Security Verification of Low-Trust Architectures

Qinhan Tan[†*], Yonathan Fisseha[‡*], Shibo Chen[‡*]

Lauren Biernacki[‡], Jean-Baptiste Jeannin[‡], Sharad Malik[†], Todd Austin[‡]

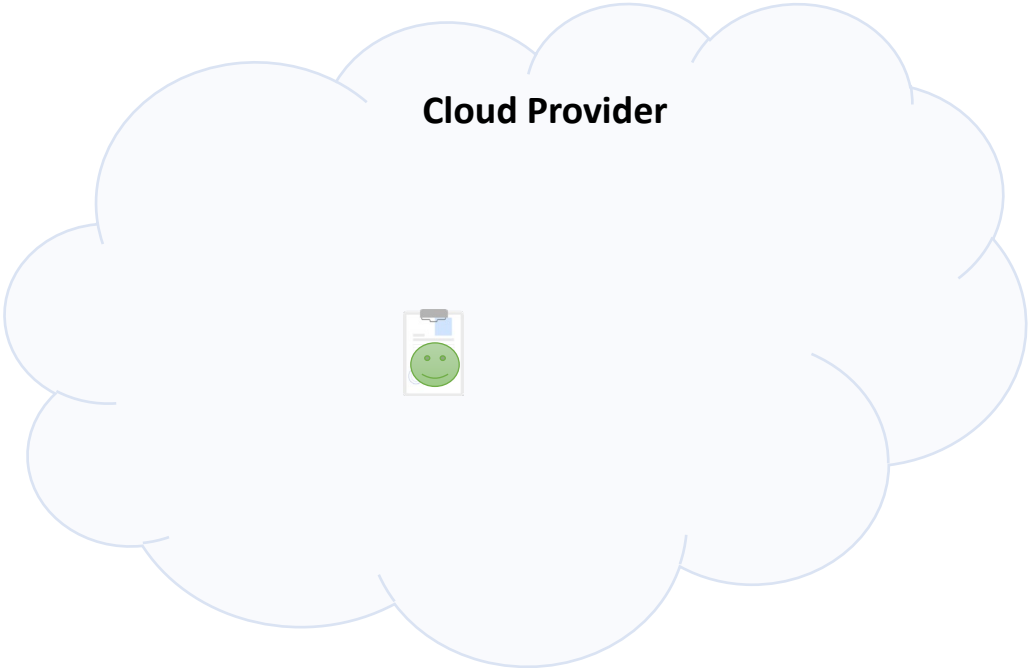
Princeton University[†], University of Michigan Ann Arbor[‡]

Co-First Authors[*]

CCS 2023

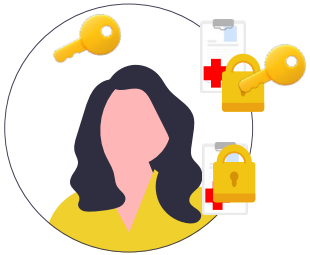
Private Computation in the Cloud?

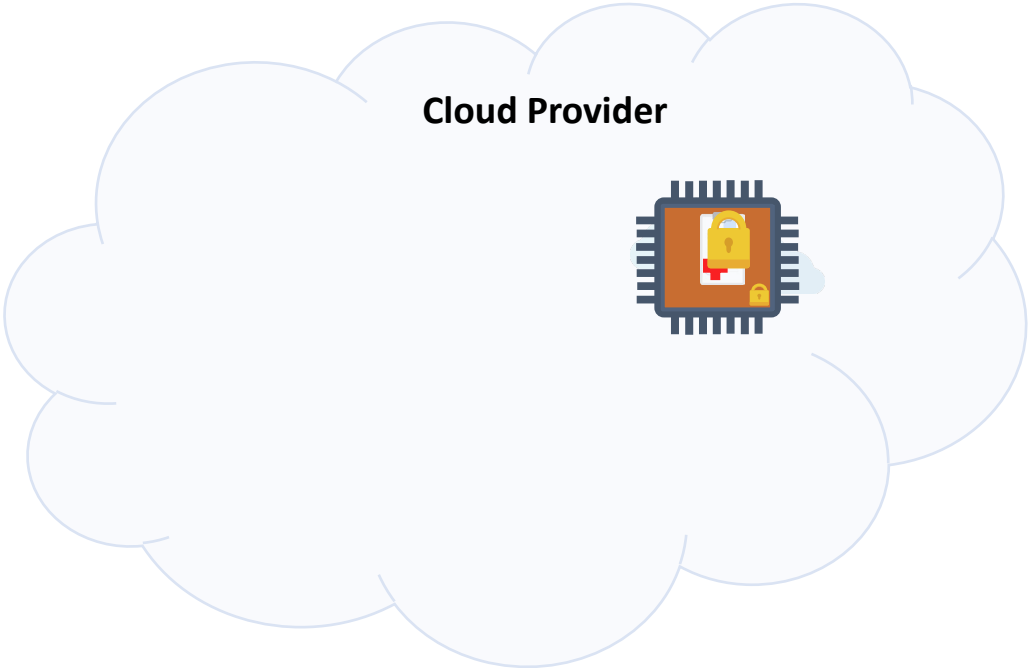
The user must trust the cloud provider with their medical data if they wish to off-load large computations. Encrypting the data at rest is not enough.



