Drew Dean
Program Manager, Information Innovation Office

PROCEED and Crowd-sourced Formal Verification

DARPA Cyber Colloquium
Arlington, VA

November 7, 2011

Do you trust the cloud?

Source: Library of Congress/Flickr

Secure communications...

Source: General Services Administration

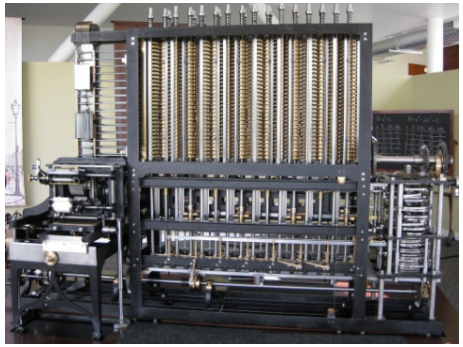
Secure storage...

Source: Christopher Bowns/Flickr

Secure computation?

PROgramming Computation on EncyrptEd Data (PROCEED)

Goal: practical computation on encrypted data without decrypting



Source: Catherine Helzerman /Fickr

Babbage Difference Engine

150 years

1832 - 1982

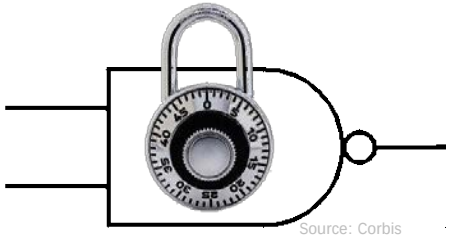
7 Orders of Magnitude

Source: Flylogic Engineering LLC; Corbis

Intel 80286

2010 - 2015

5 years



Source: Corbis

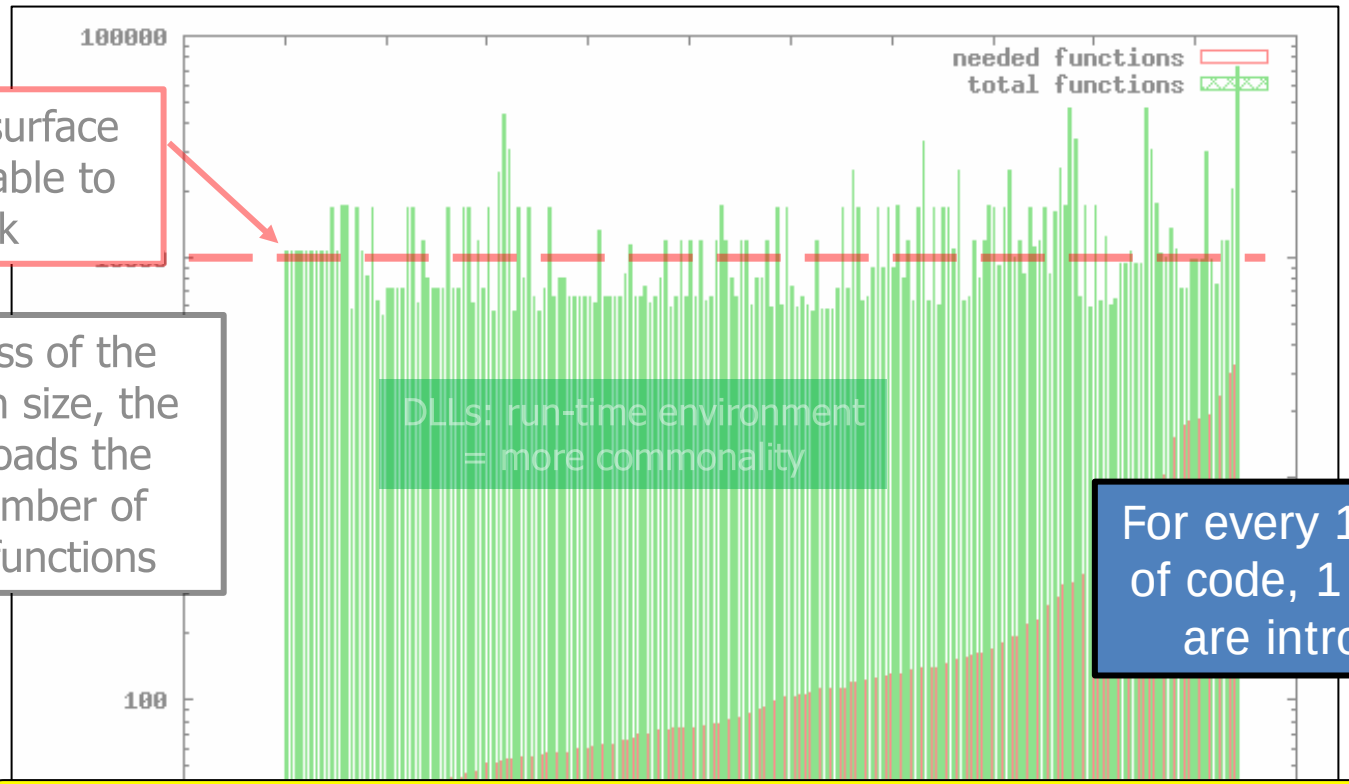
Encrypted NAND Gate

Potential Applications

- Email content-filtering guard between networks with different classification levels
- Privacy-preserving cloud-based voice over IP service
- Secure cloud-based mapping service that cannot determine your location, route, or destination

Crowd Sourced Formal Verification (CSFV)

The Problem



Constant surface area available to attack

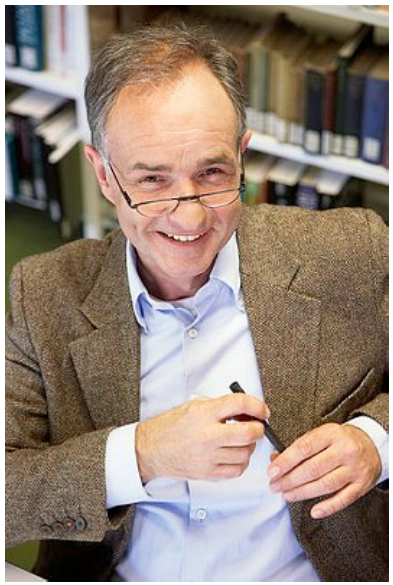
Regardless of the application size, the system loads the same number of support functions

For every 1,000 lines of code, 1 to 5 bugs are introduced.

