texttt{39FB}\ \underline{\texttt{9B76}}\ \underline{\texttt{9011}}\ \texttt{CAC0} & \quad\;\; \underline{\texttt{1985}}\ \underline{\texttt{A04C}}\ \texttt{39FB}\ \underline{\texttt{9B7A}}\ \underline{\texttt{900D}}\ \texttt{CAC0} \\
\quad\; \texttt{65F6 EBC7} & \quad\;\; \texttt{65F6 EBC7}
\end{array}
$$

This leads to the following values when hashing (in base-16):

$$
\begin{aligned}
D_0 &= \texttt{0000 0000 0000 0000 0000 0000 0000 0000} \\
D_1 &= \texttt{DA2B A119 A4F8 AA70 59ED 6FE4 188B 7969} \\
D_2 - D_1 &= \texttt{CAB1 DA86 B6D3 1442 E05C A005 7B26 C432}
\end{aligned}
$$

2 Conclusions

It is interesting to note that SPEED, though built using very strong component functions, doesn't appear to be terribly secure. The SPEED design apparently relied upon the high quality of the binary functions used, the fact that different functions were used at different points in the cipher, and the data-dependent rotations to provide resistance to cryptanalysis. Unfortunately, the most effective attacks aren't made much less powerful by any of these defenses.

Due to these weaknesses, we would recommend against using SPEED for high security applications. It's not clear whether or not someone could design a security cipher using the same sort of boolean function theory. Therefore the utility of these functions in cipher design is still an open avenue of research.

References

1. Y. Zheng, "The SPEED Cipher," in Proceedings of *Financial Cryptography '97*, Springer-Verlag.