Low-Trust Architectures

Minimize trust within the architecture to a small enclave such that the vast majority of the architecture sees only the encrypted data

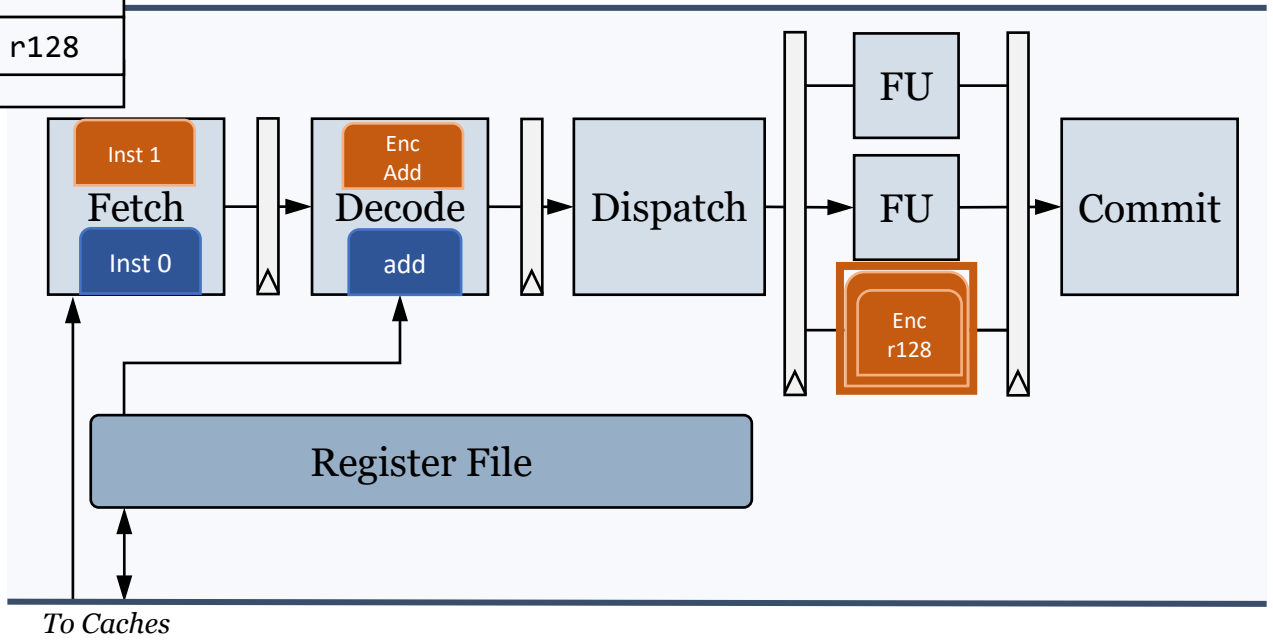




Biernacki, Lauren, Meron Zerihun Demissie, Kidus Birkayehu Workneh, Fitsum Assamnew Andargie, and Todd Austin. "Sequestered Encryption: A Hardware Technique for Comprehensive Data Privacy." In *2022 IEEE International Symposium on Secure and Private Execution Environment Design (SEED)*, pp. 73-84. IEEE, 2022.

Sequestered Encryption

ISA Extension:
`enc_addi  r128, imm`
`enc_and  r128, r128`
`...`

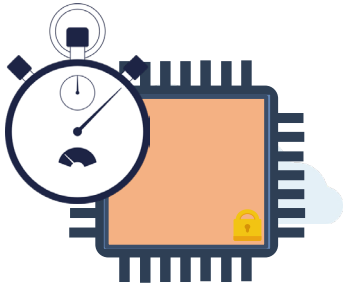


Biernacki, Lauren, Meron Zerihun Demissie, Kidus Birkayehu Workneh, Fitsum Assamnew Andargie, and Todd Austin. "Sequestered Encryption: A Hardware Technique for Comprehensive Data Privacy." In *2022 IEEE International Symposium on Secure and Private Execution Environment Design (SEED)*, pp. 73-84. IEEE, 2022.

