Compressed Σ-protocols

Lisa Kohl

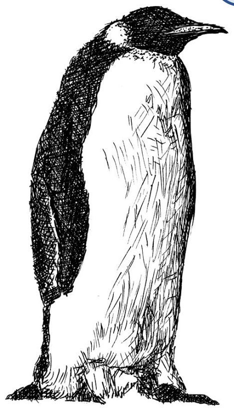
Cryptology Group, CWI Amsterdam

Foundations and Applications of Zero-Knowledge Proofs, Edinburgh

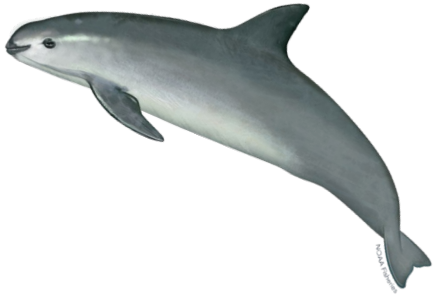
Succinct Arguments of Knowledge

I know witness w for statement X

E.g., I know $(w_1, \ldots, w_n)$ such that $X = f(w_1, \ldots, w_n)$ for some function f



Prover



Verifier (Vaquita)

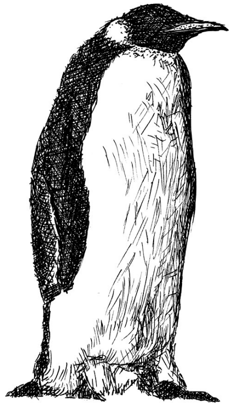
- How can the prover convince the verifier with communication $\ll \mathrm{n}$?

Disclaimer: *No zero-knowledge for now*

Today: Succinct Arguments of Knowledge via Compressed Σ-protocols

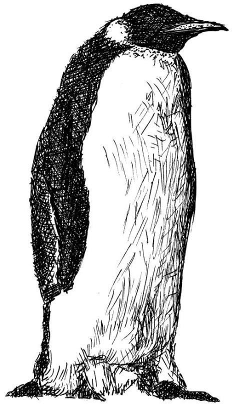
Bulletproofs vs. Compressed Σ-protocols

Bulletproofs [BCC+'16, BBB+'18]:



Inner Product Relations:
I know $\vec{u}, \vec{v} \in \mathbb{Z}_p^n$ such that
$X = Com(\vec{u}), Y = Com(\vec{v})$,
and $c = \langle \vec{u}, \vec{v} \rangle$
(where c is a scalar $c \in \mathbb{Z}_p$)

Compressed Σ-Protocols [AC'20]:



Linear Relations:
I know $\vec{w} \in \mathbb{Z}_p^n$ such that
$X = Com(\vec{w})$ *and* $y = L(\vec{w})$
(where L is a linear form
$L: \mathbb{Z}_p^n \to \mathbb{Z}_p$*)*

- Lifts Bulletproof compression mechanism to Σ-protocol theory (Part 1)
- Uses linearization techniques from arithmetic secret sharing to prove general arithmetic circuits (Part 2, if time)

Intuition/ high-level recipe

- **Knowledge Soundness:** If  convinces , it must “know” a witness
- **Succinctness:** |Communication| $\ll$ |Witness w|

- **Blue-print:** (**Here:** $\Sigma - \text{protocol}$)
  1. The prover  sends a commitment (this has to be succinct!)
  2. The verifier  challenges the prover
  3. The prover  replies to the challenge (this also has to be succinct!)

  repeat

- **Main ingredient:**
  - **Here:** Succinct homomorphic commitments

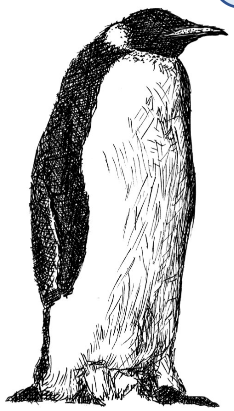
Recall: Σ-protocols

Recall: Σ-protocols

Intuition: If the prover can successfully answer on two different challenges it must *know* the witness

Knowledge error: $^1/_{|\mathcal{C}|}$

I know witness w for statement X

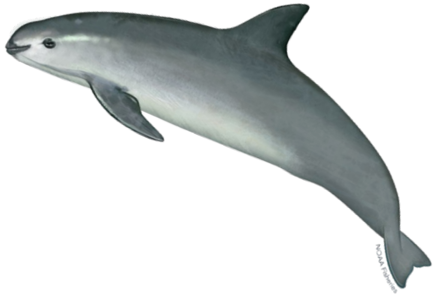


Prover

$$a \quad \longrightarrow$$

$$c \quad \longleftarrow \qquad c \xleftarrow{\$} \mathcal{C}$$

$$z \quad \longrightarrow \qquad b \in \{0,1\}$$



Verifier (Vaquita)

- Σ-protocols satisfy:
  - **Perfect completeness:** Every honest transcript is accepting (i.e., V outputs 1)
  - **(2-)Special soundness:** Giving two accepting transcripts $(a, c, z), (a, c', z')$ with $c \neq c'$ one can efficiently compute a witness $\widetilde{w}$ for X
  - **[Honest verifier zero knowledge:** Honest transcripts can be efficiently simulated (without knowing the witness w)]

Homomorphic commitments

Homomorphic commitments

Commitment scheme: Commit to w via $\boxed{w}$ such that:

- **Hiding:** $\boxed{w}$ hides w
- **Binding:** $\boxed{w}$ can only be opened to w

Additional required properties:

- **Homomorphic:** $\boxed{w} + \boxed{v} = \boxed{w+v}$
- **Succinct:** $\left|\boxed{w}\right| \ll |w|$

Example

Commitment scheme (almost): G group with generator g, $\boxed{w} := g^w$

- **Hiding:** not really (can be made hiding by multiplying h^r → Pedersen Commitments)

- **Binding:** g^w uniquely determines w

Additional required properties:

- **Homomorphic:** $g^w \cdot g^v = g^{w+v}$

- **Succinct:**

Example

Commitment scheme (almost): $g_1, \ldots g_n$ generators of G, $\boxed{\vec{w}} := g_1^{w_1} \cdot \cdots \cdot g^{w_n}$

- **Hiding:** somewhat (can be made fully hiding by multiplying h^r)
- **Binding:** Yes, if DLOG is hard

Additional required properties:

- **Homomorphic:** $g^{\vec{w}} \cdot g^{\vec{v}} = g_1^{w_1} \cdot \cdots \cdot g_n^{w_n} \cdot g_1^{v_1} \cdot \cdots \cdot g_n^{v_n} = g^{\vec{w}+\vec{v}}$
- **Succinct:** $|\boxed{\vec{w}}|$ is independent of n!

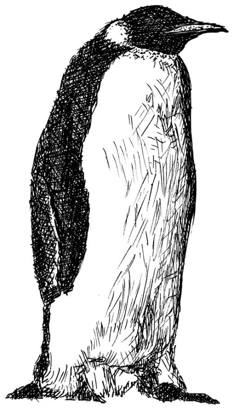
(Non-Zero-Knowledge)
Σ-Protocol for Commitment Opening

[AttemaCramer'20]

Goal: Σ-Protocol for Commitment Opening

In this talk:
G group with order p,
$g_1, \dots, g_n$ known generators

I know $(w_1, \dots, w_n) \in \mathbb{Z}_p^n$ such that $X = g_1^{w_1} \cdot \dots \cdot g_n^{w_n}$



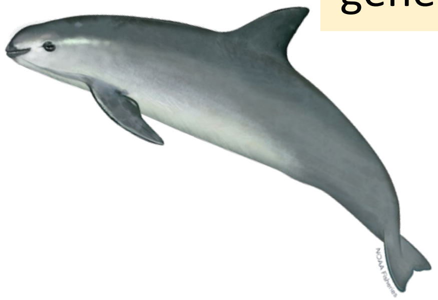
Prover

a →

c ←

z →

$c \leftarrow \mathcal{C}$



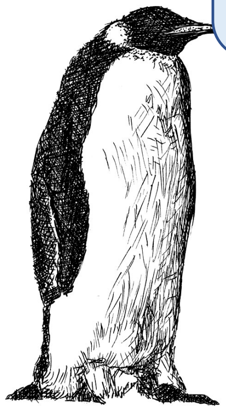
Verifier (Vaquita)

- **Completeness:** Every honest transcript is accepting (i.e., V outputs 1)
- **k-Special soundness:** Giving k accepting transcripts (a_i, c_i, z_i) with $c_i \neq c_j$ one can efficiently compute a witness $\tilde{w}$ for X
- **Succinctness:** $|\text{Communication}| \ll n$

Σ-Protocol for Commitment Opening

I know value $\vec{w}$ such that $\boxed{\vec{w}}$ can be opened to $\vec{w}$

$\boxed{\vec{w}}$

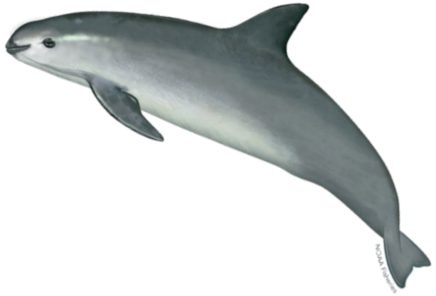


Prover

$$\xrightarrow{\quad \emptyset \quad}$$

$$\xleftarrow{\quad \emptyset \quad}$$

$$\xrightarrow{\quad \vec{w} \quad}$$



Verifier (Vaquita)

- Complete: ✓
- Special Sound: ✓
- Succinct: ✗

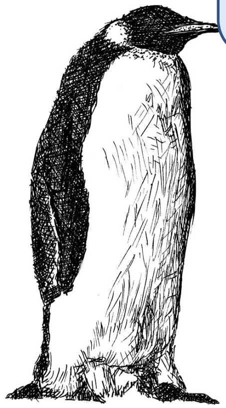
Idea: Fold $\vec{w}$
[BCC+'16, BBB+'18]

Σ-Protocol for Commitment Opening

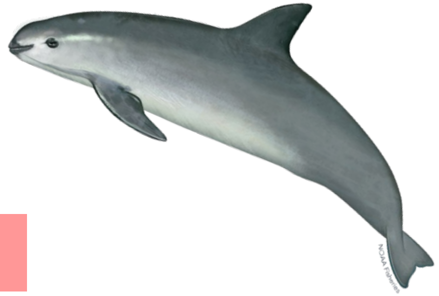
1. Attempt

I know value $\vec{w} = (\overrightarrow{w_1}, \overrightarrow{w_2})$
such that $\boxed{\vec{w}}$
can be opened to $\vec{w}$

