A PROBABILISTIC MODEL TO EVALUATE COMPUTER SOFTWARE PIRACY

R.J. Whitley(1) and T.V. Hromadka II(2)

Abstract

The growing occurrence of computer software piracy has led to a new area of research, i.e., the development of methods to be used to supply evidence that software was copied.

One method to argue that computer source code was copied is to examine the occurrence of strings of binary code (ones and zeroes) between the alleged parent and pirate codes. Given the occurrence of a lengthy identical string between codes, and that string represents a development of executable code (versus data blocks that can be argued to exist in only one fashion), a model of the probability of repetition of such a string of code occurring between so-called independently derived source codes can be formulated. The developed probabilistic results can also be approximated by a simpler formula derived herein. A computer program and example computations are presented.

(1) Professor, Department of Mathematics, University of California, Irvine, CA 92717

(2) Associate Professor, Department of Mathematics, California State University, Fullerton, CA 92634
A probabilistic model to evaluate computer software piracy

R. J. WHITLEY

Department of Mathematics, University of California, Irvine, CA 92717, U.S.A.

T. V. HROMADKA II

Department of Mathematics, California State University, Fullerton, CA 92634, U.S.A.

The growing occurrence of computer software piracy has led to a new area of research, i.e., the development of methods to be used to supply evidence that software was copied.

One method to argue that computer source code was copied is to examine the occurrence of strings of binary code (ones and zeroes) between the alleged parent and pirate codes. Given the occurrence of a lengthy identical string between codes, and that string represents a development of executable code (versus data blocks that can be argued to exist in only one fashion), a model of the probability of repetition of such a string of code occurring between so-called independently derived source codes can be formulated. The developed probabilistic results can also be approximated by a simpler formula derived herein. A computer program and example computations are presented.

INTRODUCTION

The growing occurrence of computer software piracy has led to a new area of research, i.e., the development of methods to be used to supply evidence that software was copied. This problem is difficult due to the argument that the parent software was developed by a knowledgeable person and hence the resulting code should be a more probable outcome than another statement of code. Another argument in defense is that there can only be a finite number of ways to write a segment of code.

One method to argue code was copied is to examine the occurrence of strings of binary code (ones and zeroes) between the alleged parent and pirate codes. Given the occurrence of a lengthy identical string between codes, and that string represents a development of executable code (versus data blocks that can be argued to exist in only one fashion), a model of the probability of repetition of such a string of code occurring between so-called independently derived source codes can be formulated. The developed probabilistic results can also be approximated by a simpler formula derived herein. A computer program and example computations are presented.

PROBABILISTIC MODEL

Consider tossing a fair coin. For a number \( m \) let \( p_n \) denote the probability that in \( n \) tosses there will be at least one run of heads of length \( m \). To compute this probability is the same as computing the number \( H_n \) of different outcomes of \( n \) tosses which contain at least one run of \( m \) heads, since:

\[
p_n = (1/2^n)H_n
\]  

It is useful to also consider the set \( E_n \) of all outcomes of \( n \) tosses in which at least one run of \( m \) heads occurs; \( E_n \) has \( H_n \) members. The integer \( m \) is regarded as fixed.
simplifying the notation which would otherwise be, say, $p_{n,m}$ and $H_{n,m}$.

A recursive formula can be derived by considering the events in $E_{n+1}$ and looking at two cases.

Case 1 is when the event of a run of $m$ heads has already occurred in the first $n$ trials. Letting $X_i$ be $H$ or $T$ depending on the results of the $i$-th toss, case 1 has the form $X_1X_2\ldots X_nX_{n+1}$ where $X_1X_2\ldots X_n$ is an event in $E_n$ and $X_{n+1}$ is either $H$ or $T$, there are $2H_n$ events in case 1.

Case 2 is the event, disjoint from case 1, in which there is no run of heads of length $m$ in the first $n$ trials but there is such a run in $n+1$ trials. For this to occur, the event must have the form
\[X_1X_2\ldots X_{n-m}THHH\ldots H\]

where the sequence of the last $m+1$ outcomes begins with a $T$ and is followed by $m$ $H$s. Also, for the outcome in equation (2) to belong to case 2 there cannot be a run of $m$ heads in the entries $X_1X_2\ldots X_{n-m}$; i.e., $X_1X_2\ldots X_{n-m}$ does not belong to $E_{n-m}$. Thus, there are
\[2^n - m - H_{n-m}\]
different events in case 2.

