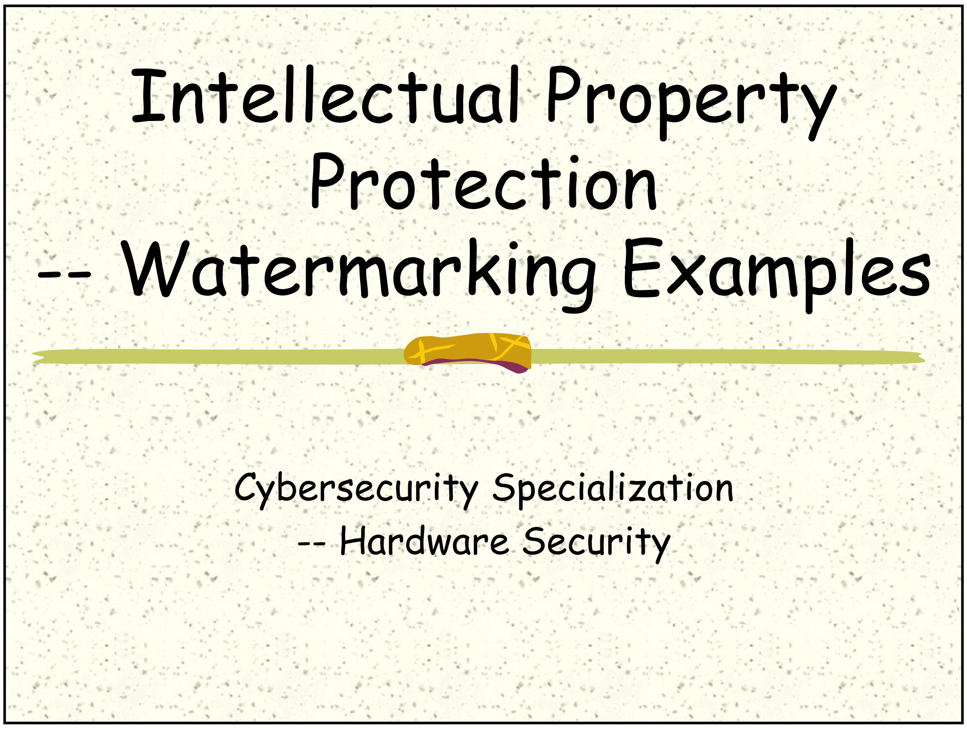
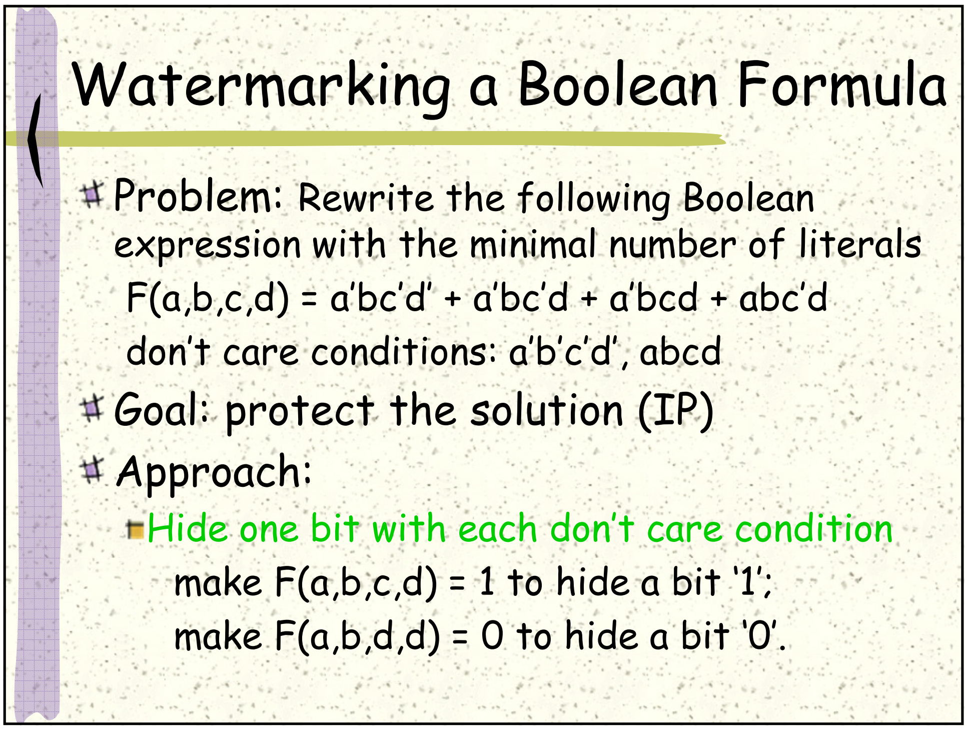


