Quantitative Plausibility of the Trojan Horse Defence against Possession of Child Pornography

R E Overill and J A M Silomon
Department of Informatics, King's College London, Strand, London,WC2R 2LS, UK
{richard.overill|jantje.a.silomon@kcl.ac.uk}

K-P Chow and Y W Law
Department of Computer Science, University of Hong Kong, Pokfulam Road, Hong Kong, PRC
{chow|ywlaw@cs.hku.hk}

Abstract: A new complexity-based metric has been developed to enable the relative plausibility of competing explanations for the existence of uncontested evidence to be determined quantitatively. This metric has been applied to the case of the Trojan horse defence against the possession of child pornography. Our results demonstrate that the Trojan horse defence in this case cannot be plausibly sustained, unless the defendant's computer was unprotected against malware.

Keywords: Trojan horse defence; child pornography; digital forensic evidence; complexity; quantitative plausibility metrics; posterior odds.

Introduction

The possession of digital images or movies containing scenes involving children that are classified as pornographic is a criminal offence in many jurisdictions. Seizure of such materials by law enforcement officers may lead to a criminal prosecution providing that the recovered digital forensic evidence is sufficiently convincing. Previous work has studied the types of digital forensic evidence that may be brought in support of such a prosecution [1, 2].
However, it is the experience of the law enforcement officers involved in such cases that the prosecution is often successfully countered by means of a Trojan horse defence (THD) [3-6]. That is, the defence claims that an automated digital process, of which the defendant was entirely oblivious, created the seized material. In effect, the defendant is claimed to have been the victim of a (targeted or random) electronic framing attack.

To be more specific, the argument from the defence lawyer typically rests on the following four assertions:

1. A Trojan installed itself on the defendant's computer;
2. The Trojan downloaded the recovered image files from a remote website (most probably not a public one);
3. The Trojan placed the downloaded image files in the location on the computer where the defendant usually works;
4. The Trojan then uninstalled itself, leaving no trace of itself on the defendant's computer.

The frequent success of this defence relies on the difficulty of proving a negative. That is, the defence's tactic is to challenge the prosecution to show that the alternative explanation for the existence of the seized material (the claimed operation of a piece of self-installing and self-uninstalling malicious software) did ***not*** occur, beyond a reasonable doubt.

Courts of law normally give the benefit of any perceived doubt to the defence side on the basis of the precept that the defendant is presumed innocent until proven guilty and as a result such prosecutions have frequently failed. The purpose of this paper is to demonstrate that the numerical plausibility of the THD in this type of case lies significantly below any notional threshold of a reasonable doubt. As a consequence the success rate for prosecutions of possession of child pornographic material should improve considerably.

Background

In the present study we have applied a complexity based model to the THD against the charge of possession of child pornography. In order to accomplish this it is first necessary to specify precisely what digital forensic evidence is typically recovered in such cases, and then to model both the human and Trojan pathways which lead to the creation of this uncontested evidence.

The typically recovered digital evidence (E) and the prosecution's associated sub-hypotheses (H) are set out as follows:

H1: Downloading of child pornography has been performed;
H2: Copying of child pornography has been performed;
H3: Viewing of child pornography has been performed;

Evidence for H1:
E1-1. Child pornography material (e.g. photos or a video) was recovered;
E1-2. Internet history or cached contents from downloading the material was found;
E1-3. Credit card payment record to child pornography website was discovered;
E1-4. Metadata of the child pornography material matched that of the items on child pornography website;
E1-5. A peer-to-peer (P2P) file-sharing program was found with traces showing that this tool was used to download child pornography;
E1-6. Emails containing child pornography attachments were found;

Evidence for H2:
E2-1. Registry entries showed that a USB device was plugged into the computer;
E2-2. Child pornography material (e.g. photos or a video) was recovered;
E2-3. Child pornography material found on the computer matched that found on the USB device;
E2-4. Child pornography material found on the computer matched that found on the CD/DVD;
E2-5. The modified timestamp predates the created timestamp of child pornography material;

Evidence for H3:
E3-1. Image or video viewing tools were found on the computer;
E3-2. Child pornography material (e.g. photos or a video) was recovered;
E3-3. Digital traces showed that the child pornographic materials were viewed by existing image or video viewing tools;
E3-4. In certain kinds of operating system (e.g. Windows XP), the access timestamp postdates the creation timestamp of the child pornographic material.