Threat Model

Run any program and observe its execution time, control flow, and memory access patterns

Observe and manipulate any software state



Observe hardware signals outside of the enclave, including timing of signals at the interface of the enclave

No post-RTL trojans

No analog attacks

Holistic Security Verification

ISA-Level Proof

Ensure all programs using the ISA preserve privacy

Paper proof based on program semantics and type system

Assumptions about implementation
as proof obligations for RTL-level proof

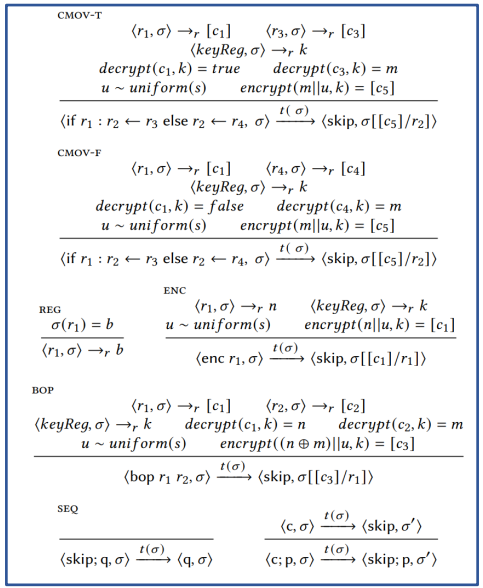
RTL-Level Proof

Ensure implementation satisfies ISA assumptions

Automatic proof based on formal verification

Final Guarantee: No direct/indirect disclosure, or backdoor, or digital side channel for any program on a specific implementation

Program Semantics and Type System



CMOV-T

$$\frac{\langle r_1,\sigma\rangle \to_r [c_1] \quad \langle r_3,\sigma\rangle \to_r [c_3] \quad \langle keyReg,\sigma\rangle \to_r k \quad decrypt(c_1,k) = true \quad decrypt(c_3,k) = m \quad u \sim uniform(s) \quad encrypt(m||u,k) = [c_5]}{\langle \text{if } r_1 : r_2 \leftarrow r_3 \text{ else } r_2 \leftarrow r_4,\ \sigma\rangle \xrightarrow{t(\sigma)} \langle \text{skip}, \sigma[[c_5]/r_2]\rangle}$$

CMOV-F

$$\frac{\langle r_1,\sigma\rangle \to_r [c_1] \quad \langle r_4,\sigma\rangle \to_r [c_4] \quad \langle keyReg,\sigma\rangle \to_r k \quad decrypt(c_1,k) = false \quad decrypt(c_4,k) = m \quad u \sim uniform(s) \quad encrypt(m||u,k) = [c_5]}{\langle \text{if } r_1 : r_2 \leftarrow r_3 \text{ else } r_2 \leftarrow r_4,\ \sigma\rangle \xrightarrow{t(\sigma)} \langle \text{skip}, \sigma[[c_5]/r_2]\rangle}$$

REG

$$\frac{\sigma(r_1) = b}{\langle r_1,\sigma\rangle \to_r b}$$

ENC

$$\frac{\langle r_1,\sigma\rangle \to_r n \quad \langle keyReg,\sigma\rangle \to_r k \quad u \sim uniform(s) \quad encrypt(n||u,k) = [c_1]}{\langle \text{enc } r_1,\sigma\rangle \xrightarrow{t(\sigma)} \langle \text{skip}, \sigma[[c_1]/r_1]\rangle}$$