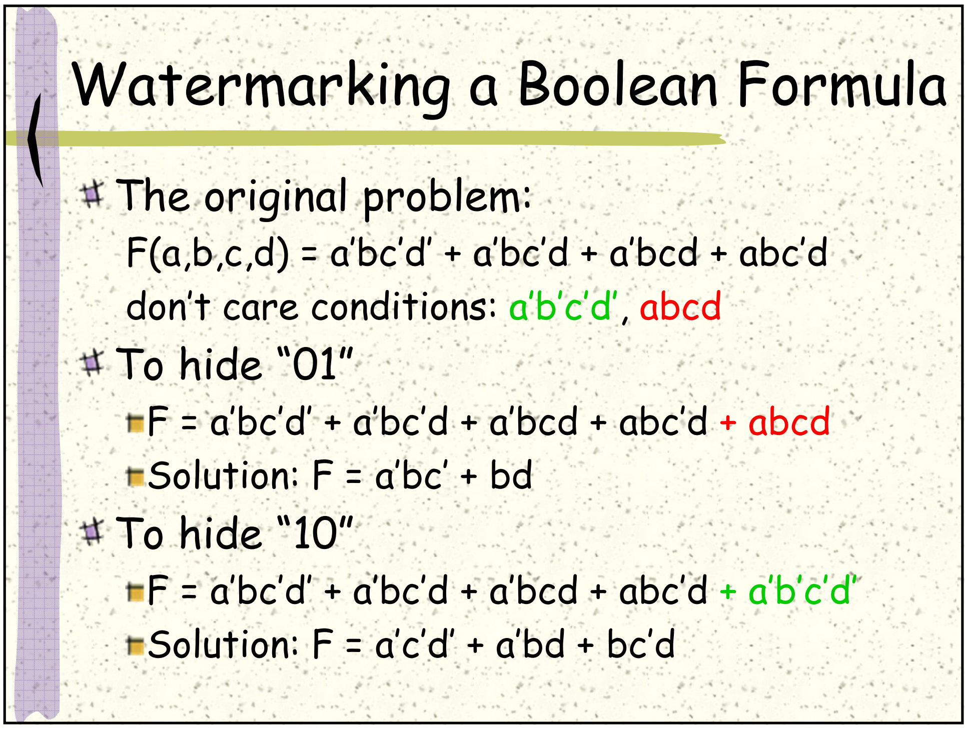
Intellectual Property Protection -- Watermarking Examples

Cybersecurity Specialization
-- Hardware Security



Watermarking a Boolean Formula

- Problem: Rewrite the following Boolean expression with the minimal number of literals
  F(a,b,c,d) = a'bc'd' + a'bc'd + a'bcd + abc'd
  don't care conditions: a'b'c'd', abcd
- Goal: protect the solution (IP)
- Approach:
  - Hide one bit with each don't care condition
    make F(a,b,c,d) = 1 to hide a bit '1';
    make F(a,b,d,d) = 0 to hide a bit '0'.

