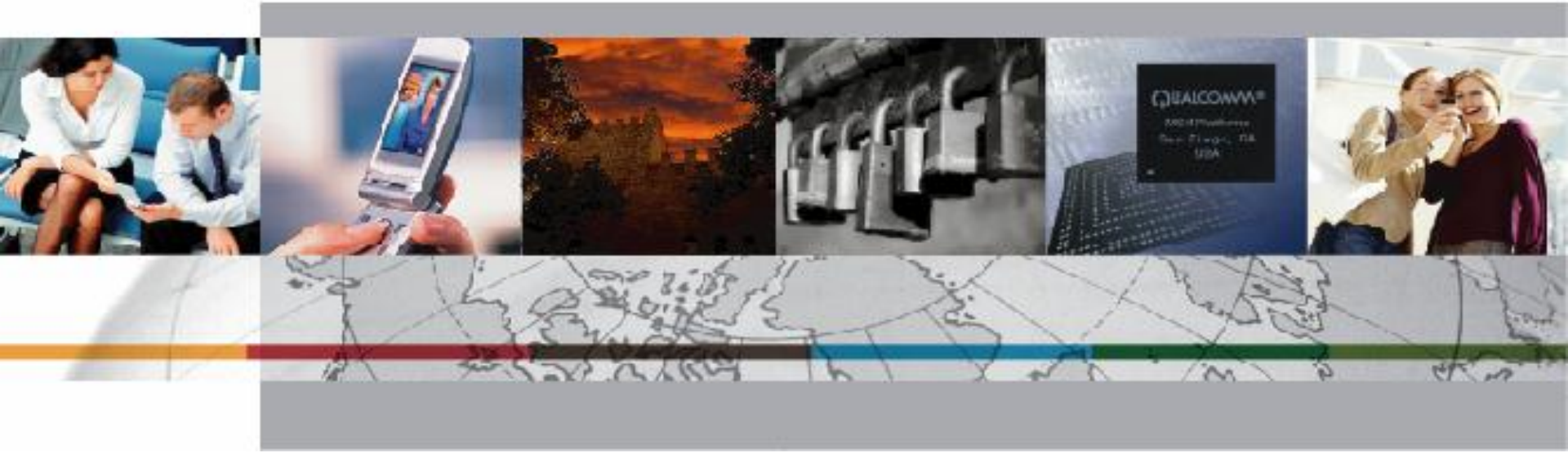
PRODUCT SECURITY INITIATIVE

QUALCOMM



A new efficient construction for wide S-boxes
2013 ESC Workshop

Greg Rose

Outline of talk

- Introduction
- My goal in life
- Named after …
- The Hidden Weighted Bit Function
- Wide S-box based on HWBF
- Performance
- Other stuff

My goal in life

- To work "close to the edge"
  - very efficient constructions
  - that are either barely secure, or a little bit insecure
- To provide plenty of opportunities for grad students and postdocs

More recently:

- To find suitable wide, computable S-box

What happened to Boole?

- (The hash function, that is)
- Pre-image attack (Dimitry Khovratovich, Ivica Nikolic, Ralf-Philipp Weinmann)
  - Just stupidity on my part
- Collisions (Florian Mendel, Tomislav Nad and Martin Schläffer)
  - S-box not 1-1
- Crossword puzzle attack
  - Bad S-box construction
  - biased
  - not bijective

O'Toole

- Named after Peter O'Toole, Born 1932
- Not dead yet
- Known to be rarely sober.
- Nominated for more Academy Awards, without winning, than any other actor

The Hidden Weighted Bit Function

- Bryant '81, Knuth Vol 4.
- paper to appear, proves nice crypto properties
  - high nonlinearity, balance, high algebraic degree, strict avalanche, exponential BDD

$W(x)$ = (

0 if $x = 0$

x_i otherwise, where $i = \mathrm{HWT}(x)$, $i \in \{1..n\}$

)

- Only n-bit to 1-bit

Extend to full word

- Maslov '07
- (I thought of it independently, but too late)
- Simply rotate the word

$W(x) = x >>> (HWT(x) \bmod n)$

- Each bit of the output is HWBF

Still some undesirable properties:

- 0, 0xFFF..F are fixed points, words of very low (resp high) Hamming Weight are short cycles
- Hamming Weight is preserved