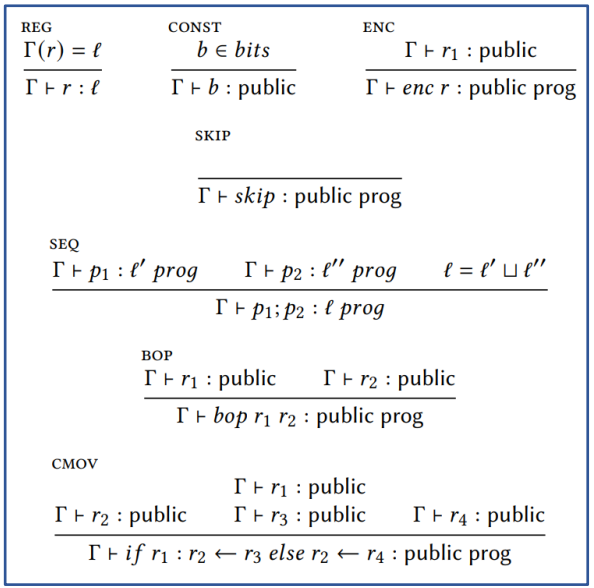
BOP

$$\frac{\langle r_1,\sigma\rangle \to_r [c_1] \quad \langle r_2,\sigma\rangle \to_r [c_2] \quad \langle keyReg,\sigma\rangle \to_r k \quad decrypt(c_1,k) = n \quad decrypt(c_2,k) = m \quad u \sim uniform(s) \quad encrypt((n \oplus m)||u,k) = [c_3]}{\langle \text{bop } r_1\ r_2,\sigma\rangle \xrightarrow{t(\sigma)} \langle \text{skip}, \sigma[[c_3]/r_1]\rangle}$$

SEQ

$$\frac{}{\langle \text{skip; q},\sigma\rangle \xrightarrow{t(\sigma)} \langle \text{q},\sigma\rangle} \qquad \frac{\langle \text{c},\sigma\rangle \xrightarrow{t(\sigma)} \langle \text{skip},\sigma'\rangle}{\langle \text{c; p},\sigma\rangle \xrightarrow{t(\sigma)} \langle \text{skip; p},\sigma'\rangle}$$

Small-step semantics



REG

$$\frac{\Gamma(r) = \ell}{\Gamma \vdash r : \ell}$$

CONST

$$\frac{b \in bits}{\Gamma \vdash b : \text{public}}$$

ENC

$$\frac{\Gamma \vdash r_1 : \text{public}}{\Gamma \vdash enc\ r : \text{public prog}}$$

SKIP

$$\frac{}{\Gamma \vdash skip : \text{public prog}}$$

SEQ

$$\frac{\Gamma \vdash p_1 : \ell'\ prog \quad \Gamma \vdash p_2 : \ell''\ prog \quad \ell = \ell' \sqcup \ell''}{\Gamma \vdash p_1; p_2 : \ell\ prog}$$

BOP

$$\frac{\Gamma \vdash r_1 : \text{public} \quad \Gamma \vdash r_2 : \text{public}}{\Gamma \vdash bop\ r_1\ r_2 : \text{public prog}}$$

CMOV

$$\frac{\Gamma \vdash r_1 : \text{public} \quad \Gamma \vdash r_2 : \text{public} \quad \Gamma \vdash r_3 : \text{public} \quad \Gamma \vdash r_4 : \text{public}}{\Gamma \vdash if\ r_1 : r_2 \leftarrow r_3\ else\ r_2 \leftarrow r_4 : \text{public prog}}$$

Security type system

Theorem 4.2 (Soundness). If

(1) $\Gamma \vdash c : \ell$

(2) $\Gamma \vdash \sigma_1 \approx_l \sigma_2$

(3) $\langle c, \sigma_1\rangle \xrightarrow{n}_* \langle \text{skip}, \sigma_1'\rangle$

(4) $\langle c, \sigma_2\rangle \xrightarrow{m}_* \langle \text{skip}, \sigma_2'\rangle$

