ZEKROM

A LIBRARY OF ARITHMETIZATION-ORIENTED CONSTRUCTIONS FOR ZK-SNARK CIRCUITS

LAURENT THOENY

CYSEC

ANTONIO DE LA PIEDRA

KUDELSKI SECURITY

PASS THE SALT 2023

- ZEKROM IS A LIBRARY FOR BUILDING PRIVACY-PRESERVING APPLICATIONS BASED ON ZK-SNARKS
- IT PROVIDES STATE-OF-THE-ART AUTHENTICATED-ENCRYPTION AND HASHING FUNCTIONS FOR ZK-APPS.
- IT RELIES ON THE RECENTLY PROPOSED SAFE API
- IT IS EASY TO EXTEND
- WRITTEN IN RUST

AGENDA

01

WHAT ARE ZK-SNARKS ?

02

HOW ENCRYPTION AND HASHING ARE USED INSIDE A ZK-SNARK ?

03

HOW CAN I USE ZK-SNARKS INSIDE MY PROJECT ?

04

ZEKROM

1. WHAT ARE ZK-SNARKS ?

ZK-SNARKS

- **Z**ERO-**K**NOWLEDGE **S**UCCINCT **N**ON-INTERACTIVE **AR**GUMENTS OF **K**NOWLEDGE
- THEY ARE A POWERFUL BUILDING BLOCK FOR CREATING PRIVACY-PRESERVING APPLICATIONS. MOST RECENT DEVELOPMENTS IN THE LAST 7-8 YEARS
- THEY ALLOW US TO CONVINCE SOMEONE THAT A PARTICULAR STATEMENT IS TRUE WITHOUT REVEALING ANY INFORMATION ABOUT IT
- **KEY APPLICATIONS**: ACCOUNTABILITY AND COMPLIANCE OF A PARTY NEED TO BE DEMONSTRATED

WHAT CAN WE BUILD WITH ZK-SNARKS ?

EXAMPLES

COMMON USE-CASES

- I KNOW THE PRE-IMAGE OF SHA3(**M**)
- I HAVE THE SECRET KEY **SK** CORRESPONDING TO THIS PUBLIC KEY PK.
- I HAVE CORRECTLY PERFORMED THE SIGNATURE ALGORITHM USING MY SECRET KEY **SK**.
- **MY IDENTITY** IS PART OF THE LIST OF AUTHORIZED PEERS
- VALIDATION OF SOME **ATTRIBUTES** E.G. MY AGE IS > 18, I LIVE IN A SET OF VALID COUNTRIES ACCORDING TO THE APPLICATION, MY NAME AND SURNAME ARE WELL-FORMED.

THE SECRET VALUES ARE NEVER REVEALED

EXAMPLES

COMMON USE-CASES

- **I KNOW A SECRET**: I KNOW THE PRE-IMAGE OF SHA3(**M**)
- **PROOF OF CORRECT EXECUTION**: I HAVE THE SECRET KEY **SK** CORRESPONDING TO THIS PUBLIC KEY PK. I HAVE CORRECTLY PERFORMED THE SIGNATURE ALGORITHM USING MY SECRET KEY **SK**.
- **PRIVATE AUTHENTICATION**: **MY IDENTITY** IS PART OF THE LIST OF AUTHORIZED PEERS
- VALIDATION OF SOME **ATTRIBUTES** E.G. MY AGE IS > 18, I LIVE IN A SET OF VALID COUNTRIES ACCORDING TO THE APPLICATION, MY NAME AND SURNAME ARE WELL-FORMED.

