Intellectual Property Protection -- Good Watermarks

Cybersecurity Specialization
-- Hardware Security

What Makes a Good Watermark?

- Correct functionality
- Low overhead
- High credibility
- Easy detectability
- Resilience
- Transparency
- Part protection
- Fairness

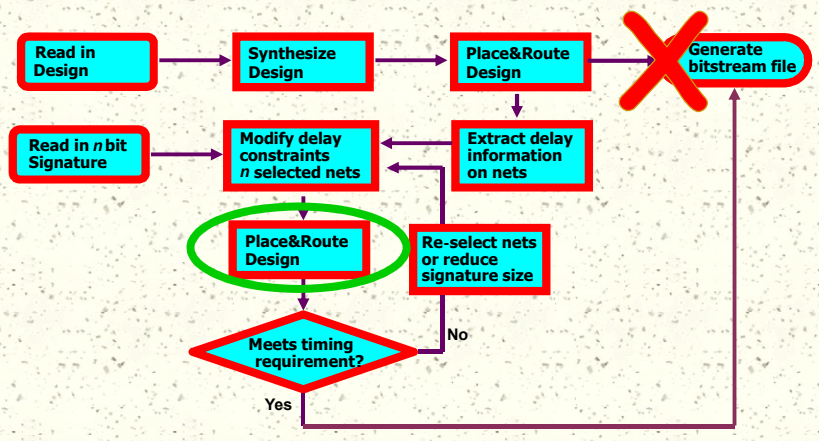
What Makes a Good Watermark?

- Correct functionality
- Low overhead
- High credibility
- Easy detectability
- Resilience
- Transparency
- Part protection
- Fairness

Zero Overhead Watermarking

- Unpredictable design overhead
  - Random constraints based on the watermark
  - Non-deterministic design tools and software
  - Controllability vs. security
- 2-phase zero-overhead watermarking
  - Design as normal for optimal performance
  - Identify places for watermark embedding without causing performance degradation
  - Embed the watermark
  - Re-design (whole or partially)

FPGA Design Watermarking



Example Watermarking Scheme

- Pick non-critical paths
- Modify the last digit of delay constraints

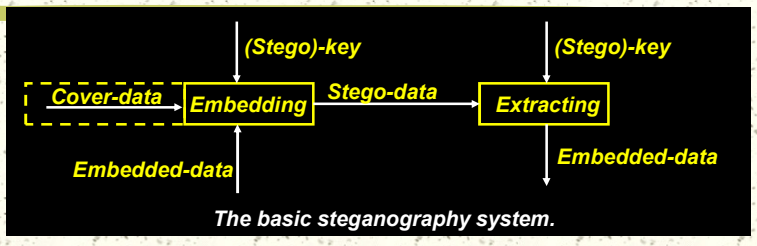
Source FF	Destination FF	Original Delay	Watermarking bit	Constrained Delay
inst_reg_0	aluinp1_reg_3	17.758	0	17.750
inst_reg_6	aluinp1_reg_3	17.755	1	17.751
inst_reg_1	aluinp1_reg_3	17.733	0	17.730
inst_reg_3	aluinp1_reg_3	17.651	0	17.650
inst_reg_6	aluinp1_reg_1	17.374	1	17.371
inst_reg_1	aluinp1_reg_1	17.352	1	17.351
inst_reg_5	aluinp1_reg_3	17.312	0	17.310
inst_reg_0	aluinp1_reg_5	17.066	1	17.061

Design Validation Results

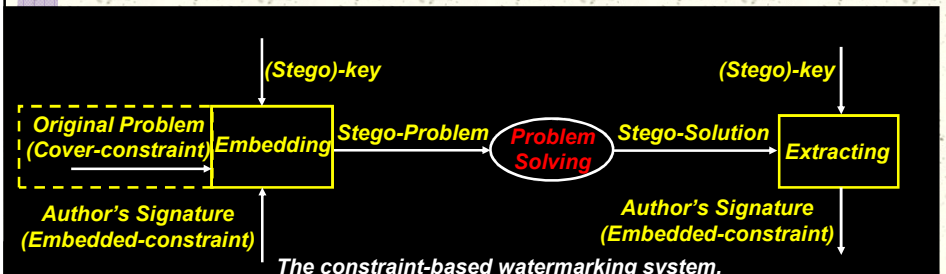
- No overhead in: performance, resource
- Design time (re-P&R): seconds
- Strong proof of authorship