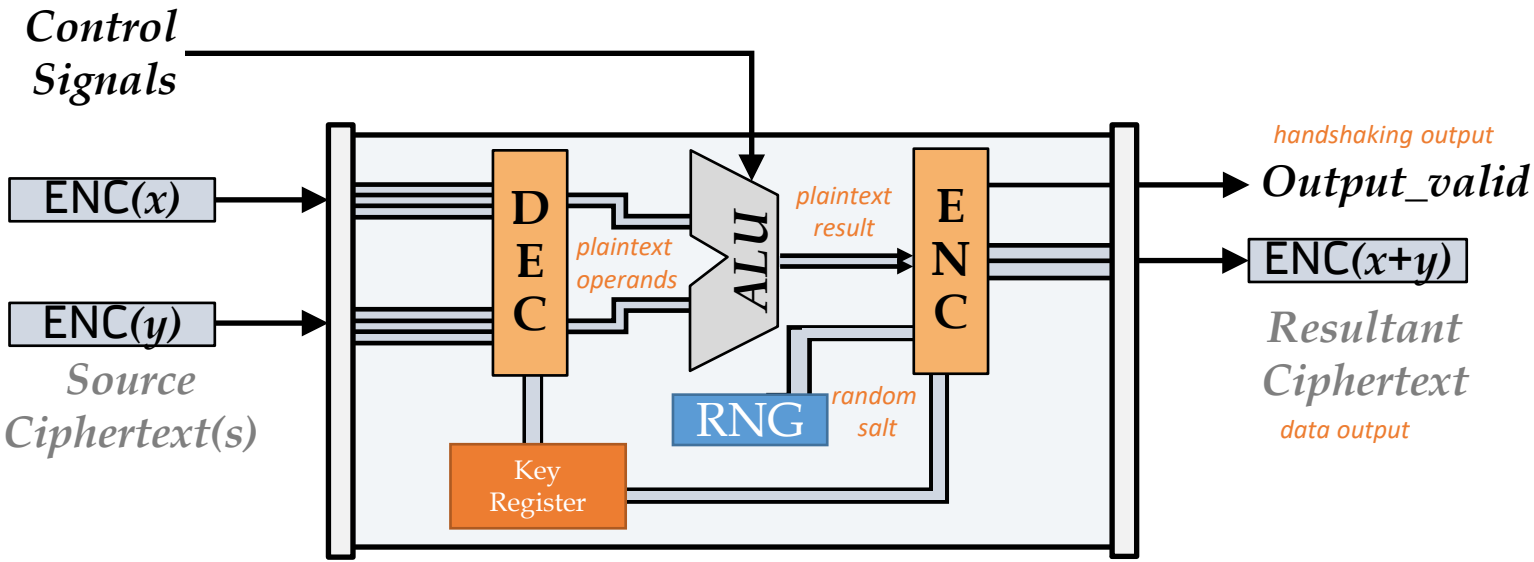
(5) $dom(\Gamma) = dom(\sigma_1) = dom(\sigma_2)$

then we have $\Gamma \vdash \sigma_1' \approx_l \sigma_2'$ and $n = m$.

