1. Introduction

This document provides a mathematical proof for the **security** of the SECURA application programming interface (API). SECURA provides non-computational information security for data at rest, cradle-to-grave file control, and ransomware resilience. SECURA is immune from classical and quantum attack.

This paper answers a single question: is SECURA actually secure? Although the paper requires an understanding of mathematical proofs, it is written to be as accessible as possible. The authors note that students of mathematics and cryptography may want to prove or derive different properties from SECURA, such as its homomorphic properties. For those students, we recommend the resources in Section 7.

While SECURA adds interoperability features to shares through a peer-to-peer distributed filing sharing system and share management and control through a distributed ledger, the proof for the security of SECURA is essentially the proof for Shamir Secret Sharing, invented by Dr. Adi Shamir while at the Massachusetts Institute of Technology. Nothing in this document is intended to suggest that the authors invented, supplemented (other than the mitigations in Section 5), or in any way contributed to Adi Shamir's work.

2. Preliminaries

2.1. SECURA parameters

An application using SECURA must decide on several parameters:

Name	Definition
Ring	Ring R upon which all polynomials R[x] will be evaluated
Polynomial	Polynomial P in R[x]
Vector	Vector Y in element of R^n where all values of Y are different and nonzero
Shares	A non-zero value x of a finite field of cardinal c, such that x is less than the number of c

3. Definition

For any secret s, algorithm S defines a x out of n threshold secret sharing scheme if it computes $S(s) = [s_1, \dots, s_n]$ and the following conditions hold:

1. Correctness: s is uniquely determined by any x shares from $\{s_1, \dots, s_n\}$ and there exists an algorithm S that efficiently computes s from these x shares.

2. Privacy: having access t any x-1 share from $\{s_1, \dots, s_n\}$ gives no information about the value of s.

4. The SECURA Proof

The SECURA proof is the proof of Shamir Secret Sharing.

The security of SECURA relies in the fact that, given a random set of at least two points x on a finite field, there is a unique polynomial that passes though all x and all n shares are equally distributed.

Given ring R, vector $v \in R^n$, polynomial P(x):

$eval(P) := [P(v_0), \ldots, P(v_n)]^T$

For any two polynomials P and Y in R[x] and scalar value $r \in R$, the following conditions hold:

1) Additivity: eval(P + Y) = eval(P) + eval(Y)

2) Multiplicativity: eval(P * Y) = eval(P) * eval(Y)

3) Multiplicativity wlog scalar values: eval(r * P) = r * eval(P)

The duality of the respective polynomial operations proves:

1) Additivity: (P + Y)(v) = P(v) + Y(v)

2) Multiplicativity: (P * Y)(v) = P(v) * Y(v)

3) Multiplicativity wlog scalar values: (r * P)(v) = r * P(v)

Therefore, eval is a linear mapping between the evaluation positions of the polynomial and result vector.

Given the vector of n shares $\vec{y}$, $\vec{P} = [P_0, \ldots, P_k]$, and $P_0 = x$:

$\vec{y} = 1P_0 + AP`$

where

$\vec{P}$ = [P_1, …, P_n]^T

Although not stated above, index i must be stored together with share s_i to recompute the original value.

The above shows that the combination of n shares are from a uniform distribution and therefore give no information about the secret value. While a theoretical adversary can guess the missing share, all guesses are equally probable.

Since all shares, up to the threshold limit, are required to recreate the secret, information about any shares below the threshold limit, including complete information about those shares, does not reveal the secret (see Section 5). Mathematically, against an active adversary, the math above holds as long as the number of compromised shares is less than $\frac{n}{3}$.

For ransomware resilience, the number of shares required to reconstruct the secret must be less than the total number of shares, and must be stored separately across independent endpoints. Wlog, the threshold for ransomware resilience *r* is correlated to the number of independent endpoints storing shares: r = $\frac{n}{3}$.

5. Vulnerabilities/Mitigations

While each share in SECURA is secure from any computational decryption, partial compromise of all shares can lead to secret reconstruction. We mitigate this vulnerability by implementing modulus and evaluation places that mitigate derandomization of

Monte-Carlo constructions. We apply the work of Maji et. all to implement Shamir Secret Sharing using evaluation places that are secure against physical bit leakage.

Additionally, before the secret in SECURA can be read, it must be reassembled in a single location. A theoretical attacker could target this single location and decrypt the shared secret. SECURA is built under the premise of a persistent advanced threat to your information. Consequently, to mitigate this (albeit remote) threat to your information, information is post-quantum encrypted before being sharded (see our paper on the CLCKD protocol for a description of our post-quantum encryption). We assume that all computational encryption fails given enough time and pressure. Consequently, in addition to information in SECURA being computationally encrypted prior to sharding, shares are only reconstructed in memory for a short period of time (long enough to be read by the user) and are then destroyed.

6. IPR

7. Acknowledgements

Thank you to Dr. Adi Shamir for inventing Shamir Secret Sharing, upon which SECURA is based. Thank you to Paul C. Kocher for his discussion on leakage attacks, and to Venkatesan Guruswami and Mary Wootters for demonstrating how to reconstruct a secret using a leakage attack. Thank you to Hemanta K. Maji, Hai H. Nguyen, Anat Paskin-Cherniavsky, Tom Suad, and Mingyuan Wang for their insight into evaluation places that are secure against physical bit leakage. Thank you to Dan Bogdanov for

explaining how to securely performing computations on secret shares.

Dr. Shamir's paper, How to Share a Secret, is available through the Massachusetts Institute of Technology at https://web.mit.edu/6.857/OldStuff/Fall03/ref/Shamir-HowToShareASecret.pdf.