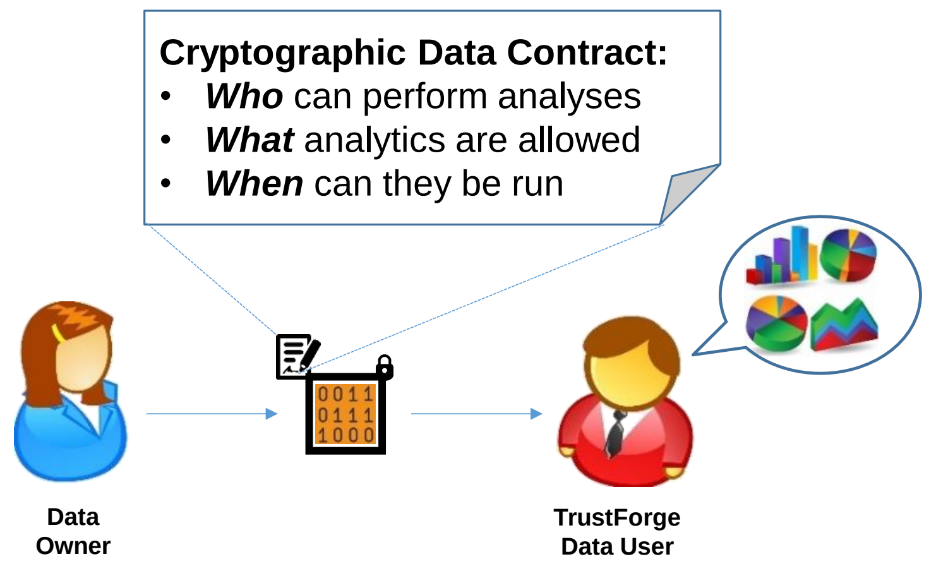
TrustForge: A Cryptographically Secure Enclave

Todd Austin, Valeria Bertacco, Alex Kisil @ Agita Labs

TrustForge Enclave Advances Privacy Tech

- Enables zero-trust data sharing
  - No trust in any system security, software, programmers or IT staff
- Enables privacy-enhanced computation, without a valid data contract:
  - Unauthorized enclaves gen exception
  - Expired contracts gen exception
  - Unauthorized codes destroy data
  - Only authorized results are visible
- Provides hardware capabilities of homomorphic encryption and zero-knowledge proofs
- Available on Amazon AWS and Microsoft Azure



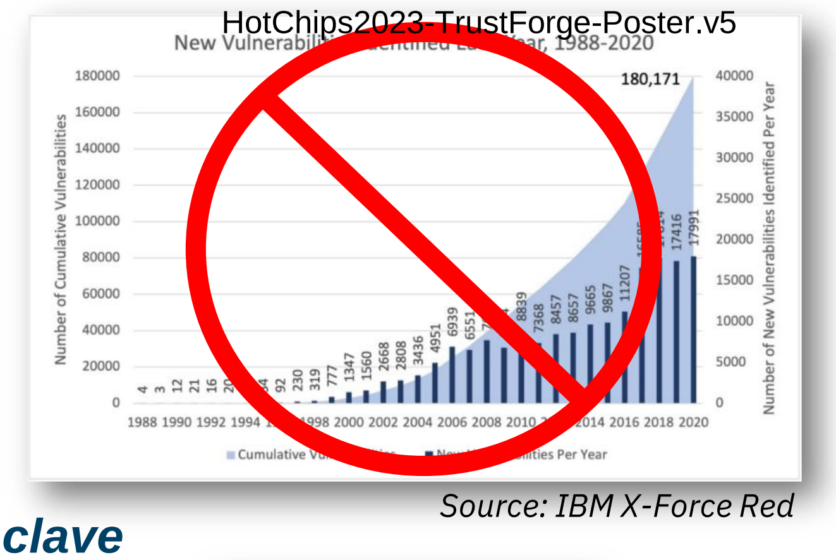
Cryptographic Data Contract:
- ***Who*** can perform analyses
- ***What*** analytics are allowed
- ***When*** can they be run

This 190k-gate functional unit implements always-encrypted computation, nullifying hacking and advancing privacy technology.