FPGA Designs		Original	Watermarked	H-distance
DAP	Resources	1011	1011	1.09%
	$f_{required}$: 40 MHz	√	√	
Video	Resources	961	961	4.39%
	$f_{required}$: 40 MHz	√	√	
RiSC	Resources	410	410	2.52%
	$f_{required}$: 50 MHz	√	√	
Addrgen	Resources	174	174	2.41%
	$f_{required}$: 50 MHz	√	√	

Constraint-Based Watermarking



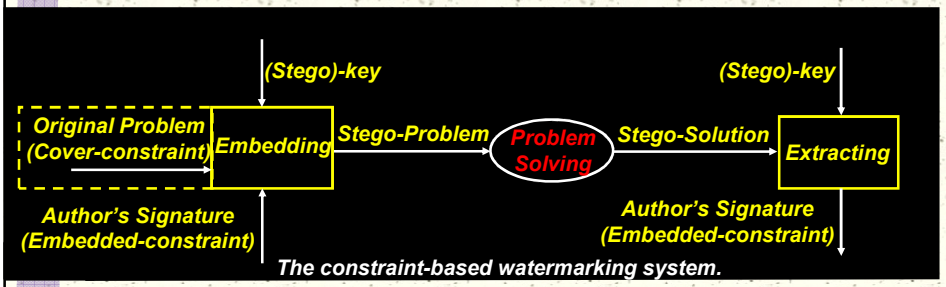
The basic steganography system.



The constraint-based watermarking system.

Constraint-Based Watermarking

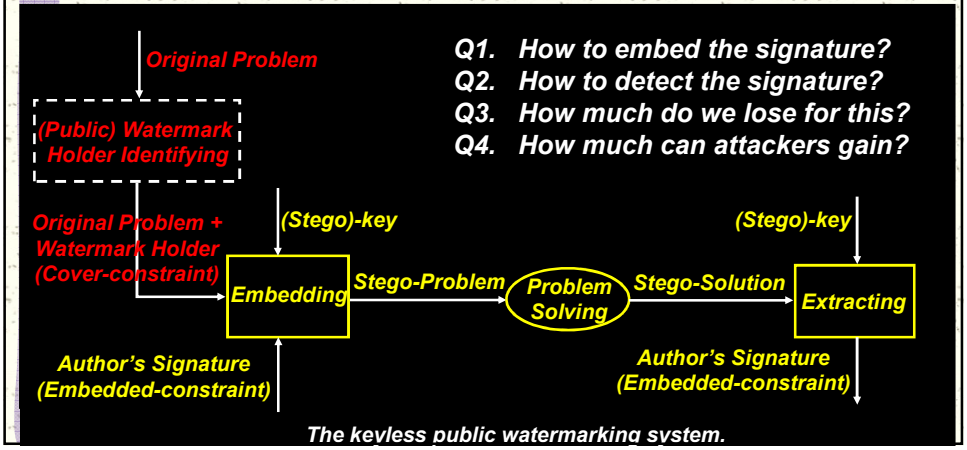
- Problem: how to discover the embedded watermark and prove authorship?



The constraint-based watermarking system.

Public Watermarking

- Problem: how to discover the embedded watermark and prove authorship?

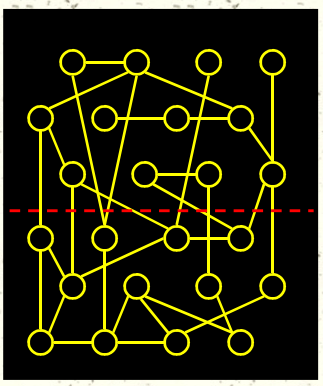


Q1. How to embed the signature?
Q2. How to detect the signature?
Q3. How much do we lose for this?
Q4. How much can attackers gain?

The keyless public watermarking system.