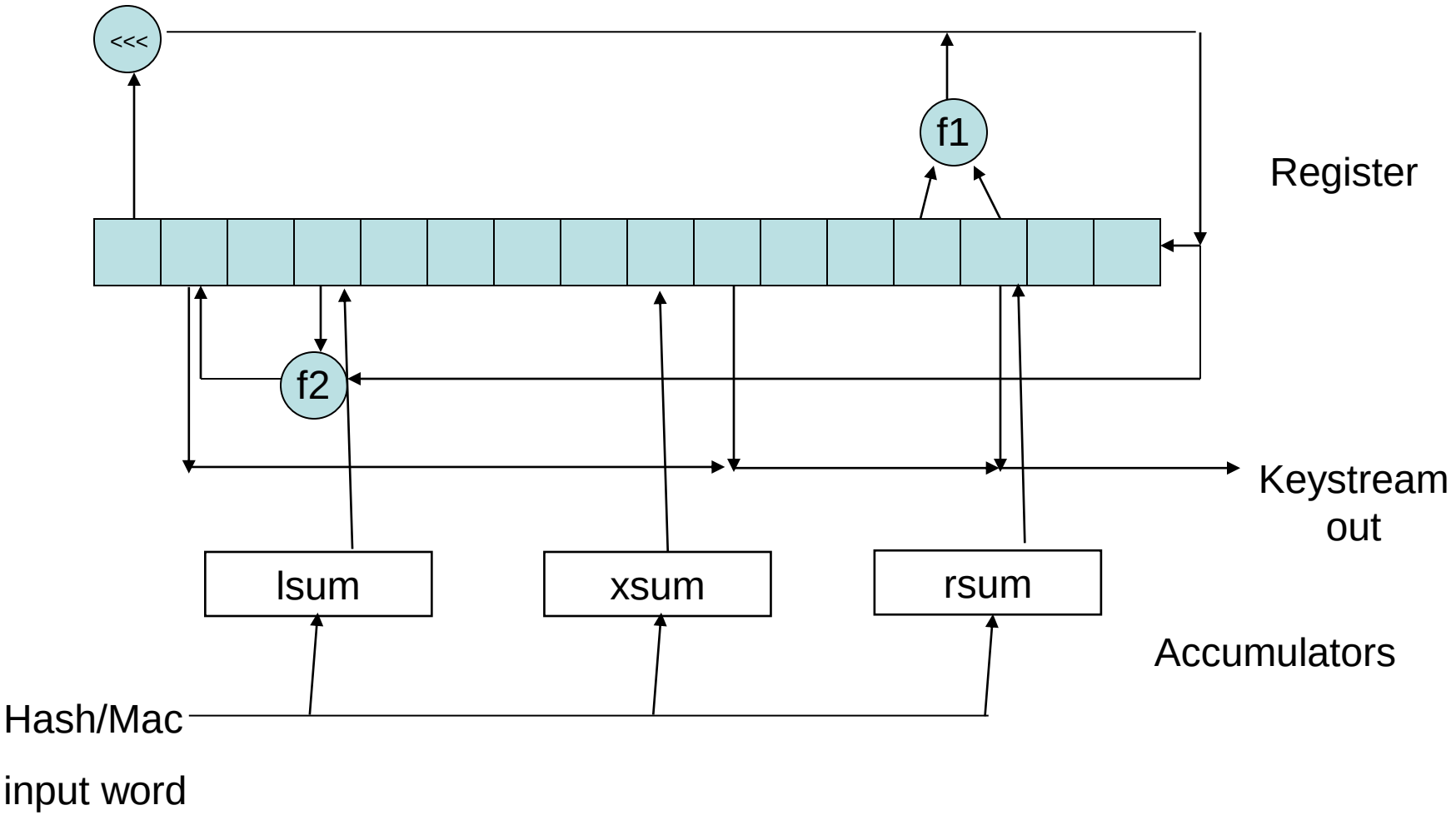
Wide S-box construction

```
W(x) {
        x = x ⊕ C1;
        return ((x >>> HWT(x)) ⊕  C2);
}
```

- C1 and C2 can be constants of "normal" Hamming Weight
- … or can be derived from some keying process
- (completely dynamic invalidates proofs)

- Defines a family of S-boxes with many desirable properties

Diagram


<<<
f1
Register
f2
Keystream
out
lsum
xsum
rsum
Accumulators
Hash/Mac
input word

Performance

- Hamming Weight is easy
  - Many CPUs have “popcount” instructions now
  - $\log_2(n)$ masks, shifts, adds otherwise
  - a few layers of gates
- XOR is easy
- Rotation is easy
  - Constant time in most implementations
  - a few layers of MUXes in hardware
- Benchmarks of “O’Toole” maybe later in week

Tesla

Antarctica