$\boxed{\vec{w}}$



Prover



Verifier (Vaquita)

$$\text{Prover} \xrightarrow{\emptyset} \text{Verifier}$$

$$\text{Prover} \xleftarrow{\emptyset} \text{Verifier}$$

$$\text{Prover} \xrightarrow{\vec{z} := \overrightarrow{w_1} + \overrightarrow{w_2}} \text{Verifier}$$

Can't verify

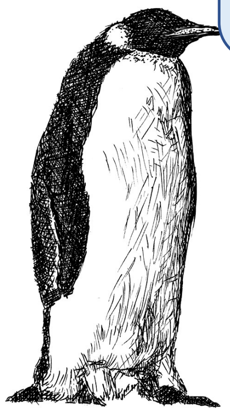
- Well-defined: ❌

Σ-Protocol for Commitment Opening

2. Attempt

I know value $\vec{w} = (\vec{w_1}, \vec{w_2})$
such that $\boxed{\vec{w}}$
can be opened to $\vec{w}$

$\vec{w_{rev}} := \begin{pmatrix} \vec{w_2} \\ \vec{w_1} \end{pmatrix}$ $\boxed{\vec{w}}$

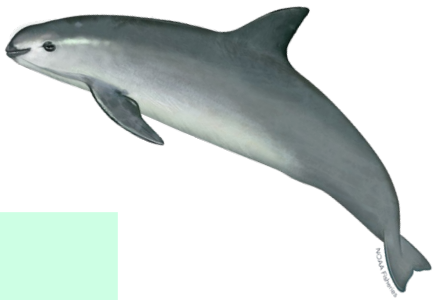


Prover

$\boxed{\vec{w_{rev}}}$ $\longrightarrow$

$\emptyset$ $\longleftarrow$

$\vec{z} := \vec{w_1} + \vec{w_2}$ $\longrightarrow$



Verifier (Vaquita)

Can check:

$\begin{pmatrix} \vec{z} \\ \vec{z} \end{pmatrix}$ valid opening for

$\boxed{\vec{w_{rev}}} + \boxed{\vec{w}}$

Problem: this linear combination is fixed

- Complete: ✓
- Special Sound: ✗

More high-level: Need **random challenge** (if the reply of the prover is fixed we cannot hope to extract the witness, as the information carried in $\vec{z}$ is smaller than $\vec{w}$

Breaking soundness: Prover can cheat using homomorphic property by sending

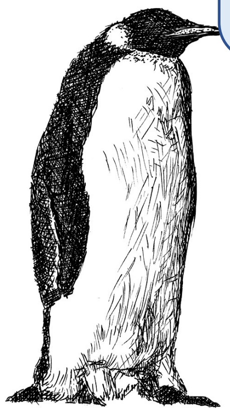
$- \boxed{\vec{w}} + \boxed{\begin{pmatrix} \vec{z} \\ \vec{z} \end{pmatrix}}$

Σ-Protocol for Commitment Opening

3. Attempt

I know value $\vec{w} = (\vec{w_1}, \vec{w_2})$
such that $\boxed{\vec{w}}$
can be opened to $\vec{w}$

$\vec{w_{rev}} := \begin{pmatrix} \vec{w_2} \\ \vec{w_1} \end{pmatrix}$ $\boxed{\vec{w}}$

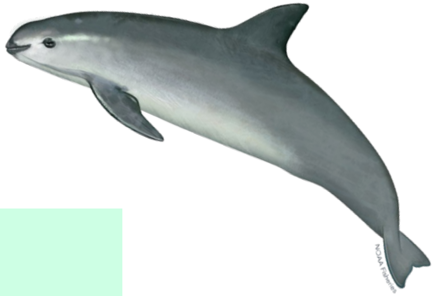


Prover

$\boxed{\vec{w_{rev}}}$

d

$??$



Verifier (Vaquita)

Can check:

$??$ valid opening for

$\boxed{\vec{w_{rev}}} + d \cdot \boxed{\vec{w}}$

- Succinct: ❌

Issue:
Would need to send
$\vec{w_2} + d \cdot \vec{w_1}$
and
$\vec{w_1} + d \cdot \vec{w_2}$
→ back to size n!!

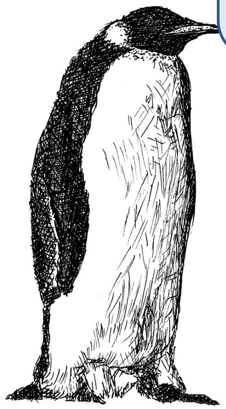
Observation: [BCC+'16, BBB+'18]
$d \cdot (\vec{w_1} + d \cdot \vec{w_2}) = \mathrm{d}^2 \cdot \vec{w_2} + \mathrm{d} \cdot \vec{w_1}$

Σ-Protocol for Commitment Opening

4. Attempt

I know value $\vec{w} = (\vec{w_1}, \vec{w_2})$
such that $\boxed{\vec{w}}$
can be opened to $\vec{w}$

$\vec{w_{rev}} := \begin{pmatrix} d^2 \cdot \vec{w_2} \\ \vec{w_1} \end{pmatrix}$ $\boxed{\vec{w}}$