A Recipe for Cryptographic TEEs...

1. Eliminate ***all*** software vulnerabilities
   - But all software is (eventually) hackable!
   - Approach: ***no S/W in the enclave***
2. Silence side channels
   - Control, memory, timing, and uArch
   - Approach: ***provably side-channel free enclave***
3. Encrypt sensitive data everywhere else
   - Eliminates trust for all remaining S/W and H/W
   - Approach: ***encrypt on exit of enclave***

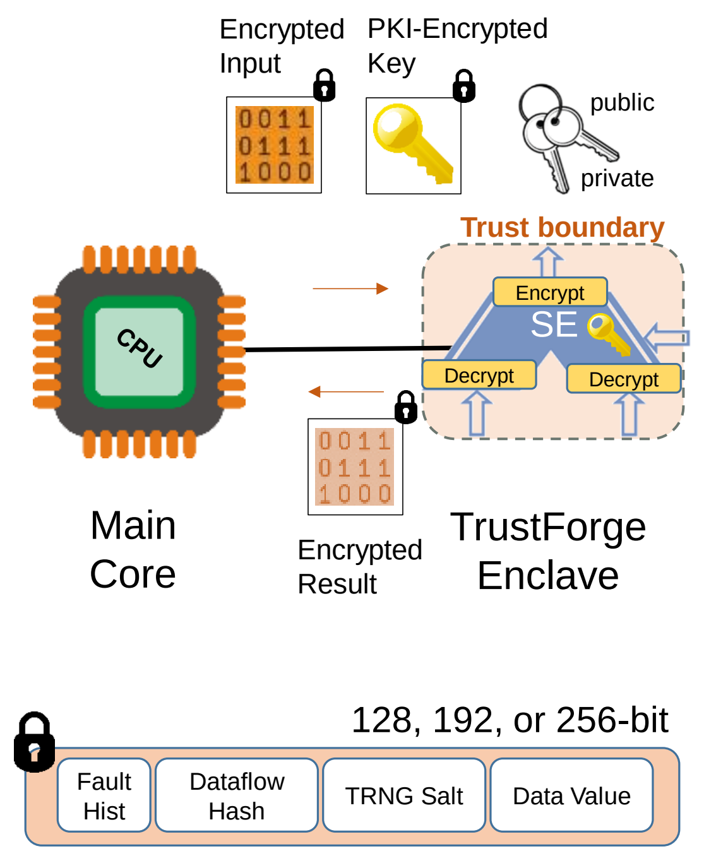
HotChips2023-TrustForge-Poster.v5



Source: IBM X-Force Red

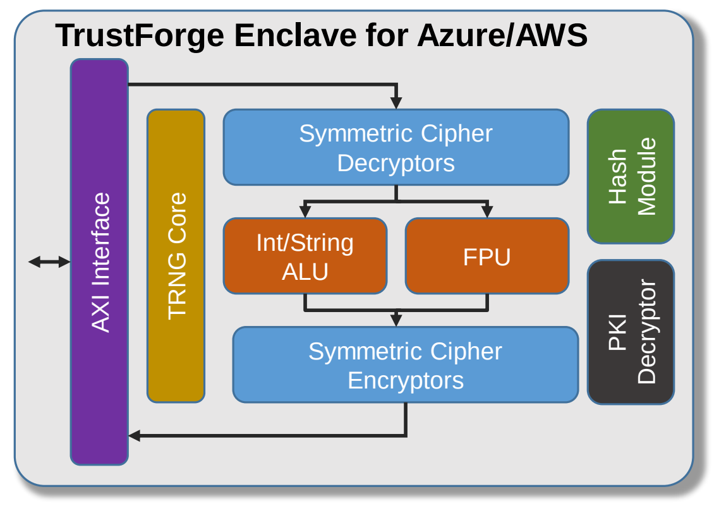
TrustForge: Inside the Zone of Trust

- TrustForge enclave ***exports a public key***, PKI-encrypted keys are decrypted in enclave
- Enclave ***supports RISC-like operations that operate directly on encrypted data***
- Enclave is decrypting, processing, checking, and re-encrypting secret data ***without S/W vulnerabilities or digital side channels***
- Enclave is protected from physical attacks, and ***data is encrypted everywhere else***