Note that it is quite common for several different download methods (e.g. browser, email, P2P) to be employed during the same session. Similarly, it is quite common for copies of the downloaded material to be made to several different media (e.g. USB, CD, DVD)

The Enhanced Complexity Model (ECM)

In our previous studies [7, 8] of the THD we developed an Operational Complexity Model (OCM) to evaluate the posterior odds of the THD *versus* the prosecution's hypothesis concerning the process by which the recovered digital evidence was created. The OCM was employed to study the five most common e-crimes in the Hong Kong region of China. One assumption made in the OCM is that all the software components necessary to construct the Trojan horse are freely available 'off-the-shelf' (OTS); any associated software integration issues are not considered.

In brief, the OCM [7, 8] employs computational complexity (CC) [9] and the GOMS Keystroke Level Model (KLM) [10] to evaluate the overall complexity of each of the *feasible routes*. These feasible routes are the alternative processes or mechanisms that are capable of creating the recovered digital forensic evidence.

Currently the OCM is configured to model a PC running Windows XP with MS Outlook and Internet Explorer. The OCM makes use of three distinct classes of numerical value, which we term *parameters*, *actual* or *exact* values, and *typical* or *average* values, respectively. Parameters are values, for example the size of a video, music or image file, which determine the overall magnitude of the process represented by a feasible route. Actual values refer to objects of known size, such as a Torrent file piece. Typical values are estimates of the average sizes of objects, for example the size of a web page.

A fundamental tenet of the OCM is that the more complex a process is, the less likely it is to occur user-obliviously (*i.e.* accidentally, unintentionally or spontaneously). The probability *p* of the explanation associated with feasible route *i* is modelled by the inverse relation:

$$p_i \propto (CC_i + KLM_i)^{-1}$$

Here, CC_i and KLM_i represent the computational complexity and the keystroke level complexity of feasible route *i*. For any number of mutually exclusive feasible routes $n>1$, the posterior odds *O* (sometimes also referred to as the odds ratio) are defined as the quotient of the posterior probabilities of feasible route *i* and the remaining n-1 feasible routes, given the recovered digital evidence *E*:

$$O(i) = \frac{\Pr(H_i|E)}{\sum_{j\neq i}\Pr(H_j|E)} = \frac{p_i}{\sum_{j\neq i} p_j}$$

Here H_i represents the hypothesis that the mechanism or process associated with feasible route *i* generated the recovered digital forensic evidence *E*. Furthermore, the posterior probability $\Pr(H_i|E)$ signifies the probability of H_i given the existence of *E*. In the present study only the simplest case of $n=2$ is required, representing the alternative feasible routes advanced by the prosecution and the defence.

$$O(i:j) = \frac{\Pr(H_i|E)}{\Pr(H_j|E)} = \frac{p_i}{p_j}$$

The Trojan horse hypothesis is modelled by the OCM as the simplest user-oblivious process that produces all of the requisite evidential traces and no others. The reason for this is not only to achieve clarity but, even more importantly, to produce a lower bound on the complexity of the Trojan horse process, which will be reflected in an upper bound on the plausibility of the Trojan horse hypothesis. Since a simpler Trojan horse model results in a higher plausibility for this alternative hypothesis, it enables the prosecution to assess the maximum plausibility of the defence's alternative explanation for the existence of the recovered evidence as a 'worst case scenario'.