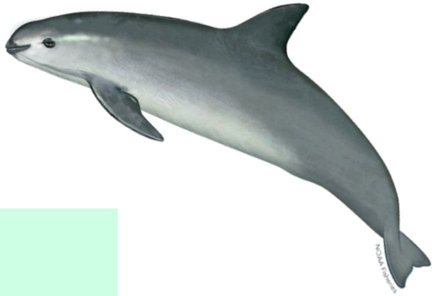


Prover

$\boxed{\vec{w_{rev}}}$ →

d ←

$\vec{z} := \vec{w_1} + d \cdot \vec{w_2}$ →



Verifier (Vaquita)

Can check:

$\begin{pmatrix} d \cdot \vec{z} \\ \vec{z} \end{pmatrix}$ valid opening for

$\boxed{\vec{w_{rev}}} + d \cdot \boxed{\vec{w}}$

Completeness:

$$\begin{pmatrix} d \cdot (\vec{w_1} + \mathrm{d} \cdot \vec{w_2}) \\ \vec{w_1} + d \cdot \vec{w_2} \end{pmatrix} = \begin{pmatrix} d^2 \cdot \vec{w_2} + d \cdot \vec{w_1} \\ \vec{w_1} + d \cdot \vec{w_2} \end{pmatrix}$$

valid opening!

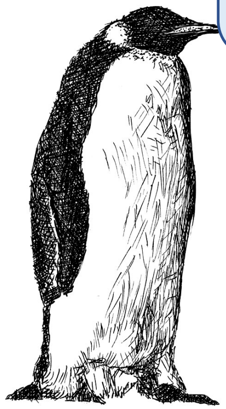
- Well-defined: ❌

Problem:

- **either:** prover **doesn't know d** in first round and **can't generate first message**
- **or:** prover **does know d** in first round and **can cheat** (as before)

Σ-Protocol for Commitment Opening

5. Attempt



I know value $\vec{w} = (\vec{w_1}, \vec{w_2})$ such that $\boxed{\vec{w}}$ can be opened to $\vec{w}$

Prover

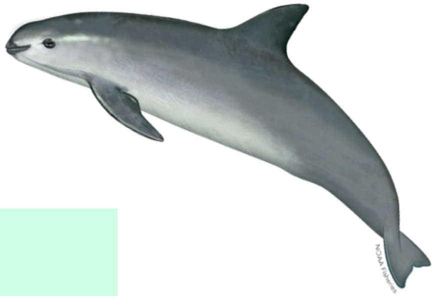
$$\vec{w_L} := \begin{pmatrix} 0 \\ \vec{w_1} \end{pmatrix} \qquad \boxed{\vec{w}}$$

$$\vec{w_r} := \begin{pmatrix} \vec{w_2} \\ 0 \end{pmatrix}$$

$\boxed{\vec{w_L}}$ $\boxed{\vec{w_R}}$ →

← d

$\vec{z} := \vec{w_1} + d \cdot \vec{w_2}$ →



Verifier (Vaquita)

Can check:

$\begin{pmatrix} d \cdot \vec{z} \\ \vec{z} \end{pmatrix}$ valid opening for

$\boxed{\vec{w_L}} + d \cdot \boxed{\vec{w}} + d^2 \cdot \boxed{\vec{w_R}}$

Completeness:

$$\begin{pmatrix} d \cdot \vec{w_1} + d^2 \cdot \vec{w_2} \\ \vec{w_1} + d \cdot \vec{w_2} \end{pmatrix}$$

valid opening!

- Complete:
- (3-)Special Sound: (see next slide)
- Succinct:

3-Special Soundness

3-Special Soundness

Assume to be given 3 accepting transcripts

- ($\cancel{\vec{w}_L}$ $\cancel{\vec{w}_R}$, d_1 , $\vec{z}_1$)
- ($\cancel{\vec{w}_L}$ $\cancel{\vec{w}_R}$, d_2 , $\vec{z}_2$)
- ($\cancel{\vec{w}_L}$ $\cancel{\vec{w}_R}$, d_3 , $\vec{z}_3$) s.t. $\cancel{\vec{w}_L} + d_i \cdot \cancel{\vec{w}} + d_i^2 \cdot \cancel{\vec{w}_R} = \cancel{\begin{pmatrix} d_i \cdot \vec{z}_i \\ \vec{z}_i \end{pmatrix}}$

Vandermonde Matrix V

- I.e., we **know an opening** for $\underbrace{\begin{pmatrix} 1 & d_1 & d_1^2 \\ 1 & d_2 & d_2^2 \\ 1 & d_3 & d_3^2 \end{pmatrix} \cdot \begin{pmatrix} \cancel{\vec{w}_L} \\ \cancel{\vec{w}} \\ \cancel{\vec{w}_R} \end{pmatrix}}$ and thus also for

$$\cancel{\vec{w}} = (0\ 1\ 0) \cdot V^{-1} \cdot$$

From communication $\mathcal{O}(n/2)$ to $\mathcal{O}(\log n)$

Another View

- **Recall:** P proves knowledge of $\vec{z}$ such that

$$\begin{pmatrix} d\cdot\vec{z} \\ \vec{z} \end{pmatrix} \text{ valid opening for } \boxed{\overrightarrow{w_L}} + d\cdot\boxed{\vec{w}} + d^2\cdot\boxed{\overrightarrow{w_R}}$$