The set $E_{n+1}$ consists of the disjoint union of case 1 and case 2, so
\[H_{n+1} = 2H_n + 2^n - m - H_{n-m}\]

Dividing by $2^n+1$ and using equation (1)
\[p_{n+1} = p_n + \frac{(1 - p_{n-m})}{2^n+1}\]

This recursively defines, $p_{n+1}$, beginning with
\[p_j = 0, j < m \text{ and } p_m = 1/2^m\]

For large $m$, some ideas of the size of the numbers involved can be obtained by considering another recursively defined sequence. Consider $z_n$ defined by
\[z_i = p_i, j = 0, 1, \ldots, 2m - 1\]
\[z_{n+1} = z_n + 1/2^{n+1}\]

Since
\[p_{2m-1} = (m + 1)/2^{m+1}\]
\[z_n = (n - m + 2)/2^{m+1} \text{ for } n \geq 2m\]

By construction,
\[z_n \geq p_n \text{ for all } n\]

Given a value $c$, $0 < c < 1$, the value of $n$ such that
\[z_n \geq c\]
is
\[n = (2^{m+1} + m - 2)\]

So $n$ will have to be at least this large, for $n \geq 2m$, to have $p_n \geq c$. For example, suppose you want to take $n$ large enough so that the probability of 100 consecutive heads (or a string of size 100) will be at least $0.5$; the smallest $n$ that will do will be at least as large as the $n$ for which $q_n \geq 0.5$, i.e.,
\[n = \frac{5}{2^{100}} + 98 = 1.27 \times 10^{30}\]

It is possible to obtain a very accurate computable approximate formula for the $p_n$ by arguing as follows: with the change of variable
\[q_n = 1 - p_n\]
equation (5) becomes the homogeneous difference equation
\[q_{n+1} = q_n - q_{n-m}/2^{n+1}\]

Substituting $q_n = x^n$ in this equation gives the related equation
\[f(x) = x^n - x + 1/2^{n+1} = 0\]

The solution $q_n$ is a linear combination of $n$-th powers of roots of equation (16); for an approximate formula it is hoped that for large $n$ the largest root will give the dominant term.

To approximate the largest root of equation (16), note that
\[f(1) = 1/2^{n+1}\]
is quite small for large $n$ and so $x_0 = 1$ would be a good initial guess at this root. Apply Newton’s method for a better guess
\[x_i = x_0 - f(x_0)/f'(x_0) = 1 - 1/2^{n+1}\]

This leads to the approximation
\[\hat{p}_n = 1 - (1/2^{n+1})^c\]

This approximation does not satisfy any of the boundary conditions of equation (6), but, as we will see below, the derived formula (20) is quite accurate.

Given $0 < c < 1$, the first value of $n$ for which $\hat{p}_n \geq c$ is
\[n = \ln(1 - c)/\ln(1 - 1/2^{n+1}) \approx 2^{n+1}(-\ln(1 - c))\]

For example, for $m = 100$, and $c = 0.5$, $n = 1.7 \times 10^{30}$; compare this with the lower bound given above.

The table below gives for $c = 0.1, 0.5$ and $n = 4(1)20$.

(a) The value of the first $n$ for which $\hat{p}_n \geq c$, computed from the recursion (10).
(b) The value of $n$ computed from the right hand side of equation (20).
(c) The lower bound of equation (13).

**REFERENCE**

\[ m = 4 \]

\[ \begin{align*}
\text{pn} &= 0.12500000 \quad \text{n} = 6 \quad \text{approx. n} = 3 \quad \text{bound} = 5 \\
\text{pn} &= 0.21679688 \quad \text{n} = 9 \quad \text{approx. n} = 7 \quad \text{bound} = 8 \\
\text{pn} &= 0.32421875 \quad \text{n} = 13 \quad \text{approx. n} = 11 \quad \text{bound} = 11 \\
\text{pn} &= 0.41693115 \quad \text{n} = 17 \quad \text{approx. n} = 16 \quad \text{bound} = 14 \\
\text{pn} &= 0.51514530 \quad \text{n} = 22 \quad \text{approx. n} = 22 \quad \text{bound} = 17 
\end{align*} \]

\[ m = 5 \]