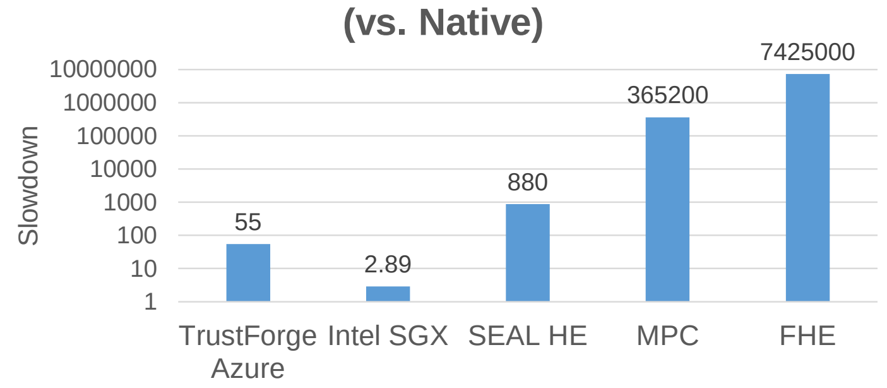
TrustForge Enclave for Azure and AWS

- Deployed on FPGA nodes in the cloud
  - % of total UltraScale+ FPGA used: **~6%, ~190k gates**
  - Logic locked, watermarked and with **forward secrecy**
  - **2-person years effort** to build, fuzz test, and verify
- TrustForge competes with
  - Homomorphic encryption (HE and FHE)
  - Multiparty computation (MPC)
  - **Bottom line: Much faster than comparable frameworks, much more secure than TEEs**

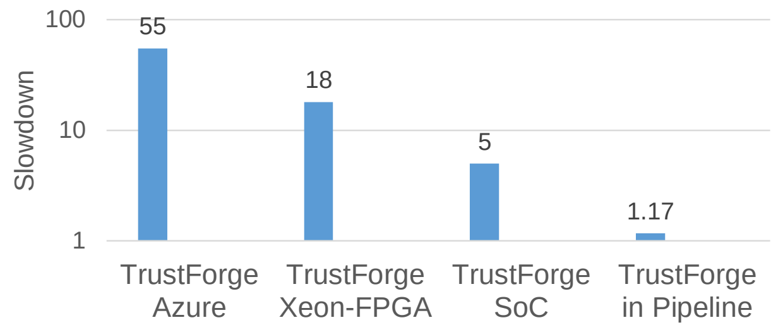


TrustForge Enclave for Azure/AWS

TrustForge Performance Comparison (vs. Native)



TrustForge Performance Headroom (vs. Native)



Use Case: Medical Data Sharing

- Privacy-preserving smartwatch-based heart monitoring
  - ***Data encryption:*** biometric data is encrypted at the sensor, sent to server encrypted
  - ***Encrypted computation:*** analysis of biometric data is performed on encrypted data, without giving access to the server or its operators
  - ***Guardrailing:*** attackers cannot manipulate analysis algorithms to trigger false warnings
  - ***Safe datagrants:*** arrhythmia analysis algorithm can expose potential health warnings to permit notification/text message to the user

Crypto-strength defenses: Not vulnerable to S/W hacking or digital side channels!

Programming for the TrustForge Enclave

- TrustForge extends development language with encrypted variables
- ***Bit-for-bit compatible***, but encrypted
- Support for ***integers, floating-point, Booleans,*** and ***strings***
- Encrypted ***ops return encrypted results***
- Decision processing on encrypted variables implemented with ***CMOV primitive***
- Secret-dependent array indexing implemented with ***ORAM primitives***