THE SECRET VALUES ARE NEVER REVEALED

RECENT PROPOSALS

- PROVING THAT A CERTAIN VULNERABILITY EXISTS **[GREEN ET AL., 2022]**
- PROVING THAT NETWORK TRAFFIC IS ENCRYPTED ACCORDING TO POLICY **[GRUBBS ET AL, 2021]**
- PROVING PROPERTIES ABOUT INFERENCE MODELS IN MACHINE LEARNING
- FIGHTING DISINFORMATION THROUGH PROVING AN IMAGE ORIGIN AND TRANSFORMATIONS **[BONEH ET AL., 2023]**

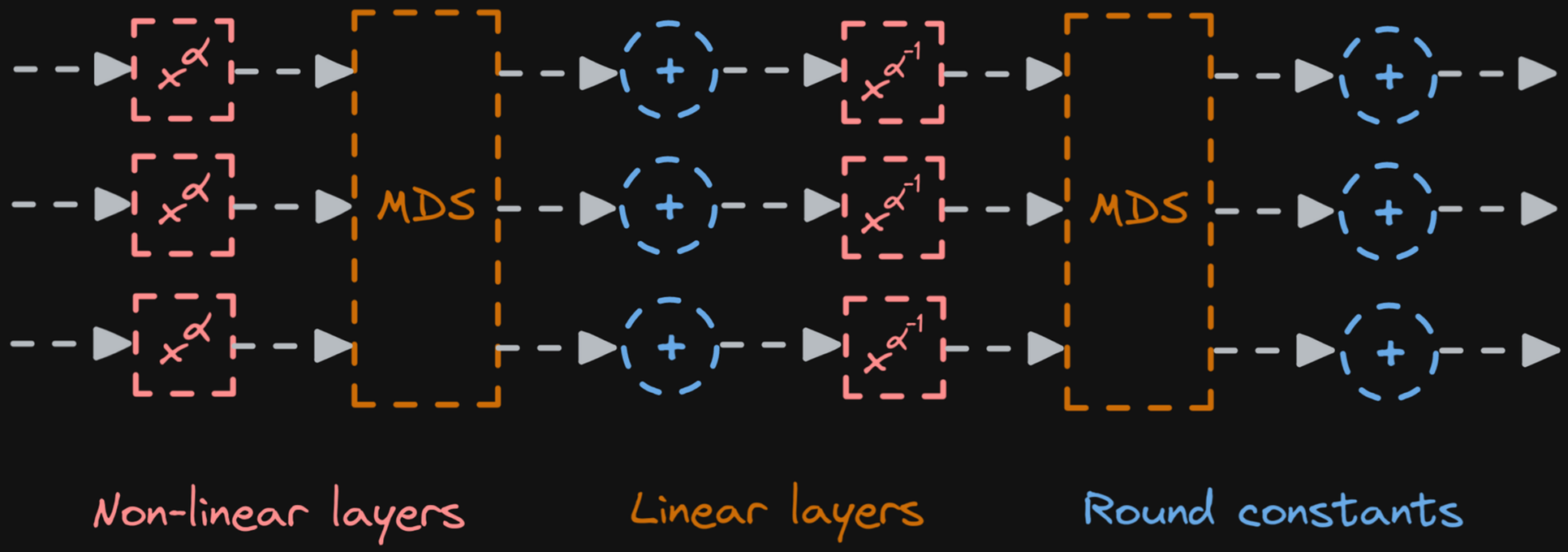
2. USING AUTHENTICATED ENCRYPTION AND HASHING

HASHING AND ENCRYPTION

- MANY OF THE USE-CASES REQUIRE AEAD/HASHING AS THE UNDERLYING CRYPTOGRAPHIC PRIMITIVE
- HOWEVER, IN PROVING SCHEMES WE DO COMPUTATIONS WITH FIELD ELEMENTS BY DESIGN, AND MEASURE THE PERFORMANCE IN TERMS OF ADD/MUL OPERATIONS IN THE COMPUTATION
- THIS MEANS THAT MANY TRADITIONAL SOLUTION (E.G. SHA2, AES) WILL NOT PERFORM WELL IN CIRCUITS. THEIR DESIGN RELY ON COSTLY OPERATION (TYPICALLY BITWISE OPERATIONS, LIKE XORING)