\[ \begin{align*}
\text{pn} &= 0.10937500 \quad \text{n} = 15 \quad \text{approx. n} = 7 \quad \text{bound} = 9 \\
\text{pn} &= 0.21021271 \quad \text{n} = 17 \quad \text{approx. n} = 14 \quad \text{bound} = 15 \\
\text{pn} &= 0.31159061 \quad \text{n} = 25 \quad \text{approx. n} = 23 \quad \text{bound} = 22 \\
\text{pn} &= 0.41017180 \quad \text{n} = 34 \quad \text{approx. n} = 33 \quad \text{bound} = 28 \\
\text{pn} &= 0.50324029 \quad \text{n} = 44 \quad \text{approx. n} = 44 \quad \text{bound} = 34 
\end{align*} \]

\[ m = 6 \]

\[ \begin{align*}
\text{pn} &= 0.10034180 \quad \text{n} = 17 \quad \text{approx. n} = 13 \quad \text{bound} = 16 \\
\text{pn} &= 0.20497159 \quad \text{n} = 32 \quad \text{approx. n} = 29 \quad \text{bound} = 29 \\
\text{pn} &= 0.30319989 \quad \text{n} = 48 \quad \text{approx. n} = 46 \quad \text{bound} = 42 \\
\text{pn} &= 0.40420780 \quad \text{n} = 67 \quad \text{approx. n} = 65 \quad \text{bound} = 55 \\
\text{pn} &= 0.50301594 \quad \text{n} = 89 \quad \text{approx. n} = 89 \quad \text{bound} = 67 
\end{align*} \]

\[ m = 7 \]

\[ \begin{align*}
\text{pn} &= 0.10260123 \quad \text{n} = 32 \quad \text{approx. n} = 27 \quad \text{bound} = 30 \\
\text{pn} &= 0.20149082 \quad \text{n} = 61 \quad \text{approx. n} = 57 \quad \text{bound} = 56 \\
\text{pn} &= 0.30083370 \quad \text{n} = 94 \quad \text{approx. n} = 91 \quad \text{bound} = 81 \\
\text{pn} &= 0.40061751 \quad \text{n} = 132 \quad \text{approx. n} = 131 \quad \text{bound} = 107 \\
\text{pn} &= 0.50144943 \quad \text{n} = 178 \quad \text{approx. n} = 177 \quad \text{bound} = 132 
\end{align*} \]

\[ m = 8 \]

\[ \begin{align*}
\text{pn} &= 0.10156655 \quad \text{n} = 60 \quad \text{approx. n} = 54 \quad \text{bound} = 57 \\
\text{pn} &= 0.20094174 \quad \text{n} = 119 \quad \text{approx. n} = 114 \quad \text{bound} = 108 \\
\text{pn} &= 0.30051364 \quad \text{n} = 186 \quad \text{approx. n} = 183 \quad \text{bound} = 159 \\
\text{pn} &= 0.40091199 \quad \text{n} = 264 \quad \text{approx. n} = 262 \quad \text{bound} = 210 \\
\text{pn} &= 0.50097240 \quad \text{n} = 356 \quad \text{approx. n} = 355 \quad \text{bound} = 261 
\end{align*} \]

\[ m = 9 \]

\[ \begin{align*}
\text{pn} &= 0.10006459 \quad \text{n} = 114 \quad \text{approx. n} = 108 \quad \text{bound} = 109 \\
\text{pn} &= 0.20046290 \quad \text{n} = 234 \quad \text{approx. n} = 228 \quad \text{bound} = 211 \\
\text{pn} &= 0.30008661 \quad \text{n} = 369 \quad \text{approx. n} = 365 \quad \text{bound} = 314 \\
\text{pn} &= 0.40044137 \quad \text{n} = 526 \quad \text{approx. n} = 523 \quad \text{bound} = 416 \\
\text{pn} &= 0.50038894 \quad \text{n} = 711 \quad \text{approx. n} = 710 \quad \text{bound} = 518 
\end{align*} \]

\[ m = 10 \]

\[ \begin{align*}
\text{pn} &= 0.10013437 \quad \text{n} = 223 \quad \text{approx. n} = 216 \quad \text{bound} = 212 \\
\text{pn} &= 0.20012739 \quad \text{n} = 463 \quad \text{approx. n} = 457 \quad \text{bound} = 417 \\
\text{pn} &= 0.30008859 \quad \text{n} = 735 \quad \text{approx. n} = 730 \quad \text{bound} = 622 \\
\text{pn} &= 0.40005301 \quad \text{n} = 1049 \quad \text{approx. n} = 1046 \quad \text{bound} = 827 \\
\text{pn} &= 0.50017293 \quad \text{n} = 1421 \quad \text{approx. n} = 1420 \quad \text{bound} = 1031 
\end{align*} \]