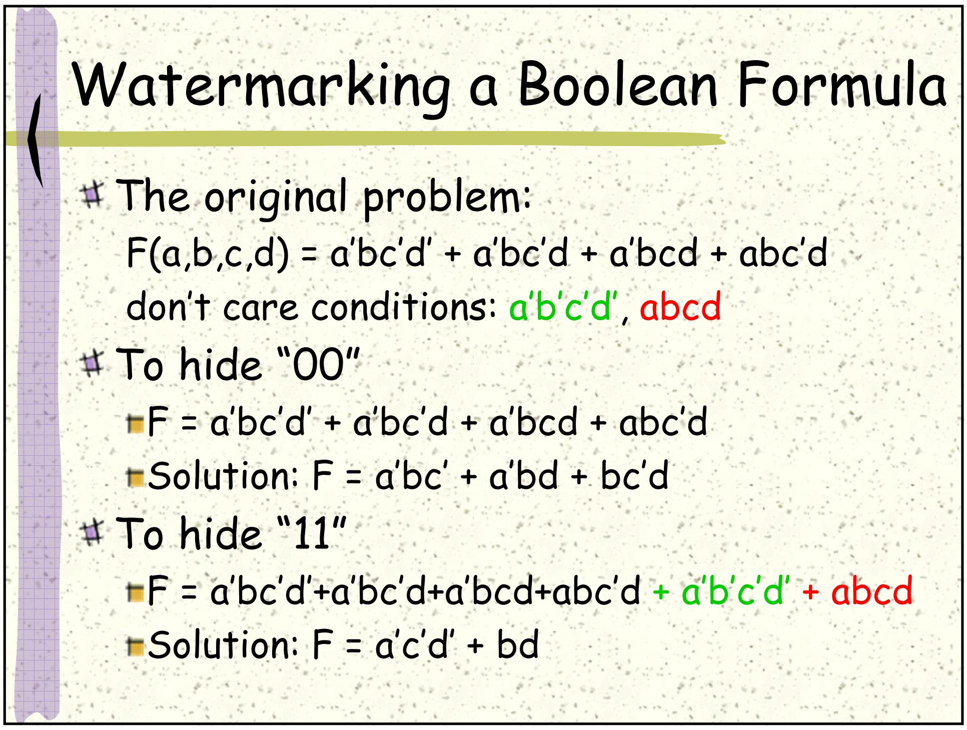


Watermarking a Boolean Formula

- The original problem:

  F(a,b,c,d) = a’bc’d’ + a’bc’d + a’bcd + abc’d

  don’t care conditions: a’b’c’d’, abcd
- To hide “01”
  - F = a’bc’d’ + a’bc’d + a’bcd + abc’d + abcd
  - Solution: F = a’bc’ + bd
- To hide “10”
  - F = a’bc’d’ + a’bc’d + a’bcd + abc’d + a’b’c’d’
  - Solution: F = a’c’d’ + a’bd + bc’d

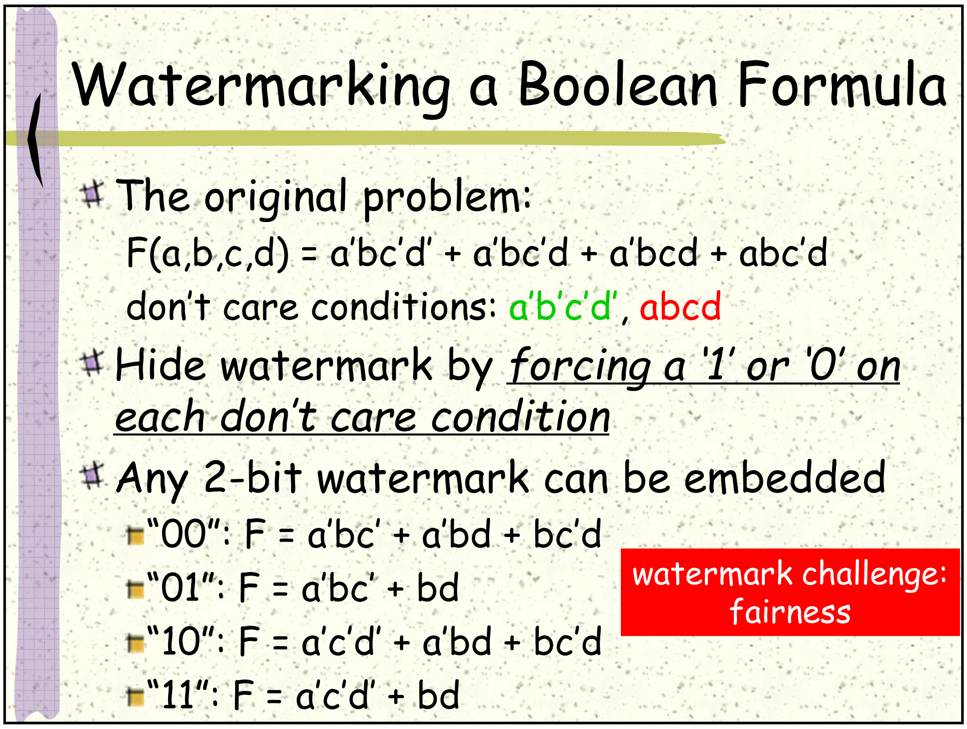


Watermarking a Boolean Formula

- The original problem:

  F(a,b,c,d) = a’bc’d’ + a’bc’d + a’bcd + abc’d

  don’t care conditions: a’b’c’d’, abcd
- To hide “00”
  - F = a’bc’d’ + a’bc’d + a’bcd + abc’d
  - Solution: F = a’bc’ + a’bd + bc’d
- To hide “11”
  - F = a’bc’d’+a’bc’d+a’bcd+abc’d + a’b’c’d’ + abcd
  - Solution: F = a’c’d’ + bd



Watermarking a Boolean Formula

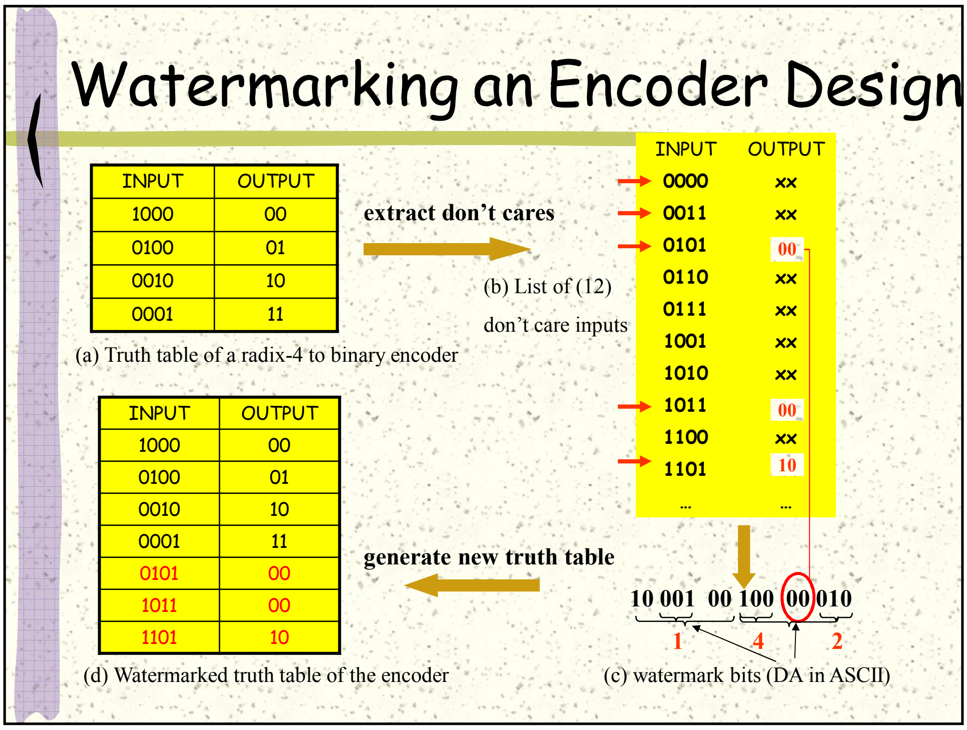
- The original problem:

  F(a,b,c,d) = a'bc'd' + a'bc'd + a'bcd + abc'd

  don't care conditions: a'b'c'd', abcd

- Hide watermark by *forcing a '1' or '0' on each don't care condition*
- Any 2-bit watermark can be embedded
  - "00": F = a'bc' + a'bd + bc'd
  - "01": F = a'bc' + bd
  - "10": F = a'c'd' + a'bd + bc'd
  - "11": F = a'c'd' + bd

watermark challenge: fairness



Watermarking an Encoder Design

INPUT	OUTPUT
1000	00
0100	01
0010	10
0001	11

(a) Truth table of a radix-4 to binary encoder

extract don't cares

(b) List of (12) don't care inputs

INPUT	OUTPUT
0000	xx
0011	xx
0101	00
0110	xx
0111	xx
1001	xx
1010	xx
1011	00
1100	xx
1101	10
...	...

