Software Watermarking:
Models and Dynamic Embeddings

Christian Collberg    Clark Thomborson

Dept. of Computer Science
The University of Auckland

Outline

• Watermarking definition
• How it works (or does not work)
• Design issues
• Watermark types and examples
• Dynamic Graph Watermarking
• Advantages and disadvantages
What is Watermarking?

- Watermarking embeds a secret message into a cover message.
- Example: Media watermarking
  - secret: copyright notice
  - cover: digital image

Problem Definition

- Embed a structure $W$ into a program $P$ such that:
  - $W$ can be reliably located and extracted from $P$ even after $P$ has been transformed.
  - $W$ is stealthy.
  - $W$ has a high data rate
  - embedding $W$ into $P$ does not affect the performance of $P$.
  - $W$ has a mathematical property that proves that its presence in $P$ is a result of deliberate actions.
How watermarking works

Kinds of attacks
Tamperproofing

Alice

Bob

Fingerprinting

Alice

Bob
Design issues

- required data rate
  - how large can the watermark be compared to size of program.
- form of cover program
  - will distribution be virtual machine code or native binary code.
- expected threat model
  - what kinds of de-watermarking attacks can we expect from Bob.

Static watermarking

Stored in application executable.

- data watermarks
  - embedded in string sections, debugging information sections, etc.
- code watermarks
  - embedded in the code section.
Dynamic watermarking

Stored in program’s execution state.

- easter egg watermark
  - given a certain input, display a gif file.
- dynamic data structure watermark
  - store watermark in a data structure. Use a watermark extraction routine to read contents of data structure.
- dynamic execution trace watermark
  - use addresses or instruction sequences to store watermark.

Watermarking examples

```c
const \nC="Copyright..."

char V;
switch e {
    case 1:V='C'
    case 5:V='O'
        ....
}

if input == "I"
    display(Teampic)

string V
if input == "I"
    V[1]='C';
    V[2]='O';
    ....

push 'C'
push 'O'
    ....
```
Dynamic Graph Watermarking

- embed a watermark in the topology of a dynamically built graph structure.
- Advantages
  - due to pointer aliasing, code manipulating dynamic graph structures is hard to analyze.
  - semantics-preserving transformations are hard to construct.
  - Easy to tamperproof.

Scheme overview

Alice

\[
P = \text{prime}();
Q = \text{prime}();
n = P \times Q;
\]

\[
p = \text{new node}();
q = \text{new node}();
\text{addedge}(p, q)
\]

\[
p = \text{new node}();
q = \text{new node}();
\text{addedge}(p, q)
\]

\[
\text{if tampered}(W) \text{ die};
\]
Scheme overview

Embedding the Watermark

Radix-\( k \) encoding  
\( (k = 6) \)

\[
61 \times 73 = 3.6^4 + 2.6^3 + 3.6^2 + 4.6^1 + 1.6^0
\]
Attacks against the watermark

- add extra pointers to the nodes.
- rename and reorder the fields in the node.
- add levels of indirection by splitting nodes into several linked parts.
- add extra bogus nodes preventing us from finding the root.

Disadvantages

- Watermarks the entire program, not individual modules.
- No immunity from additive attacks.
- Statistical attack may be successful – it depends on the nature of the watermark.
Conclusion

- Software watermarking is a process of embedding a secret message into code.
- It's a relatively new area of academic research.
- Several kinds of attacks still not addressed well enough.