HASHING AND ENCRYPTION

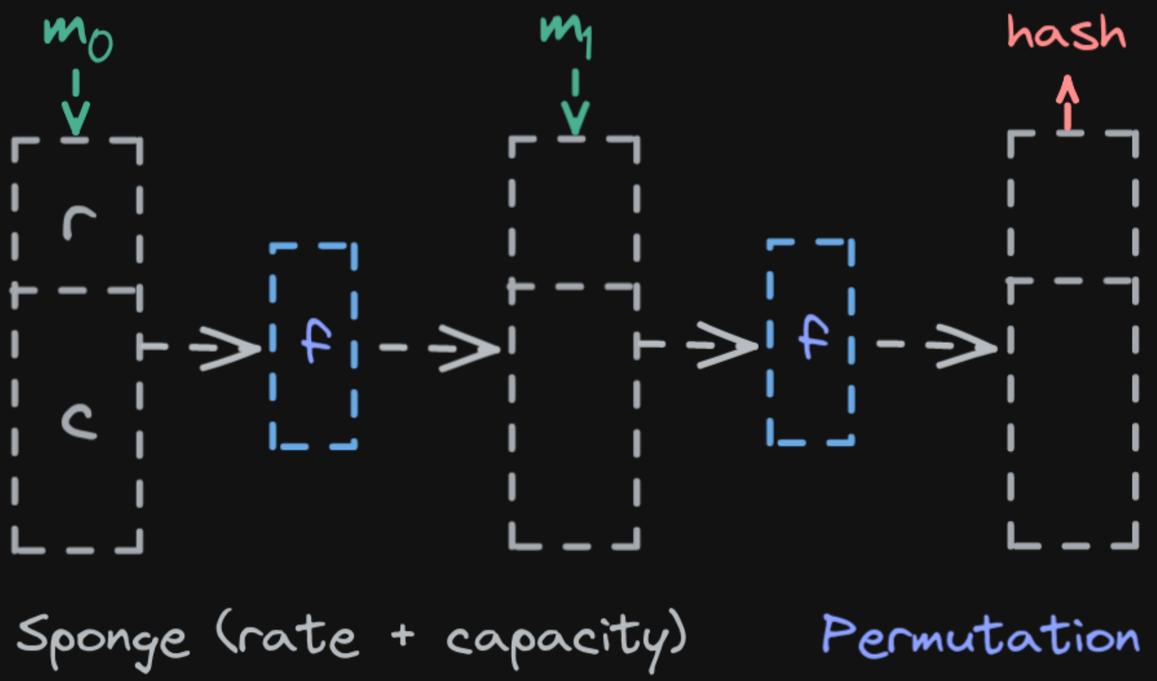
ONE SOLUTION IS **ARITHMETIZATION-ORIENTED CONSTRUCTIONS**, NEW PERMUTATIONS DESIGNED TO USE FIELD ELEMENTS (E.G RESCUE-PRIME*)



* CREDITS TO THE RESCUE-PRIME SPECIFICATION BY ALAN SZEPIENIEC, TOMER ASHUR AND SIEMEN DHOOGHE

HASHING AND ENCRYPTION

THEN, USING THE **SAFE API*** ALLOWS US TO TRANSFORM SAID PERMUTATIONS TO DIFFERENT CRYPTOGRAPHIC PRIMITIVES (AEAD, HASHING, MERKLE TREE, ETC.)



* SAFE (SPONGE API FOR FIELD ELEMENTS) - PROPOSAL BY DMITRY KHOVRATOVICH ET. AL.

3. HOW CAN I USE ZK-SNARKS IN MY PROJECT ?

DESIGN THE APPLICATION

PICK YOUR APPROACH

THE HARD WAY (WRITE THE CIRCUIT)

POSSIBLE WITH A DSL (CIRCOM, LEO)

OR A LOWER LEVEL LIBRARY

VS

USE AN EXISTING LIBRARY GADGET

EASIER WITHOUT PRIOR KNOWLEDGE

DONE IN CIRCOMLIB, HALO2, **ZEKROM**

DESCRIBE YOUR APP

PRIVATE AND PUBLIC INPUTS

CONSTANTS AND OUTPUTS

OPERATIONS (ADD/MUL), CONSTRAINTS

CHOOSE A PROVING SCHEME

SCHEME	VERIFY TIME	TRUSTED SETUP
GROTH 16	Constant, few ms	Per circuit
PLONK (2019)	Constant, few ms	Universal
Marlin (2019)	Constant, few ms	Universal
Halo2 (2020)	Variable, but in general short	Universal

ENJOY !