Non-interference theorem

Assumptions discharged as proof obligations

SE Enclave Implementation

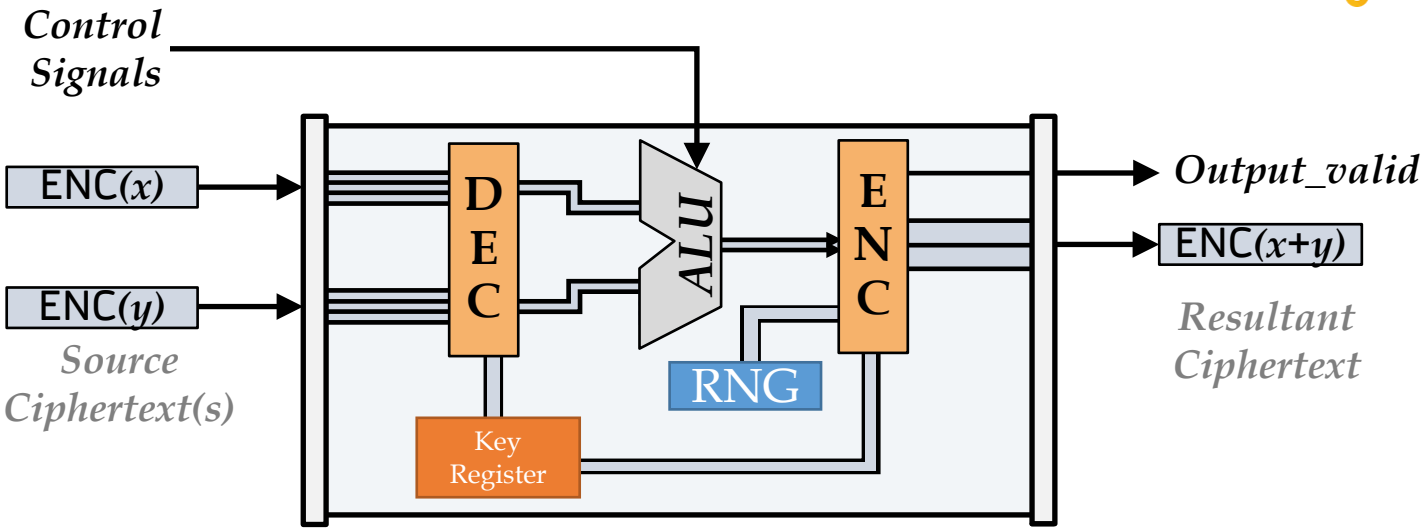


- ENC and DEC apply 10-round AES

RTL Proof Obligations Discharged from ISA Level

1. Encryption has enough randomness

Reason: ISA semantics require a pseudo-random distribution

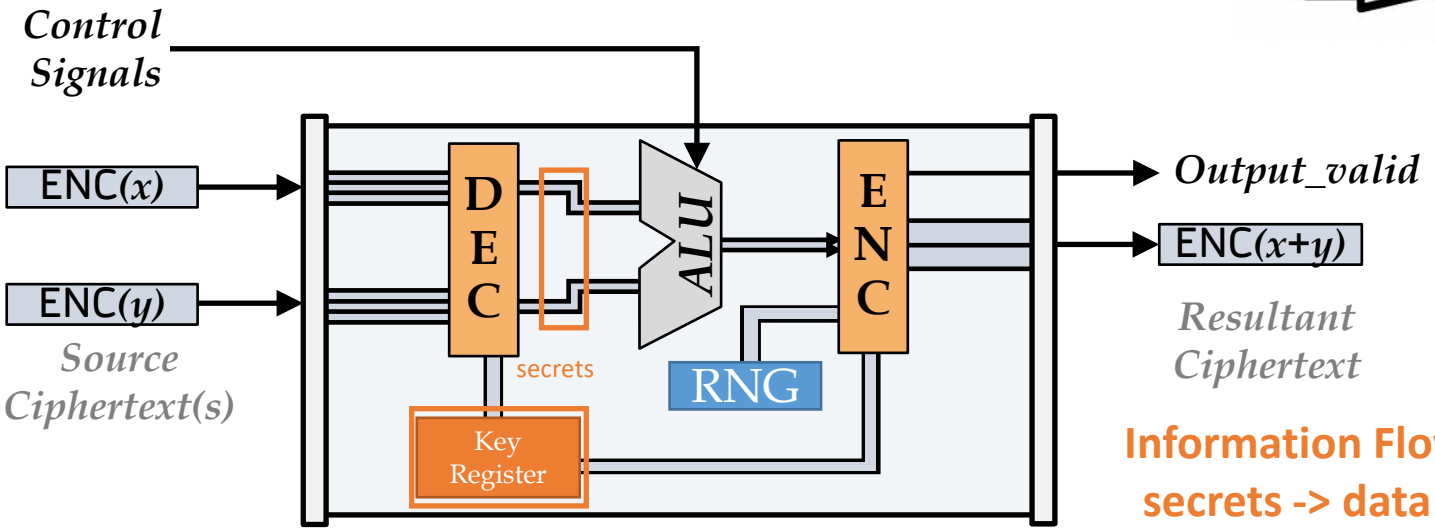


Functional Verification for RNG

RTL Proof Obligations Discharged from ISA Level

2. No functional leakage

Reason: type system assumes all instruction outputs have low security labels

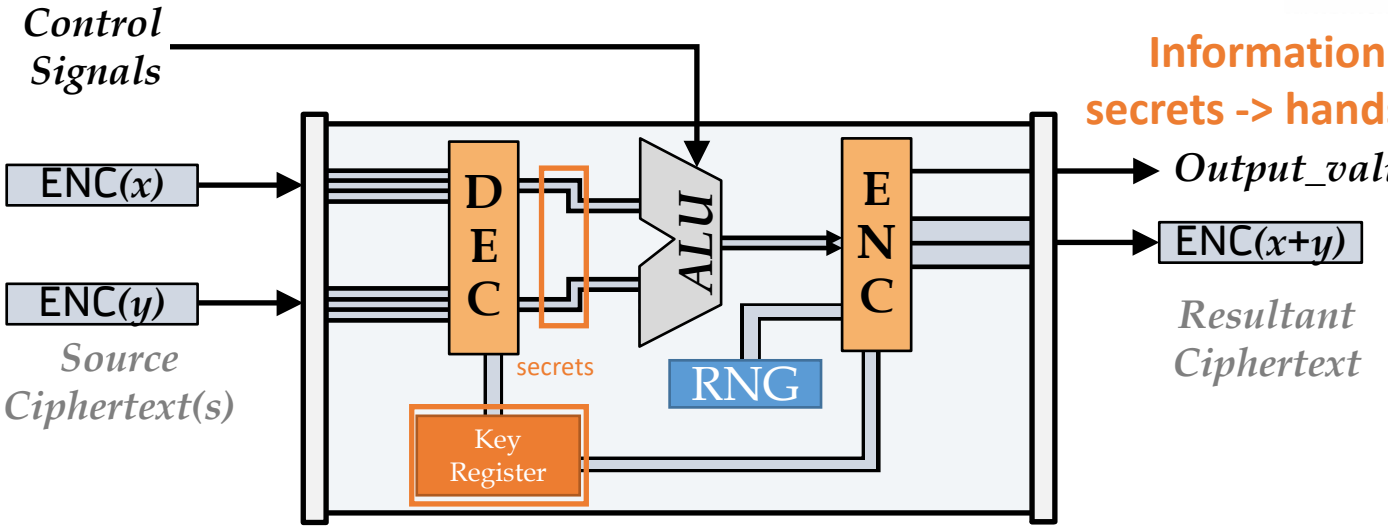


Information Flow Check
secrets -> data output

RTL Proof Obligations Discharged from ISA Level

3. No timing leakage

Reason: ISA semantics assume secret-independent transition time of instructions

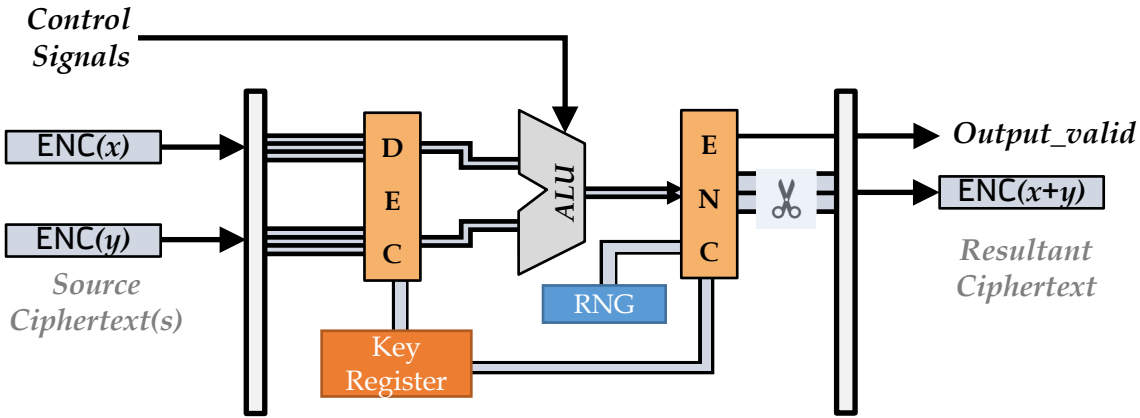


Information Flow Check
secrets -> handshaking output