In reality, however, the OTS nature of the OCM is an idealised assumption, and to reflect this fact an Enhanced Complexity Model (ECM) has recently been developed. The ECM builds upon the OCM by taking into account the additional effort needed to implement and integrate the required Trojan horse software components, by means of Halstead's effort (E) metric [11]. In the ECM, the CC and KLM metrics from the OCM are augmented by the addition of Halstead's E metric. It is appropriate to combine the Halstead E metric directly with the CC and KLM metrics from the OCM since all of them refer to fundamental operations at the token / byte level. This justifies the choice of Halstead's E-metric in the present context over McCabe's Cyclomatic Number, Boehm's COCOMO lines of code, or Albrecht's Function Points as measures of software complexity. Associated with Halstead's E metric is a further term, here denoted by KLM(E), which represents the actual typing of the source code. The new complexity metric may be represented symbolically by the relation:

$$\text{ECM} = \text{CC} + \text{KLM(CC)} + \text{E} + \text{KLM(E)}.$$

Halstead's E metric for a program [11] is defined in terms of the following software quantities:

> n_1 is the number of distinct operators (logical, relational, arithmetic, reserved words, type qualifiers and storage class specifiers);
>
> n_2 is the number of distinct operands (constants, identifiers and type specifiers);

N_1 is the total number of operators

N_2 is the total number of operands

Program vocabulary $n = n_1 + n_2$

Program length $N = N_1 + N_2$

Program volume $V = N \times log_2 n$

Programming difficulty $D = (n_1 \times N_2)/(2 \times n_2)$

Programming effort $E = D \times V$

Note that Halstead's volume metric V is an information theoretic measure of the information content of the program; it describes the size of an implementation based on the number of operations performed and the number of operands handled by the program. Halstead's difficulty measure *D* is related to the difficulty of implementing or understanding the program, based on the number of unique operators in the program and the ratio of the ratio of the number of unique operands to the total number of operands. Halstead's effort measure *E* is a software complexity measure directly proportional to the time required for implementing the software, and is based on the program volume and the program difficulty.

Although the present THD model is designed to be specific to the possession of child pornography, it should be emphasized that the model can be both adapted to other variants of the THD for this e-crime and also generalised to many other distinct e-crime scenarios (*e.g.* [7,8]). Since our THD model is both modular and programming language independent, the above adaptations and generalisations are relatively straightforward to accomplish. However, a more detailed verification of the model itself could only be obtained by performing a full implementation and evaluation.

Results and Discussion

The ECM is used to model the complexity of the alternative processes that lead to the creation of the evidence listed above. The numerical results are presented in Table 1, with those for the earlier OCM being given for comparison. Note that only the ECM results refer to the construction and operation of the Trojan horse since the in the OCM the Trojan horse code is assumed to be OTS.

For the purposes of this study we have adopted the following somewhat simplified scenario:

- A single image file of size 1MB was downloaded from the website (normally many such images would be downloaded during the same session);
- The material was downloaded directly from the website (often an email file-bot and/or P2P file-sharing might also be employed in the same session);
- The downloaded material was copied to a USB drive (but not to a CD or a DVD).

	OCM		**ECM**	
	Non-Trojan	**Trojan**	**Non-Trojan**	**Trojan**
CC	11,569,216	19,232,355	11,569,216	19,232,355
KLM(CC)	1,730	-	1,730	-
E	-	-	-	13,850,047
KLM(E)	-	-	-	1,381,959
Total	11,570,946	19,232,355	11,570,946	34,464,361

TABLE - OCM and ECM complexities for possession of child pornography material

Given that the probability of a process occurring unintentionally is inversely proportional to its complexity, the data in Table 1 can be used to determine the posterior odds against the THD for the possession of child pornography material. In the case of the fully-OTS OCM the posterior odds are 1.367 while for the non-OTS ECM they lengthen to 2.979. Note that both these odds ratios represent lower bounds since they do not take into account any intentionality (*mens rea*) on the part of the defendant in performing the image selection, payment and downloading activities in the non-Trojan scenarios. Furthermore, it will be noticed that so far no account has been taken of the degree of up-to-date malware protection in operation on the defendant's computer. Such systems typically have a 98% probability of detecting and neutralising Trojan horse malware [12]. Taking this statistic into account, the OCM odds ratio lengthens to 117.4 while the ECM odds ratio is extended to 197.9.