\[ m = 11 \]

\[ \begin{align*}
\text{pn} &= 0.10014117 \quad \text{n} = 440 \quad \text{approx. n} = 432 \quad \text{bound} = 418 \\
\text{pn} &= 0.20010925 \quad \text{n} = 921 \quad \text{approx. n} = 914 \quad \text{bound} = 828 \\
\text{pn} &= 0.30002588 \quad \text{n} = 1466 \quad \text{approx. n} = 1461 \quad \text{bound} = 1237 \\
\text{pn} &= 0.40007713 \quad \text{n} = 2096 \quad \text{approx. n} = 2092 \quad \text{bound} = 1647 \\
\text{pn} &= 0.50010228 \quad \text{n} = 2841 \quad \text{approx. n} = 2839 \quad \text{bound} = 2056 
\end{align*} \]
<table>
<thead>
<tr>
<th>( m )</th>
<th>( p_n )</th>
<th>( n )</th>
<th>( \text{approx. } n )</th>
<th>( \text{bound} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>0.10002148</td>
<td>872</td>
<td>863</td>
<td>829</td>
</tr>
<tr>
<td></td>
<td>0.20007727</td>
<td>1836</td>
<td>1828</td>
<td>1648</td>
</tr>
<tr>
<td></td>
<td>0.30004899</td>
<td>2928</td>
<td>2922</td>
<td>2467</td>
</tr>
<tr>
<td></td>
<td>0.40005115</td>
<td>4189</td>
<td>4185</td>
<td>3286</td>
</tr>
<tr>
<td></td>
<td>0.50002444</td>
<td>5680</td>
<td>5678</td>
<td>4105</td>
</tr>
<tr>
<td>13</td>
<td>0.10001049</td>
<td>1736</td>
<td>1726</td>
<td>1649</td>
</tr>
<tr>
<td></td>
<td>0.20000118</td>
<td>3664</td>
<td>3656</td>
<td>3287</td>
</tr>
<tr>
<td></td>
<td>0.30000111</td>
<td>5850</td>
<td>5844</td>
<td>4926</td>
</tr>
<tr>
<td></td>
<td>0.40001930</td>
<td>8374</td>
<td>8369</td>
<td>6564</td>
</tr>
<tr>
<td></td>
<td>0.50002542</td>
<td>11359</td>
<td>11357</td>
<td>8202</td>
</tr>
<tr>
<td>14</td>
<td>0.10000197</td>
<td>3463</td>
<td>3452</td>
<td>3288</td>
</tr>
<tr>
<td></td>
<td>0.20000648</td>
<td>7321</td>
<td>7312</td>
<td>6565</td>
</tr>
<tr>
<td></td>
<td>0.30001379</td>
<td>11695</td>
<td>11688</td>
<td>9842</td>
</tr>
<tr>
<td></td>
<td>0.40001230</td>
<td>16744</td>
<td>16739</td>
<td>13119</td>
</tr>
<tr>
<td></td>
<td>0.50000004</td>
<td>22715</td>
<td>22713</td>
<td>16395</td>
</tr>
<tr>
<td>15</td>
<td>0.10000995</td>
<td>6917</td>
<td>6905</td>
<td>6566</td>
</tr>
<tr>
<td></td>
<td>0.20000637</td>
<td>14634</td>
<td>14624</td>
<td>13120</td>
</tr>
<tr>
<td></td>
<td>0.30000511</td>
<td>23383</td>
<td>23375</td>
<td>19673</td>
</tr>
<tr>
<td></td>
<td>0.40000411</td>
<td>33483</td>
<td>33477</td>
<td>26227</td>
</tr>
<tr>
<td></td>
<td>0.50000496</td>
<td>45429</td>
<td>45426</td>
<td>32780</td>
</tr>
<tr>
<td>16</td>
<td>0.10000635</td>
<td>13823</td>
<td>13810</td>
<td>13121</td>
</tr>
<tr>
<td></td>
<td>0.20000494</td>
<td>29259</td>
<td>29248</td>
<td>26228</td>
</tr>
<tr>
<td></td>
<td>0.30000420</td>
<td>46759</td>
<td>46750</td>
<td>39335</td>
</tr>
<tr>
<td></td>
<td>0.40000225</td>
<td>66961</td>
<td>66955</td>
<td>52442</td>
</tr>
<tr>
<td></td>
<td>0.50000095</td>
<td>90855</td>
<td>90852</td>
<td>65549</td>
</tr>
<tr>
<td>17</td>
<td>0.10000074</td>
<td>27633</td>
<td>27620</td>
<td>26229</td>
</tr>
<tr>
<td></td>
<td>0.20000049</td>
<td>58507</td>
<td>58496</td>
<td>52443</td>
</tr>
<tr>
<td></td>
<td>0.30000012</td>
<td>93509</td>
<td>93500</td>
<td>78658</td>
</tr>
<tr>
<td></td>
<td>0.40000015</td>
<td>133916</td>
<td>133910</td>
<td>104872</td>
</tr>
<tr>
<td></td>
<td>0.50000144</td>
<td>181708</td>
<td>181704</td>
<td>131086</td>
</tr>
<tr>
<td>18</td>
<td>0.10000119</td>
<td>55254</td>
<td>55239</td>
<td>52444</td>
</tr>
<tr>
<td></td>
<td>0.20000098</td>
<td>117004</td>
<td>116991</td>
<td>104873</td>
</tr>
<tr>
<td></td>
<td>0.30000027</td>
<td>187010</td>
<td>187000</td>
<td>157302</td>
</tr>
<tr>
<td></td>
<td>0.40000081</td>
<td>267827</td>
<td>267820</td>
<td>209731</td>
</tr>
<tr>
<td></td>
<td>0.50000007</td>
<td>363412</td>
<td>363409</td>
<td>262159</td>
</tr>
<tr>
<td>19</td>
<td>0.10000047</td>
<td>110494</td>
<td>110479</td>
<td>104874</td>
</tr>
<tr>
<td></td>
<td>0.20000029</td>
<td>233996</td>
<td>233983</td>
<td>209732</td>
</tr>
<tr>
<td></td>
<td>0.30000011</td>
<td>374011</td>
<td>374001</td>
<td>314589</td>
</tr>
<tr>
<td></td>
<td>0.40000027</td>
<td>535647</td>
<td>535639</td>
<td>419447</td>
</tr>
<tr>
<td></td>
<td>0.50000001</td>
<td>726821</td>
<td>726817</td>
<td>524304</td>
</tr>
<tr>
<td>20</td>
<td>0.10000066</td>
<td>220973</td>
<td>220957</td>
<td>209733</td>
</tr>
<tr>
<td></td>
<td>0.20000024</td>
<td>467980</td>
<td>467966</td>
<td>419448</td>
</tr>
<tr>
<td></td>
<td>0.30000025</td>
<td>748013</td>
<td>748002</td>
<td>629163</td>
</tr>
<tr>
<td></td>
<td>0.40000014</td>
<td>1071287</td>
<td>1071279</td>
<td>838878</td>
</tr>
<tr>
<td></td>
<td>0.50000005</td>
<td>1453639</td>
<td>1453635</td>
<td>1048593</td>
</tr>
</tbody>
</table>
APPENDIX B. SOURCE CODE