NOW USE THE API IN YOUR APP

PERFORM THE SETUP PHASE IF NEEDED

THEN PROVING AND VERIFYING STEPS

UNDER THE HOOD

ARITHMETIC CIRCUIT DEFINITION

- WE TRANSFORM OUR APPLICATION IN AN ARITHMETIC CIRCUIT WITH PUBLIC INPUTS (X), PRIVATE INPUTS (WITNESSES), GATES (ADD, MUL, CUSTOM), WIRES AND PUBLIC OUTPUTS.
- WE DECIDE HOW TO CONSTRAINT THE CIRCUIT E.G. A PUBLIC MESSAGE SHOULD BE EQUAL TO THE SECRET OUTPUT OF A HASH FUNCTION.

UNDER THE HOOD

ARITHMETIC CIRCUIT DEFINITION

- WE TRANSFORM OUR APPLICATION IN AN ARITHMETIC CIRCUIT WITH PUBLIC INPUTS (X), PRIVATE INPUTS (WITNESSES), GATES (ADD, MUL, CUSTOM), WIRES AND PUBLIC OUTPUTS.
- WE DECIDE HOW TO CONSTRAINT THE CIRCUIT E.G. A PUBLIC MESSAGE SHOULD BE EQUAL TO THE SECRET OUTPUT OF A HASH FUNCTION.

ENCODING POLYNOMIALS

- THE CIRCUIT IS TRANSFORMED IN A SET OF CONSTRAINTS AND ENCODED IN INTERPOLATION POLYNOMIALS ACCORDING TO THE PARAMETERS OF THE CIRCUIT.

UNDER THE HOOD

ARITHMETIC CIRCUIT DEFINITION

- WE TRANSFORM OUR APPLICATION IN AN ARITHMETIC CIRCUIT WITH PUBLIC INPUTS (X), PRIVATE INPUTS (WITNESSES), GATES (ADD, MUL, CUSTOM), WIRES AND PUBLIC OUTPUTS.
- WE DECIDE HOW TO CONSTRAINT THE CIRCUIT E.G. A PUBLIC MESSAGE SHOULD BE EQUAL TO THE SECRET OUTPUT OF A HASH FUNCTION.

ENCODING POLYNOMIALS

- THE CIRCUIT IS TRANSFORMED IN A SET OF CONSTRAINTS AND ENCODED INTO INTERPOLATION POLYNOMIALS ACCORDING TO THE PARAMETERS OF THE CIRCUIT.

PROVING SYSTEM

- WE PROVE DIFFERENT PROPERTIES OF THE POLYNOMIALS USING A PCS AND AN (NON-INTERACTIVE) PROOF SYSTEM

4. ZEKROM

ZEKROM

CONTEXT

- ORIGINALLY A MASTER'S THESIS/RESEARCH PROJECT
- NOW RELEASED AS AN OPEN-SOURCE LIBRARY
- WITH THE GOAL TO BE FRIENDLY TOWARDS IMPLEMENTERS

CAPABILITIES

- SCHEMES: RESCUE-PRIME, GRIFFIN, NEPTUNE, REINFORCED CONCRETE & CIMINION
- PROVING SYSTEMS: GROTH16, MARLIN (ARKWORKS-RS), HALO2 (HALO2)
- TOOLS: BENCHMARK 101, CONSTANTS VERIFICATION
- EASE-OF-USE: HIGH-LEVEL API, CHOICES GUIDE AND RESEARCH ARTICLE

TRY IT YOURSELF

- HTTPS://GITHUB.COM/KUDELSKISECURITY/ZEKROM-ARKWORKS
- HTTPS://GITHUB.COM/KUDELSKISECURITY/ZEKROM-HALO2
- HTTPS://GITHUB.COM/KUDELSKISECURITY/ZEKROM-EXAMPLE

QUESTIONS

?