- **Alternatively:** $\vec{z}$ is valid opening for $\boxed{\vec{z}}$ under new generators:

$$g_1^{d\cdot z_1}\cdot\ldots\cdot g_{n/2}^{d\cdot z_{n/2}}\cdot g_{n/2+1}^{z_1}\cdot\ldots\cdot g_n^{d\cdot z_{n/2}} = (g_1^d\cdot g_{n/2+1})^{z_1}\cdot\ldots\cdot(g_{n/2}^d\cdot g_n)^{z_{n/2}}$$

Recursive Folding

$$\boxed{\overrightarrow{w_L}} + d \cdot \boxed{\vec{w}} + d^2 \cdot \boxed{\overrightarrow{w_R}}$$

- **Now:**  proves knowledge of $\vec{z} \in \mathbb{Z}_p^{n/2}$ s.t. $\boxed{\vec{z}}$ opens to $\vec{z}$
- Instead of sending $\vec{z}$ we can repeat the folding procedure!
- **Important:** Use fresh challenge each time → more communication rounds
- **After $\log n$ repetitions:** Only have to send $z \in \mathbb{Z}_p$
- **Overall communication:** $2 \cdot \log n \cdot |\boxed{}| + \log p$

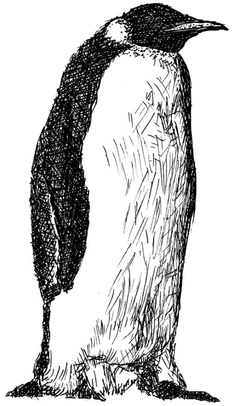
Size of a group element

What did we get?

Result: Σ-Protocol for Commitment Opening

In this talk:
G group with order p,
$g_1, \dots, g_n$ known generators

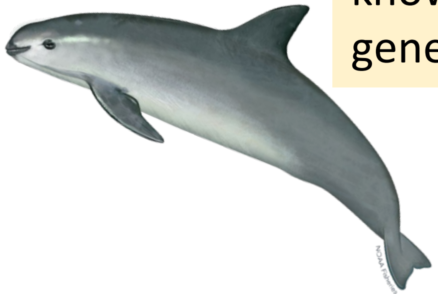
I know $(w_1, \dots, w_n) \in \mathbb{Z}_p^n$ such that $X = g_1^{w_1} \cdot \dots \cdot g_n^{w_n}$



a →

← c $\quad c \leftarrow \mathcal{C}$

z →



Prover

Verifier (Vaquita)

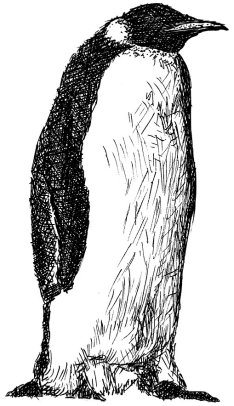
- **Completeness:** Every honest transcript is accepting (i.e., V outputs 1)
- **$(3, 3, \dots, 3)$-Special soundness:** Giving a "tree of accepting transcripts" one can efficiently compute a witness $\tilde{w}$ for X
- **Succinctness:** $|\text{Communication}| \approx \log n \cdot |G|$

[AttemaCramer**Kohl**'21] Tight Analysis of Knowledge Extractor
→ Knowledge Error $\leq 2 \log n / p$

Succinctness & Zero Knowledge?

Adding Zero-Knowledge

- Simply start with a standard (non-succinct) Σ-protocol → HVZK



$w, X = g^w$ $\boxed{w}$

$r \leftarrow \mathbb{Z}_p$

g^r $\boxed{r}$ $\longrightarrow$

c $\longleftarrow$

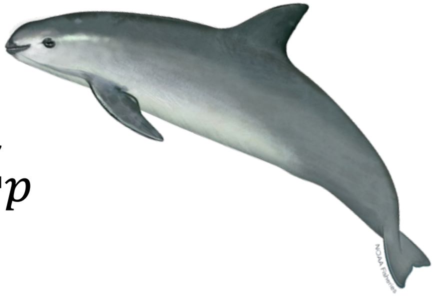
$z = r + c \cdot w$ $\longrightarrow$

Prover

$X = g^w$

$\boxed{w}$

$c \leftarrow \mathbb{Z}_p$



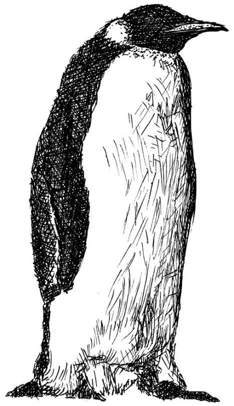
Verifier (Vaquita)

$$g^z \stackrel{?}{=} g^r \cdot (g^w)^c$$

$$\boxed{z} \stackrel{?}{=} \boxed{r} + c \cdot \boxed{w}$$

Adding Zero-Knowledge

- Can generalize this to homomorphic commitments!