The posterior odds obtained in this study are somewhat shorter than those found in our previous work on the THD against five common e-crimes [7, 8]. The main reason for this is that in the present case the hypothecated Trojan horse does not need to carry a payload of data files in order to frame its victim. It simply downloads material from the same website that the defendant is alleged to have visited by the prosecution, in order to create the recovered evidence of downloading. The principal complication for the Trojan horse is that it first has to steal the victim's credit card data in order to pay for the downloaded material at the website. It achieves this by installing a key-logger that records the victim's keyboard and mouse activity over an extended period of time and transmits this data to the Trojan horse. The Trojan periodically scans the data for the necessary information arising from other incidental transactions and then presents the appropriately formatted credit card data to the website to accomplish the payment and download.

Thus our ECM model of the THD must take into account the self-installation and un-installation of the Trojan horse as well as its installation and un-installation of the key-logger. It also models the search of the key-logger output for credit card data making use of the Knuth-Morris-Pratt (KMP) string searching algorithm [13] (equivalent to the Unix *grep* command). In addition it models all the Trojan's necessary interactions with the website, generated using the AutoHotkey AutoScriptWriter [14]. Finally, it examines Windows Registry keys to ensure that a USB drive is available to receive a copy of the material. The executable code for all these operations comprises the Trojan horse payload in this case.

Note that the results presented here do not distinguish between a random or a targeted attack by the hypothecated Trojan horse. It is immaterial whether its initial installation was initiated as the result of a phishing attack, a drive-by download, a spear-phishing email, or a social network message. Thus the posterior odds derived here are applicable, whatever mode of THD the defence chooses to adopt.

In many jurisdictions the prosecution is required to show that the defendant possessed intentionality or *mens rea* of the illegal act. We observe here that evidence of online payment for downloads demonstrates a form of intentionality, especially if a number of images have been purchased. In the absence of any evidence of online payment or of access to the website in question, the defence would have to modify their THD to claim that the recovered images were included as data in the Trojan horse's payload. However, the resulting increased payload size would actually increase the overall complexity of the Trojan horse's task, thereby rendering this modified THD even less plausible.

We are also in a position to estimate the likely reduction in plausibility due to the non-recovery a particular evidential trace using the simple schemata developed previously [15]. In the case of a single missing evidential trace from sub-hypothesis H1, we find that the plausibility is reduced to between 80% and 94% of the value when all the evidential traces are present, depending on the schema used.

In some cases, forensic investigation of the defendant's computer may reveal the presence of spurious malware (unconnected with the hypothecated Trojan). Such a finding would be a cause for concern since it could indicate that the computer was not operating with effective anti-malware protection, which would make the assertion of the THD less implausible. However, it would also be necessary to ascertain forensically that the discovered malware had not been 'planted' in the defendant's computer in an attempt to render the THD more plausible.

Conclusions and Further Work

In this paper we have demonstrated how the THD against the charge of possession of child pornography can be modelled and analysed to determine quantitatively its plausibility *vis-à-vis* the prosecution's contention. Complexity based metrics have been employed to derive the odds ratio for the two competing explanations, given the uncontested recovered digital evidence. Although the ECM odds ratio against the THD in this case is relatively short (2.979 equates to a 75.0%**:**25.0% balance of probabilities in favour of the prosecution), the presence of an operational up-to-date anti-malware scanner on the defendant's computer lengthens the odds ratio to 197.9, equating to 99.5%**:**0.5% in favour of the prosecution. This latter figure is well above any notional threshold for satisfying the legal criterion of beyond reasonable doubt required in criminal prosecutions. We also take this opportunity to reiterate that these odds ratios represent lower bounds since they do not attempt to attribute any intentionality to the defendant's actions.

