CAC :

- Design Level
- Implementation Level.
- Deployment Level.

Main challenges:

- Difficult to understand guarantees and fine-print caveats.
- Broad field, complex and rapidly evolving.

Design Level.

Symbolic
[Formal methods community]

Computational.
[Cryptography community]

Why? Because math is the only proof of security.

Current: Pen and Paper math [Error prone]

↳ Methods to minimize errors.

- → Code based game playing ↳ Add realism.
- → Universal composability

Still error prone.

Math proofs are hard, hence we need machines that can help. [Halevi 05]

Symbolic Model.

→ Primitives are black boxes.

→ Terms are atomic

⇒ Adv needs to know full k to decrypt.

Symbolic security
- Trace properties.
- Equivalence properties

What is the difference?

Computational Model.

Keys and messages are bitstrings as opposed to blackboxes in symbolic model.

Computational security →
- Game based.
- Simulation based.

Computational security →
- Concrete
- Asymptotic.

Game based. → Games between challengers and adversaries.

Methodology → Game hopping [Shoup / BR]

Simulation based → "Real" and "Ideal"

More complicated but support composition theorems.

Concrete security → Involves quantifying the security by bounding the success prob.

(t, ϵ) - secure scheme.

Asymptotic security →

Views run-time of adversary and success prob as fuctions of some parameter

Secure scheme is one which is broken by polynomial time adv with negligible prob.

CAC has focused on game-based, concrete security notions.