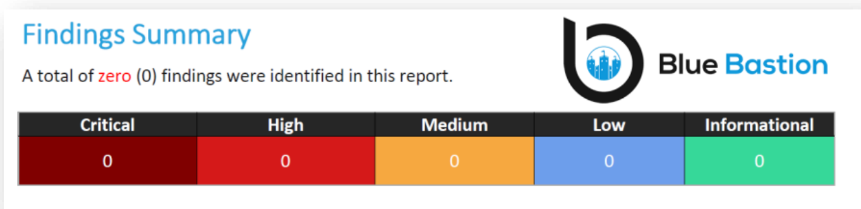
Type Class	C++ Data Types
Integer	enc_int, enc_uint, enc_long, enc_ulong
Floating Point	enc_float, enc_double
Boolean	enc_bool
String/Char	enc_char, enc_char, enc_string

```
x = enc_cmov( secret, x+1, x);
y = enc_cmov(!secret, y+1, y);
```

```
for(int i=0;  i<len(arr);  i++;)
    ret = enc_cmov(i==secret, arr[i], ret);
```

TrustForge Security Analysis

- Worked with In-Q-Tel & Blue Bastion, pen tested for 3 months
  - Red team had full access to all IP
  - ***Zero vulnerabilities found***
- Formal verification with Princeton
  - Secure for ***any program*** on a ***specific implementation***
  - ***Zero vulnerabilities found***
  - Proofs to appear in ACM CCS 2023

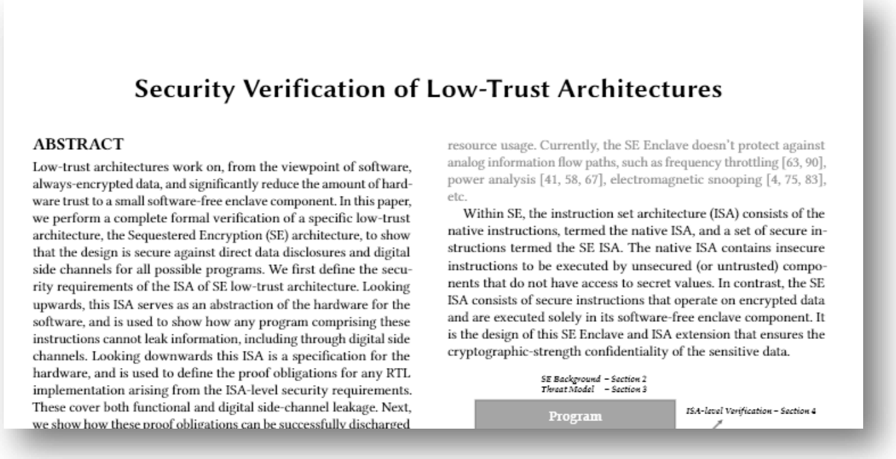


Findings Summary

Blue Bastion

A total of zero (0) findings were identified in this report.

Critical	High	Medium	Low	Informational
0	0	0	0	0



Security Verification of Low-Trust Architectures

ABSTRACT

Low-trust architectures work on, from the viewpoint of software, always-encrypted data, and significantly reduce the amount of hardware trust to a small software-free enclave component. In this paper, we perform a complete formal verification of a specific low-trust architecture, the Sequestered Encryption (SE) architecture, to show that the design is secure against direct data disclosures and digital side channels for all possible programs. We first define the security requirements of the ISA of SE low-trust architecture. Looking upwards, this ISA serves as an abstraction of the hardware for the software, and is used to show how any program comprising these instructions cannot leak information, including through digital side channels. Looking downwards this ISA is a specification for the hardware, and is used to define the proof obligations for any RTL implementation arising from the ISA-level security requirements. These cover both functional and digital side-channel leakage. Next, we show how these proof obligations can be successfully discharged

resource usage. Currently, the SE Enclave doesn't protect against analog information flow paths, such as frequency throttling [63, 90], power analysis [41, 58, 67], electromagnetic snooping [4, 75, 83], etc.

Within SE, the instruction set architecture (ISA) consists of the native instructions, termed the native ISA, and a set of secure instructions termed the SE ISA. The native ISA contains insecure instructions to be executed by unsecured (or untrusted) components that do not have access to secret values. In contrast, the SE ISA consists of secure instructions that operate on encrypted data and are executed solely in its software-free enclave component. It is the design of this SE Enclave and ISA extension that ensures the cryptographic-strength confidentiality of the sensitive data.