$\underbrace{10\ \underbrace{001}\ 00}_{1}\ \underbrace{\underbrace{100}_{4}\ 00\ \underbrace{010}_{2}}$

(c) watermark bits (DA in ASCII)

generate new truth table

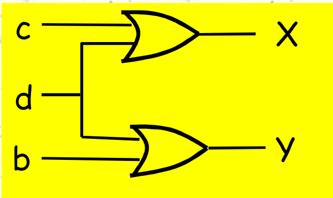
INPUT	OUTPUT
1000	00
0100	01
0010	10
0001	11
0101	00
1011	00
1101	10

(d) Watermarked truth table of the encoder

Watermarking an Encoder Design

INPUT	OUTPUT
1000	00
0100	01
0010	10
0001	11

$X = c + d$
$Y = b + d$



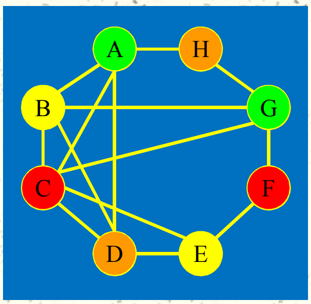
INPUT	OUTPUT
1000	00
0100	01
0010	10
0001	11
0101	00
1011	00
1101	10

$X = c + a'b'$
$Y = bc + b'c'd$

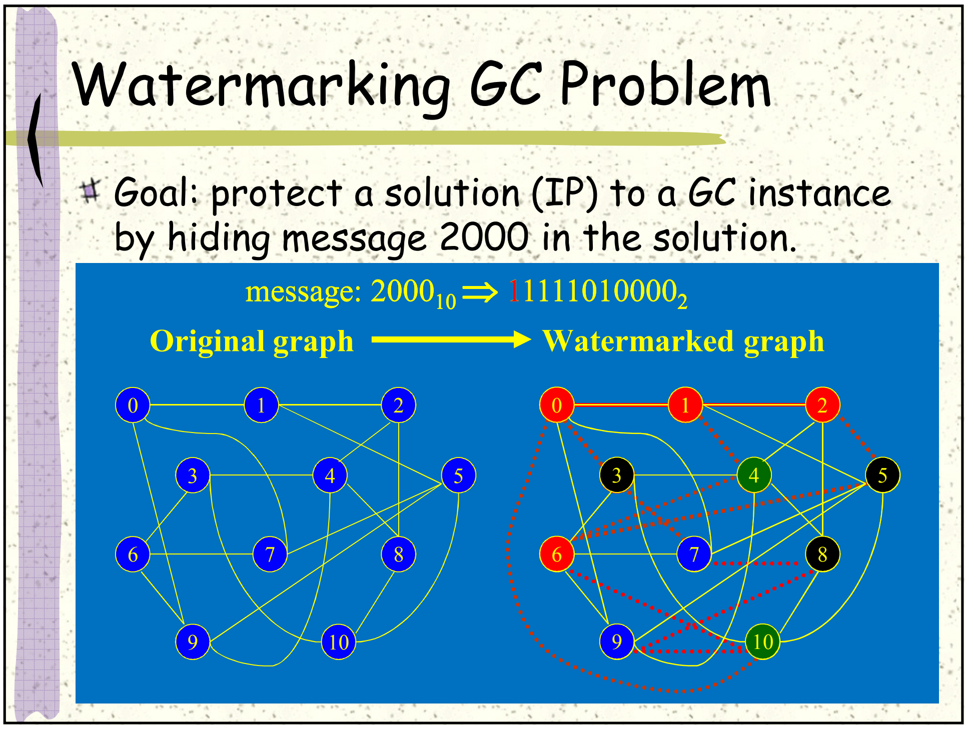
watermark challenge:
design overhead

Graph Coloring (GC) Problem

- Given: an undirected graph, color the graph by assigning each vertex a color
- Subject to: two vertices that are connected by an edge cannot receive the same color
- Minimize: the number of colors required



- 2-colorable is trivial
- 3-colorable is NP-complete
- One of the most important NP-complete problems
- Many applications in VLSI



Watermarking GC Problem

- Goal: protect a solution (IP) to a GC instance by hiding message 2000 in the solution.

message: $2000_{10} \Rightarrow 11111010000_2$

Original graph → Watermarked graph