Graph Partitioning (GP) Problem

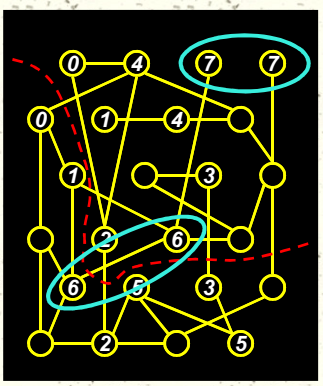
- Partition the vertices into two disjoint subsets such that
  - the subsets are balanced
  - connection is minimized
- Example:
  - Perfectly balanced, 12 vertices in each subset
  - 10 edges being cut
- GP problem is NP-complete



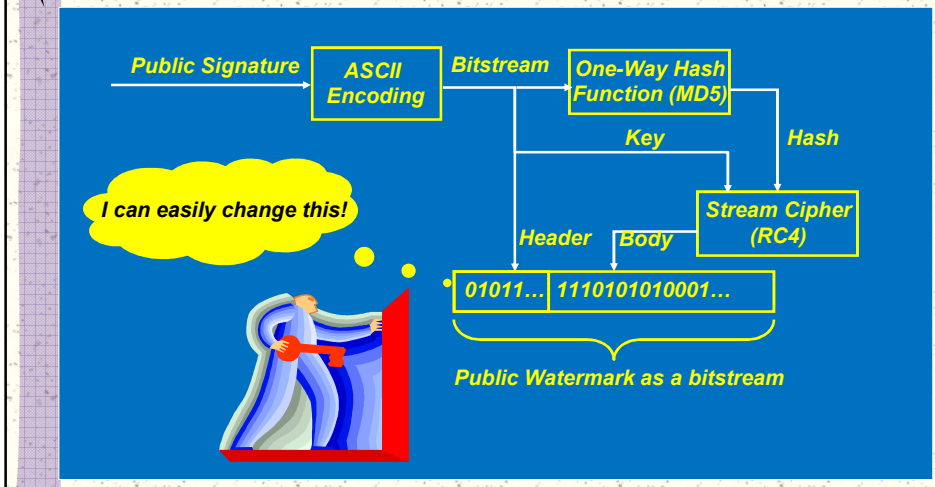
Public Watermarking GP Problem

- Make the followings public:
  - 8 pairs of nodes
  - Public watermarking rule:
    - 0: pair in the same subset
    - 1: pair in different subsets
- Example: embedding 'O'
  - 'O' in ASCII: 01001111
  - Partition the 8 pairs first
  - Partition the rest nodes
  - Everyone can detect and verify this

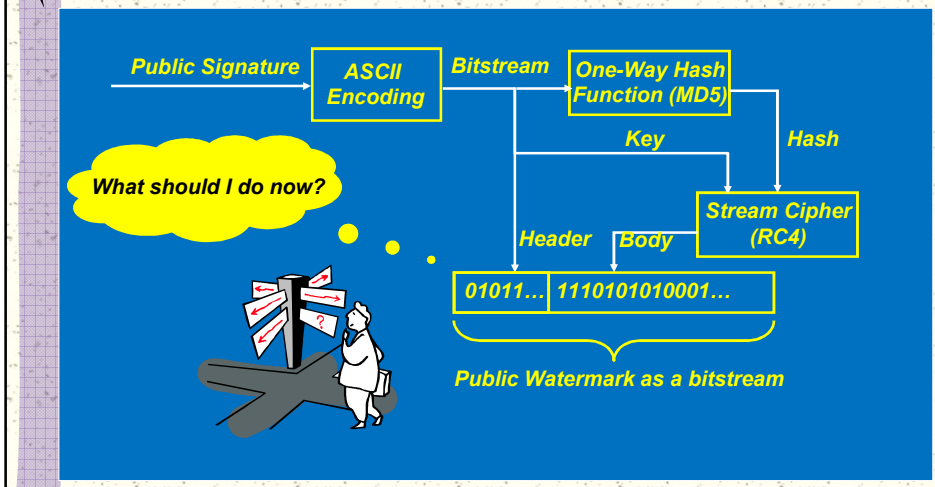
→ public watermark



Secure the Public Watermark



Secure the Public Watermark



Detection/Authentication Is Easy