Multiple Results on Multiple Encryption

Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir
The Security of Multiple Encryption:

- Given a block cipher with $n$-bit plaintexts and $n$-bit keys, we would like to enhance its security via **sequential composition**.

- Assuming that
  - the basic block cipher has **no weaknesses**
  - the $k$ keys are independent and chosen.

How secure is the resultant composition?
Double and Triple Encryptions:

- Double DES and triple DES were widely used by banks, so their security was thoroughly analyzed.

- By using a Meet in the Middle (MITM) attack, Diffie and Hellman showed in 1981 that double encryption can be broken in time $T=2^n$ and space $S=2^n$. Note that $TS=2^{2n}$.

- Given the same amount of space $S=2^n$, we can break triple encryption in time $T=2^{2n}$, so again $TS=2^{3n}$. 
How Secure is $k$-encryption for $k>3$?

- The fun really starts at quadruple encryption ($k=4$), which was not well studied so far, since we can show that breaking $4$-encryption is not harder than breaking $3$-encryption when we use $2^n$ space!

- Our new attacks:
  - use the smallest possible amount of data ($k$ known plaintext/ciphertext pairs which are required to uniquely define the $k$ keys)
  - Never err (if there is a solution, it will always be found)
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)

\[
\begin{array}{c|c}
  k &= 2 \\
  c &= 1 \\
\end{array}
\]
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T = 2^{cn}$)

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k =$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c =$</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)

<table>
<thead>
<tr>
<th>$k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)

<table>
<thead>
<tr>
<th>$k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
The time complexity of our new attacks (expressed by the coefficient \( c \) in the time formula \( T = 2^{cn} \))

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>k</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)

<table>
<thead>
<tr>
<th>$k =$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c =$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
The time complexity of our new attacks (expressed by the coefficient $c$ in the time formula $T=2^{cn}$)

<table>
<thead>
<tr>
<th>$k$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c$</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>
The “Magic Numbers” of rounds:

- We gain some time at each magic number, and the savings accumulate as k increases.

- There is an infinite number of magic numbers, starting with $k=4, 7, 11, 16, 22, 29, 37, 46, 56,...$ which grow quadratically.

- We can prove the optimality of our new attacks within a broad class of possible algorithms which we call Dissection Attacks.
Using the New Techniques to Solve Non-cryptographic Combinatorial Search Problems

- Consider for example the knapsack problem of finding a 0/1 solution for
  \[ x_1*a_1 + x_2*a_2 + x_3*a_3 + \ldots + x_n*a_n = v \]

- We can represent the knapsack problem as a \( k \)-encryption problem for any desired \( k \).
Example: Given the 6 generators \( a_1, \ldots, a_6 \), we describe the knapsack problem of representing the number \( v \) as a triple encryption with the three independent 2-bit keys \((x_1, x_2), (x_3, x_4), (x_5, x_6)\)

Starting with plaintext \( P=0 \), we first add to it \( x_1* a_1 + x_2* a_2 \) to get the first ciphertext. We then encrypt it a second time by adding to it \( x_3* a_3 + x_4* a_4 \), and finally encrypt it a third time by adding to it \( x_5* a_5 + x_6* a_6 \) to get the final ciphertext \( C=v \)
Using the New Techniques to Solve Non-cryptographic Combinatorial Search Problems

- The knapsack problem can thus be described as the problem of finding the $k$ keys of $n/k$ bits each that map the initial plaintext $0$ to the final ciphertext $v$

- By using our new 7-encryption attack, we can solve hard knapsack problems in time $T=2^{4n/7}$ and space $S=2^{n/7}$

- This is a faster attack than the best previously published knapsack solving algorithm (by Becker, Coron, Joux) for such a small memory complexity
Concluding Remarks:

- Breaking multiple encryption is much easier than previously believed.

- Many combinatorial search problems can be described as k-encryption problems, and then solved more efficiently by our new generic techniques.