Declassification

- Information flow between data output and secrets is safe after being fully encrypted but will be captured

- Need to declassify such information flow
  - Cut off the fully-encrypted ciphertext after the encryption unit



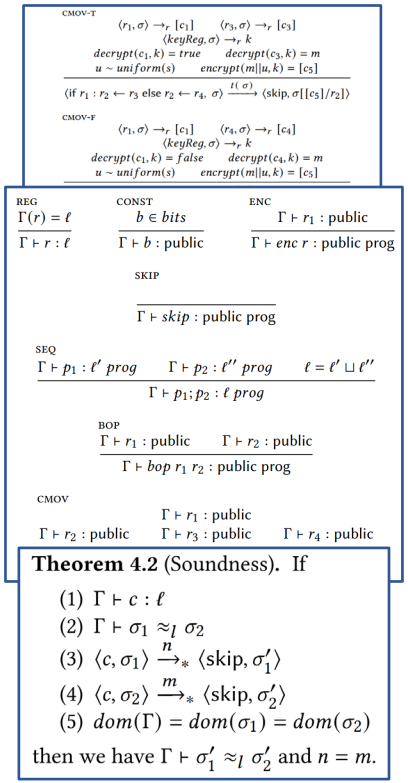
Experiment

Setup	Processor: two Intel Xeon 5222 cores Tool: Cadence JasperGold 2021

SE Variant	Register Bits	Result	Leakage	Time	Memory
Default	6544	secure	no leakage	0.1s	1.6GB
Rolled AES	1412	secure	no leakage	0.1s	0.7GB
Cache	12784	secure	no leakage	0.1s	1.6GB
Vulnerable Rolled AES	1412	insecure	functional (plaintext→data, key→data)	109.4s	2.5GB
Vulnerable Multiplier	6737	insecure	timing (plaintext→output_valid)	63.3s	4.7GB
Vulnerable Cache	12752	insecure	timing (plaintext→output_valid)	402.4s	14.7GB
Vulnerable RSA	4328	insecure	timing, functional (key→output_valid, key→data)	0.1s	0.3GB