Prover

$\vec{w}$ $\quad \boxed{\vec{w}}$

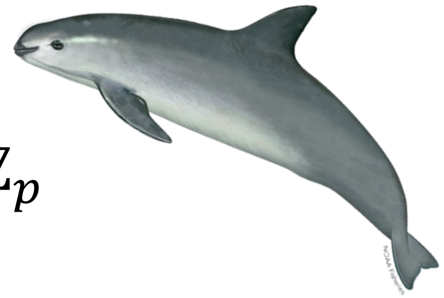
$r \leftarrow \mathbb{Z}_p$

$\boxed{\vec{r}}$ $\longrightarrow$

$\longleftarrow$ c

$\vec{z} = \vec{r} + c \cdot \vec{w}$ $\longrightarrow$

$\boxed{\vec{w}}$



$c \leftarrow \mathbb{Z}_p$

Verifier (Vaquita)

$$\boxed{\vec{z}} \stackrel{?}{=} \boxed{\vec{r}} + c \cdot \boxed{\vec{w}}$$

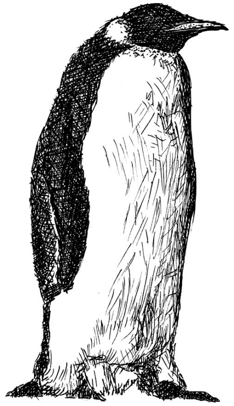
- Instead of sending the third round message:
  -  proves knowledge of opening of $\boxed{\vec{r}} + c \cdot \boxed{\vec{w}}$

Compressed Σ-protocols for Proving Linear Forms

[AttemaCramer'20]

Goal: Σ-Protocol for Linear Relations

Linear Relations:
I know $\vec{w} \in \mathbb{Z}_p^n$ such that
$X = Com(\vec{w})$ *and* $y = L(\vec{w})$
(where L *is a linear form*
$L: \mathbb{Z}_p^n \to \mathbb{Z}_p$*)*



Prover

a

$\longrightarrow$

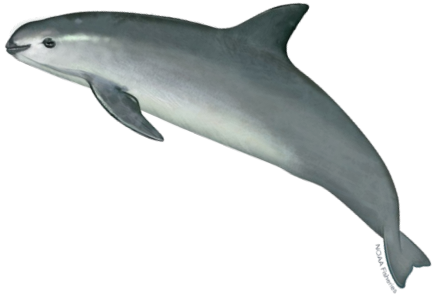
c

$\longleftarrow$

z

$\longrightarrow$

$c \leftarrow \mathcal{C}$

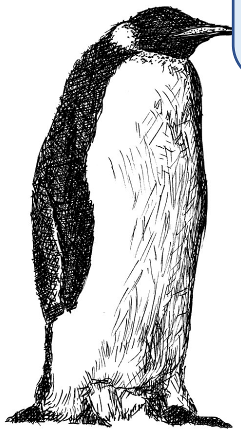


Verifier (Vaquita)

Σ-Protocol for Linear Relations

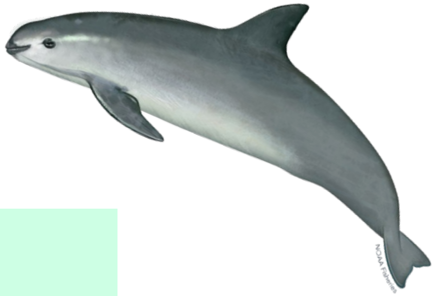
L is a linear form
$L: \mathbb{Z}_p^n \to \mathbb{Z}_p$

I know value $\vec{w} = (\overrightarrow{w_1}, \overrightarrow{w_2})$ such that $\boxed{\vec{w}}$ can be opened to $\vec{w}$ and $y = L(\vec{w})$



Prover

$\boxed{\vec{w}}$ y



Verifier (Vaquita)

$$\overrightarrow{w_L} := \begin{pmatrix} 0 \\ \overrightarrow{w_1} \end{pmatrix} \qquad \overrightarrow{w_r} := \begin{pmatrix} \overrightarrow{w_2} \\ 0 \end{pmatrix}$$

$\boxed{\overrightarrow{w_L}}$ $\boxed{\overrightarrow{w_R}}$ $L(\overrightarrow{w_L}), L(\overrightarrow{w_R})$ →

← d

$\vec{z} := \overrightarrow{w_1} + d \cdot \overrightarrow{w_2}$ →

Can check:

$\begin{pmatrix} d \cdot \vec{z} \\ \vec{z} \end{pmatrix}$ valid opening for

$\boxed{\overrightarrow{w_L}} + d \cdot \boxed{\vec{w}}$

$+ d^2 \cdot \boxed{\overrightarrow{w_R}}$

And:

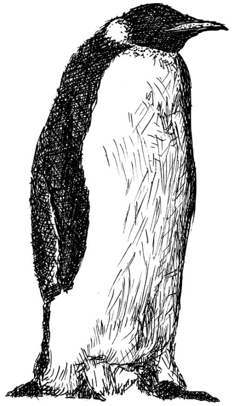
$$L\left(\begin{pmatrix} d \cdot \vec{z} \\ \vec{z} \end{pmatrix}\right) = L(\overrightarrow{w_L}) + d \cdot y + d^2 \cdot L(\overrightarrow{w_R})$$

Σ-protocols for Circuit ZK

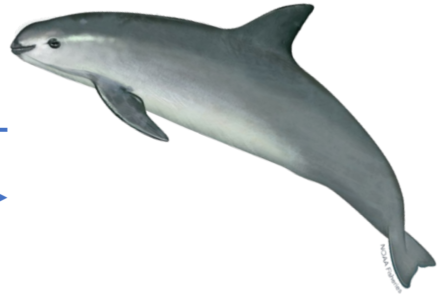
The missing part: How to prove correctness of multiplication gates

Goal: Σ-Protocol for Circuit ZK

I know $(w_1, \dots, w_n)$ such that $f(w_1, \dots, w_n) = 0$ for some function f



