Programming Languages and Cryptography

Formal verification of cryptography

(of proofs and impl)

Help non-experts program crypto

(prog lang. design, protocol synthesis)

Application areas

(cryptocurrencies, consensus, trusted hardware)

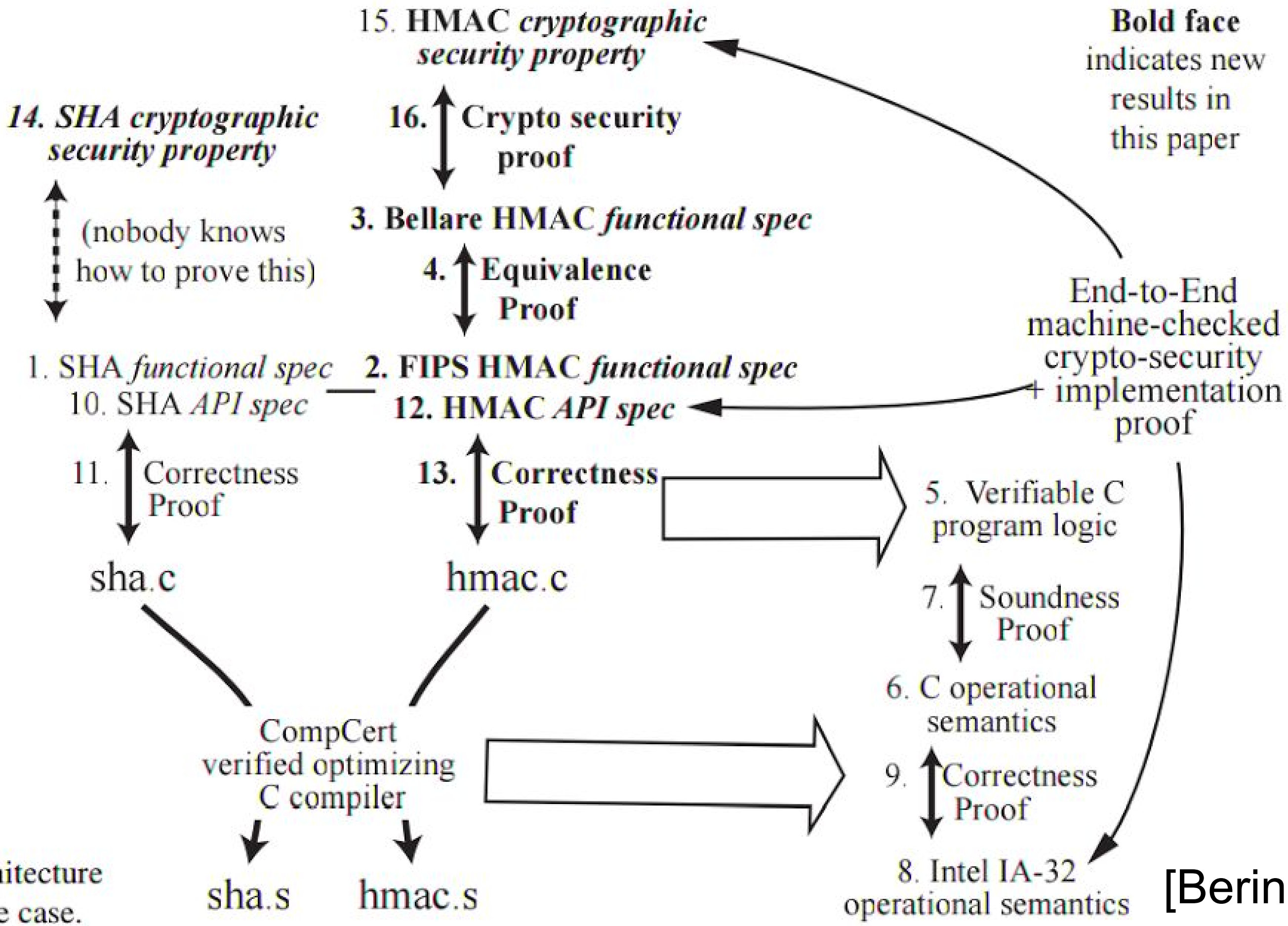
15. **HMAC *cryptographic security property***
Bold face indicates new results in this paper
14. SHA cryptographic security property
16. Crypto security proof
(nobody knows how to prove this)
3. Bellare HMAC *functional spec*
4. Equivalence Proof
End-to-End machine-checked crypto-security + implementation proof
1. SHA *functional spec*
10. SHA *API spec*
2. FIPS HMAC *functional spec*
12. HMAC *API spec*
11. Correctness Proof
13. Correctness Proof
5. Verifiable C program logic
sha.c
hmac.c
7. Soundness Proof
6. C operational semantics
CompCert verified optimizing C compiler
9. Correctness Proof
sha.s
hmac.s
8. Intel IA-32 operational semantics

Figure 1: Architecture of our assurance case.

[Beringer et al.]

FCF, miTLS, curve25519

Oblivc, ObliVM, wysteria, ABY

Pinocchio, Buffet, jSNARK

Hawk

What are the recent exciting results?

How can PL help crypto and vice versa?

Where are we stuck?

Community-wide short-, medium-, long-term agenda?

What do you want to say to NSF?