One aspect not explored in this work is the issue of bug prevalence in the software components that have to be implemented for the ECM of the Trojan horse. Halstead [11] offers an additional metric $B = V/3000$ for the number of bugs in the alpha version of the software implementation. This metric could be used to estimate the extra effort required to remove the salient bugs from the Trojan horse code which would add significantly to the overall complexity of the task, and hence reduce still further the plausibility of the THD.

Acknowledgement

The authors acknowledge Testwell for the grant of an evaluation licence for their *CMT++ - Complexity Measures Tool for C/C++* [16] which was used to calculate Halstead's E metric results.

References

[1] Chow, K P, Law, Y W F, Kwan, Y K M and Lai K Y, The Rules of Time on NTFS File System, Proceedings of the Second International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE '07) IEEE Computer Society Washington, DC (2007) pp. 71–85.

[2] Law Y W F, Chow, K P, Lai, K Y P, Tse K S H and Tse, W H K, Digital Child Pornography: Offender or not Offender, in Technology for Facilitating Humanity and Combating Social Deviations: Interdisciplinary Perspectives (Eds. Martin, V M, Garcia-Ruiz, M A & Edwards, A), Information Science Reference, IGI Global (2011) ch. 1.

[3] Haagman, D and Ghavalas, B, Trojan Defence: A Forensic View, Digital Investigation, 2 (1) (2005) 23–30.

[4] Ghavalas, B and Philips, A, Trojan Defence: A Forensic View, part II, Digital Investigation, 2 (2) (2005) 133–136.

[5] Mason, S, Trusted Computing and Forensic Investigations, Digital Investigation, 2 (3) (2005) 189–192.

[6] Brenner, S W, Carrier, B and Henninger J, The Trojan Horse Defence in Cybercrime Cases, Santa Clara Computer & High Tech. Law J., 21 (1) (2004) 9–61.

[7] Overill, R E, Silomon, J A M and Chow, K-P, A Complexity Based Model for Quantifying Forensic Evidential Probabilities, in Proc. 3rd International Workshop on Digital Forensics (WSDF 2010), Krakow, Poland, 15–18 February 2010, pp.671–676.

[8] Overill, R E and Silomon, J A M, A Complexity Based Forensic Analysis of the Trojan Horse Defence, in Proc. 4th International Workshop on Digital Forensics (WSDF 2011), Vienna, Austria, 22–26 August 2011, pp.764–768.

[9] Papadimitriou, C H, *Computational Complexity*, Addison-Wesley, Reading, MA (1994).

[10] Kieras, D, Using the Keystroke Level Model to Estimate Execution Times, University of Michigan (2001), available online at: http://www.cs.loyola.edu/~lawrie/CS774/S06/homework/klm.pdf

[11] Halstead, M H. *Elements of Software Science*. Amsterdam: Elsevier North-Holland (1977).

[12] AV-Comparatives, Anti-Virus Comparative On-demand Detection of Malicious Software (August 2011, last revision: 27 September 2011), pp.5–6, available online at: http://www.av-comparatives.org/images/stories/test/ondret/avc_od_aug2011.pdf

[13] Knuth, D E, Morris, J H and Pratt, V, Fast pattern matching in strings, *SIAM Journal on Computing* **6** (2) (1977) 323–350.

[14] AutoHotkey AutoScriptWriter, available at http://www.autohotkey.com/

[15] Overill, R E and Silomon, J A M, Six Simple Schemata for Approximating Bayesian Belief Networks, in Cyberforensics: Issues and Perspectives (Ed. G R S Weir), University of Strathclyde Publishing (2011), pp.65 – 72.

[16] Testwell, CMT++ - Complexity Measures Tool for C/C++, available at http://www.testwell.fi/