Prover



Verifier (Vaquita)

- **Completeness:** Every honest transcript is accepting (i.e., V outputs 1)
- **Knowledge soundness:** A successful prover must "know" the witness
- **Succinctness:** $|\text{Communication}| \ll n$

Here: Consider f to be an **arithmetic circuit**, i.e., only to consist of additions and multiplications over a (large) finite field $\mathbb{F}$, **known to all parties**

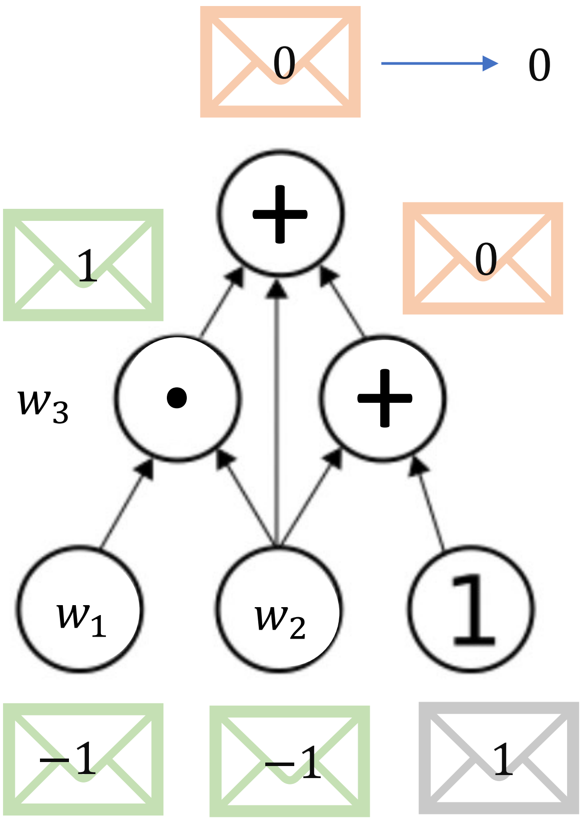
Note: Not succinct!

A blue print for zero knowledge proofs

[CramerDamgård'97]

Goal: Prove $f(w) = 0$ without revealing w

1. Write $f: \mathbb{F}^m \to \mathbb{F}$ as **arithmetic circuit** with multiplication and addition gates
2. **Extend witness** w **to all intermediary results of multiplication gates**
3. Commit to the extended witness using a **homomorphic commitment scheme**
4. Evaluate addition gates **homomorphically and open final result**
5. **Prove correctness** of multiplication gates



$f(w_1, w_2) = w_1 \cdot w_2 + w_2 + w_2 + 1$

Witness: $w_1 = -1, w_2 = -1, w_3 := 1$

Compressed Σ-protocols for Proving Many Multiplications

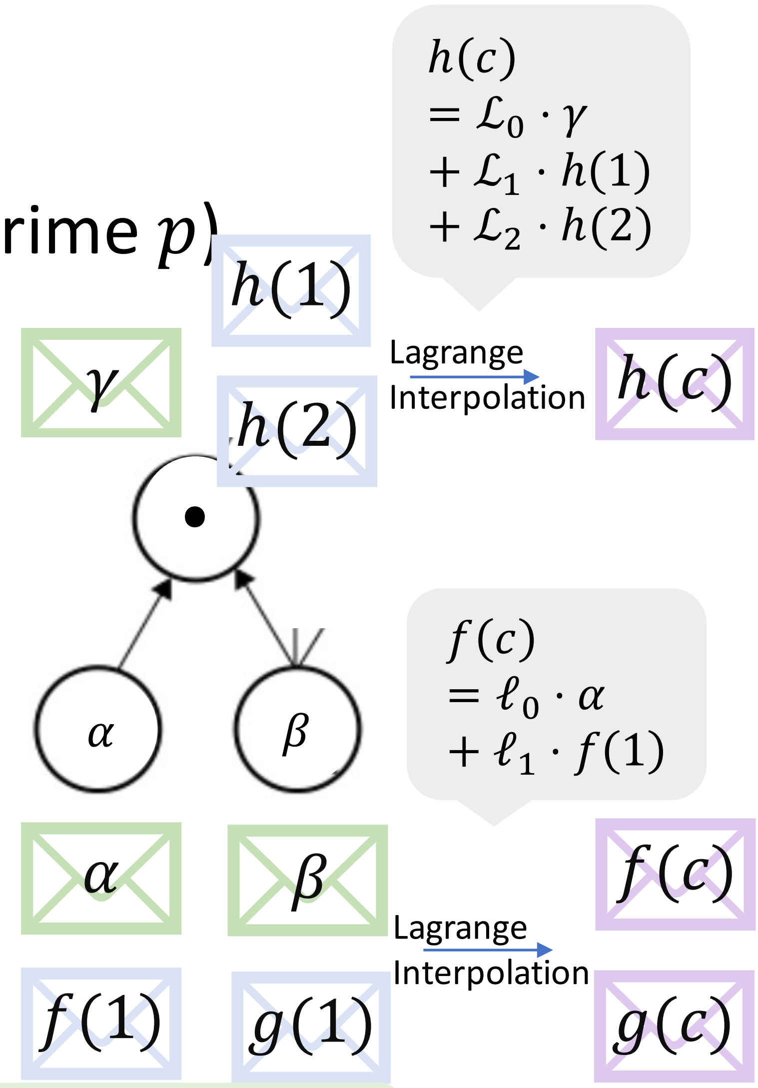
[CramerDamgård'97, CramerDamgårdMaurer'00, CramerDamgårdPastro'12 AttemaCramer'20]