End-to-End Program-to-Implementation Security Verification of Low-Trust SE Architecture

ISA-Level: Security Guarantee for any Programs



CMOV-T

$$\frac{\langle r_1,\sigma\rangle \to_r [c_1] \quad \langle r_3,\sigma\rangle \to_r [c_3] \quad \langle keyReg,\sigma\rangle \to_r k \quad decrypt(c_1,k)=true \quad decrypt(c_3,k)=m \quad u \sim uniform(s) \quad encrypt(m\|u,k)=[c_5]}{\langle \text{if } r_1 : r_2 \leftarrow r_3 \text{ else } r_2 \leftarrow r_4,\ \sigma\rangle \xrightarrow{t(\sigma)} \langle \text{skip}, \sigma[[c_5]/r_2]\rangle}$$

CMOV-F

$$\frac{\langle r_1,\sigma\rangle \to_r [c_1] \quad \langle r_4,\sigma\rangle \to_r [c_4] \quad \langle keyReg,\sigma\rangle \to_r k \quad decrypt(c_1,k)=false \quad decrypt(c_4,k)=m \quad u \sim uniform(s) \quad encrypt(m\|u,k)=[c_5]}{\langle \text{if } r_1 : r_2 \leftarrow r_3 \text{ else } r_2 \leftarrow r_4,\ \sigma\rangle \xrightarrow{t(\sigma)} \langle \text{skip}, \sigma[[c_5]/r_2]\rangle}$$

REG

$$\frac{\Gamma(r)=\ell}{\Gamma \vdash r : \ell}$$

CONST

$$\frac{b \in bits}{\Gamma \vdash b : \text{public}}$$

ENC

$$\frac{\Gamma \vdash r_1 : \text{public}}{\Gamma \vdash enc\ r : \text{public prog}}$$

SKIP

$$\frac{}{\Gamma \vdash skip : \text{public prog}}$$

SEQ

$$\frac{\Gamma \vdash p_1 : \ell'\ prog \quad \Gamma \vdash p_2 : \ell''\ prog \quad \ell = \ell' \sqcup \ell''}{\Gamma \vdash p_1; p_2 : \ell\ prog}$$

BOP

$$\frac{\Gamma \vdash r_1 : \text{public} \quad \Gamma \vdash r_2 : \text{public}}{\Gamma \vdash bop\ r_1\ r_2 : \text{public prog}}$$

CMOV

$$\frac{\Gamma \vdash r_1 : \text{public} \quad \Gamma \vdash r_2 : \text{public} \quad \Gamma \vdash r_3 : \text{public} \quad \Gamma \vdash r_4 : \text{public}}{}$$

Theorem 4.2 (Soundness). If

(1) $\Gamma \vdash c : \ell$
(2) $\Gamma \vdash \sigma_1 \approx_l \sigma_2$
(3) $\langle c, \sigma_1\rangle \xrightarrow{n}_* \langle \text{skip}, \sigma_1'\rangle$
(4) $\langle c, \sigma_2\rangle \xrightarrow{m}_* \langle \text{skip}, \sigma_2'\rangle$
(5) $dom(\Gamma) = dom(\sigma_1) = dom(\sigma_2)$

then we have $\Gamma \vdash \sigma_1' \approx_l \sigma_2'$ and $n = m$.

Discharge Proof Obligations

SE Variant	Register Bits	Result	Leakage	Time	Memory
Default	6544	secure	no leakage	0.1s	1.6GB
Rolled AES	1412	secure	no leakage	0.1s	0.7GB
Cache	12784	secure	no leakage	0.1s	1.6GB
Vulnerable Rolled AES	1412	insecure	functional (plaintext→data, key→data)	109.4s	2.5GB
Vulnerable Multiplier	6737	insecure	timing (plaintext→output_valid)	63.3s	4.7GB
Vulnerable Cache	12752	insecure	timing (plaintext→output_valid)	402.4s	14.7GB
Vulnerable RSA	4328	insecure	timing, functional (key→output_valid, key→data)	0.1s	0.3GB

RTL-Level: Security Guarantee for Specific Implementations

Final Guarantee: No direct/indirect disclosure, or backdoor, or digital side channel for any program on a specific implementation