Computes the probability \( p[n] \) of getting at least one run of heads of input length \( m \) in \( n \) trials of a fair coin. The values of \( n \) for which \( p[n] \) first exceeds \( 0.5 \) are printed. An approximate formula and a simple lower bound for \( n \), valid for \( n \) great than or equal to \( 2m \), are also printed.

```pascal
uses printer;
var
    p: array[0..50] of double;
    n: longint;
    k1, k2, k3, i, m: integer;
    c, L: double;
begin
    writeln('input m');
    readln(m);
    writeln(lst, 'm = ', m);
    c := exp(-(m+1)*ln(2));
    L := 0.1;
    for i := 0 to m-1 do begin
        p[i] := 0;
    end;   {for}
    p[m] := exp(-m*ln(2));
    n := m;
    repeat
        k1 := n mod (m+1);
        k2 := (n+1) mod (m+1);
        k3 := (n-m) mod (m+1);
        p[k2] := p[k1] + c * (1 - p[k3]);
        if (p[k2] > L) then begin
            writeln(lst, 'pn = ', p[k2]:1:8, ' n = ', n+1,
                    ' approx. n = ', round(-ln(1-L)*exp((m+1)*ln(2))),
                    ' bound = ', trunc(L*exp((m+1)*ln(2)) + m-2));
            L := L + 0.1;
        end;   {if}
        n := n+1;
    until (L > 0.5);
    writeln(lst, '');
end. {program}
```