Linearizing Multiplication Gates

[CramerDamgård'97]

Shamir secret sharing: (assume $\mathbb{F} = \mathbb{Z}_p$ for large prime p)

1. chooses random $f(X)$, $g(X)$ of degree 1 such that
   - $f(0) = \alpha, g(0) = \beta$
2. sets $h(X) \coloneqq f(X) \cdot g(X)$
3. commits to:
   - $f(1), g(1)$ (note: together with α, β this fully determines f, g)
   - $h(1), h(2)$ (note: together with γ this fully determines h)
4. sends a challenge $c \leftarrow \mathbb{Z}_p \setminus \{0\}$
5. sends the opening $f(c), g(c), h(c)$
6. checks if openings are correct & $h(c) = f(c) \cdot g(c)$



Zero knowledge: hiding of commitments + f, g random $\rightarrow f(c), g(c)$ random

Soundness: binding of commitments + $h - f \cdot g \neq 0$ has at most 2 zero positions

Proving Many Multiplication Gates (1/2)

[CramerDamgård'97, CramerDamgårdMaurer'00, CramerDamgårdPastro'12, AttemaCramer'20]

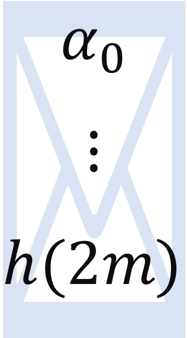
Now: m multiplication gates $\alpha_i, \beta_i, \gamma_i = \alpha_i \cdot \beta_i$ $(0 \leq i < m)$

1. **Packed secret sharing:**  chooses random f, g of degree m s.t. $f(i) = \alpha_i, g(i) = \beta_i$
2.  sets $h(X) \coloneqq g(X) \cdot f(X)$
3.  additionally commits to $f(m), g(m), h(m), \ldots, h(2m)$
4.  sends a challenge $c \leftarrow \mathbb{Z}_p \setminus \{0\}$
5.  sends the opening $f(c), g(c), h(c)$
6.  checks if openings are correct & $h(c) = f(c) \cdot g(c)$

Issue: Communication scales with the size of the circuit

Observation:

Can pack all values in **succinct vector commitment** and use **Σ-protocols for linear forms** to prove correct openings $f(c), g(c), h(c)$



Proving Many Multiplication Gates (2/2)

[CramerDamgård’97, CramerDamgårdMaurer’00, CramerDamgårdPastro’12, AttemaCramer’20]

- More precisely, we have to prove three linear forms L_1, L_2, L_3:

$$(\ell_0\ \ell_1 \ldots \ell_m\ 0\ \ldots\ \ldots 0) \begin{pmatrix}\alpha_0\\ \vdots \\ f(m) \\ \vdots\end{pmatrix} = f(c)$$

$$(0 \ldots 0\ \ell_0\ \ell_1 \ldots \ell_m\ 0 \ldots 0)\begin{pmatrix}\vdots\\ \beta_0\\ \vdots\\ h(m)\\ \vdots\end{pmatrix} = g(c)$$

$$(0 \ldots \ldots 0\ \mathcal{L}_0\ \mathcal{L}_1 \ldots \mathcal{L}_{2m})\begin{pmatrix}\vdots\\ \gamma_0\\ \vdots\\ h(2m)\end{pmatrix} = h(c)$$

Only need Σ-protocols for linear forms

Σ-protocols for Circuit ZK

[AttemaCramer'20]

From Multiplications to Circuit ZK

[AttemaCramer'20]

Observation:

1. Wires α_i, β_i are determined by affine forms $u_i(w_1, \dots, w_n, \gamma_1, \dots, \gamma_m), v_i(w_1, \dots, w_n, \gamma_1, \dots, \gamma_m)$
2. Same for the output value $\mathrm{f}(w_1, \dots, w_n)$

Strategy:

1. Instead of committing to α_i, β_i use the affine forms to define f, g
2. Finally, show $\mathrm{f}(w_1, \dots, w_n) = 0$ as required

Fiat-Shamir and Parallel Repetition

Some Notes on Multi-Round Σ-Protocols

- **Parallel repetition of Σ-protocols:**
  - **2-special soundness:** t-fold parallel repetition also satisfies 2-special soundness $\rightarrow$ knowledge error decreases exponentially to $1/|\mathcal{C}|^t$
  - **k-special soundness:** t-fold parallel repetition only satisfies $((k-1)^t+1)$-special soundness $\rightarrow$ extractor becomes inefficient for large t
  - $(\boldsymbol{k_1}, \dots, \boldsymbol{k_n})$**-special soundness:** not clear if it satisfies meaningful notion of special soundness
- [AttemaFehr'22]: Parallel repetition reduces the knowledge error to κ^t
- [AttemaFehrKlooss'22]:
  - Fiat Shamir of $(\boldsymbol{k_1}, \dots, \boldsymbol{k_n})$**-special sound protocols** has linear soundness loss Q
  - Fiat Shamir of **t-fold** $(\boldsymbol{k_1}, \dots, \boldsymbol{k_n})$**-special sound protocols** has exponential soundness loss Q^μ if $t > \mu$

Thank you!