Are there fundamental scientific reasons that prevent us from doing better?
No: *"There are no intrinsic laws of nature in cyber-security as there are in…physics, chemistry, or biology."*
[JASON Report on Science of Cyber-Security, 2010]

Formal Verification

- Formal verification can obtain 0.1 - 0.5 bugs per KLOC, however:
  - Extremely expensive: software development costs increase by 2x to 100x
    - seL4 microkernel formal verification took 11 person-years
  - Fundamental formal verification problems resist automation
    - Computationally undecidable: Heuristics have improved, but remain incomplete



Source: Corbis

CSFV

Source: morgueFile

The Concept: Crowd Sourced Formal Verification

"Game-ify" Geeky Formal Verification

Applies game solutions to the original formal verification problem

Exploits a large user base requiring no formal verification expertise

Code

Model

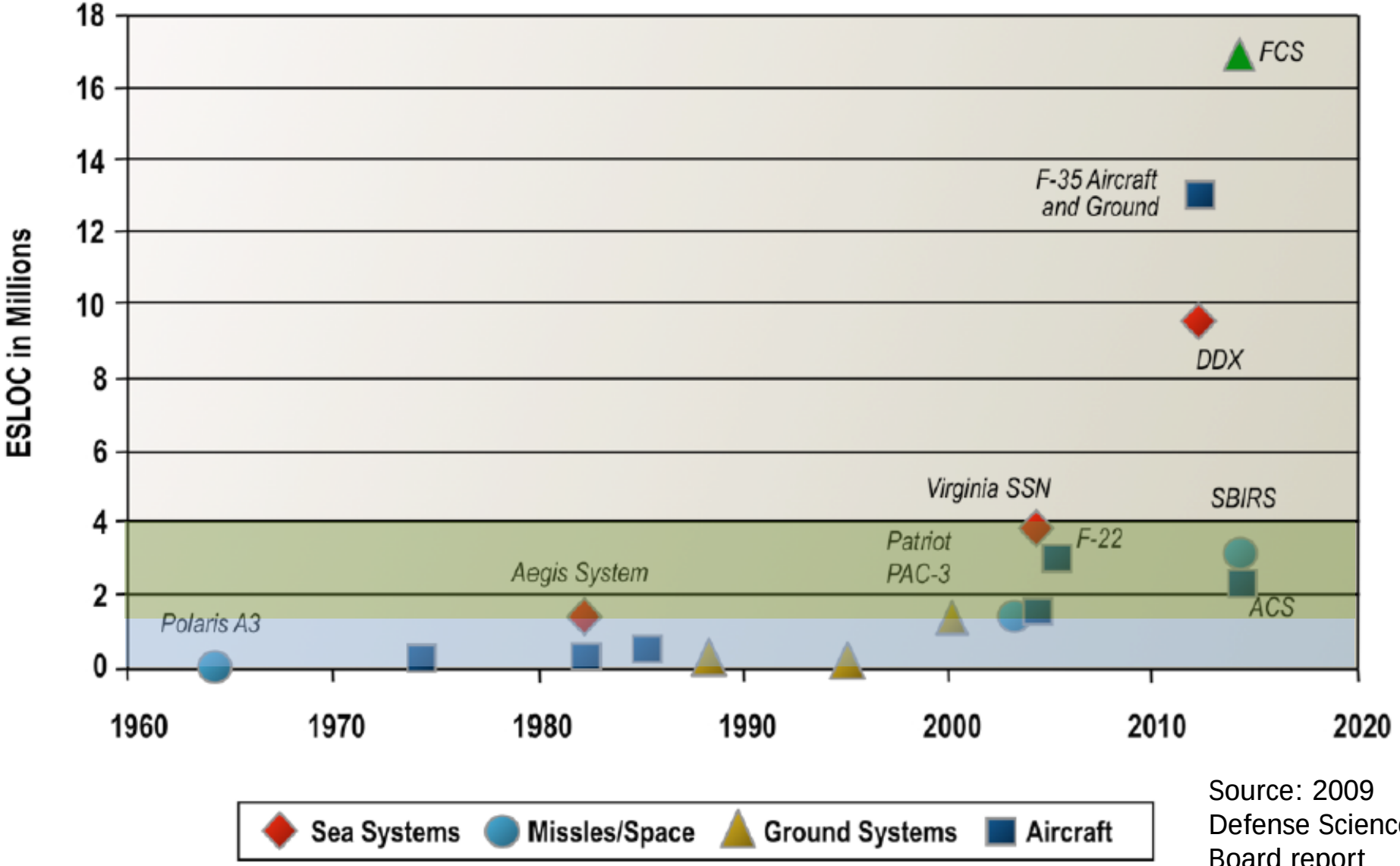
Verified Code

Verified Model

Source: University of Washington

CSFV New Capabilities

Scalability to DoD Software Systems



ESLOC = Executable Source Lines Of Code

Source: 2009 Defense Science Board report

Contact Information

Watch for Special Notice SN 12-17 to be released on FedBizOpps (fbo.gov)

Drew Dean

Drew.Dean@darpa.mil