Ozirus's Dilemma: Predict, Decide, Prove

Ozirus Morency

August 2025

$$\boxed{\mathsf{OD} = (\mathrm{Sig} \circ \mathrm{Merkle}) \circ (\mathrm{CRT} \circ \mathrm{Round}_{\Delta}) \circ (\mathrm{Alloc} \circ \mathrm{Hazard}_{\tau} \circ \mathrm{Intensity})}$$

Abstract

This document introduces a three-part method for high-stakes decisions under uncertainty. The method is called Ozirus's Dilemma and it has a simple identity: a Structured Decision Arena *predicts* hazard and allocates resources, a Policy Computation Plane *packages* and compresses the decision, and a Receipt Generation Module *proves* the outcome. The framework uses risk models with regime switching and Hawkes processes to estimate near-term hazard. It solves a convex optimisation problem to divide minutes across actions. It quantises the result and encodes it using number theoretic residues. It then commits the data with a Merkle root and signs it. Worked examples show how to apply the method in a security operations centre and in a hospital. A final section explains how to protect and license the work. All sentences are short and clear.

Contents

1 Introduction

Decisions with heavy impact need structure. Ozirus's Dilemma gives you that structure. It is a set of rules for predicting risk, deciding on actions, and proving what you did. It turns data into a plan and a proof. The plan shows how to act when things happen fast. The proof shows others that you followed the plan. This document explains the identity and the steps. It also gives two examples and explains how to protect and license the method.

2 Identity and Mathematics

This section defines the identity and the math behind the three parts: the Structured Decision Arena (SDA), the Policy Computation Plane (PCP), and the Receipt Generation Module (RGM).

2.1 Notation

- t_k: minutes since the k-th recent event.
- μ: baseline intensity, in $\texttt{min}^{-1}$.
- ω: excitation weight, dimensionless.
- δ: decay rate, in $\texttt{min}^{-1}$.
- $\boldsymbol{\zeta}^\top \boldsymbol{\varphi}(c_t)$: context bump, dimensionless.
- λ: current intensity, in $\texttt{min}^{-1}$.
- τ: horizon in minutes.
- h: probability of at least one event in the horizon, dimensionless.
- L_i: impact weight for action i.
- B: total minutes available.
- u_i: minutes assigned to action i.
- $u_i^{\max}$: maximum minutes for action i.
- Δ: quantisation step.
- $\tilde{\boldsymbol{x}}$: quantised integer vector.
- p_i: pairwise coprime primes.
- $\boldsymbol{a}_i$: short integer row for residue projection.
- r_i: residue modulo p_i.
- C: Merkle root.
- σ: digital signature.

2.2 Core Identity

$$\mathcal{D} \xrightarrow{\text{Intensity}_{\mu,\omega,\delta,\zeta,\varphi}} \lambda \xrightarrow{\text{Hazard}_\tau} h \xrightarrow{\text{Alloc}_{B,\{u_i^{\max}\}}} (u_i) \xrightarrow{\text{Round}_\Delta} \tilde{\boldsymbol{x}} \xrightarrow{\text{CRT}_{A,\{p_i\}}} (r_i) \xrightarrow{\text{Merkle, Sig}} \mathcal{E}.$$

$$\text{OD}(\mathcal{D}) = \big(\text{Sig} \circ \text{Merkle}\big) \circ \big(\text{CRT}_{A,\{p_i\}} \circ \text{Round}_\Delta\big) \circ \big(\text{Alloc}_{B,\{u_i^{\max}\}} \circ \text{Hazard}_\tau \circ \text{Intensity}_{\mu,\omega,\delta,\zeta,\varphi}\big)\big(\mathcal{D}\big).$$

SDA predicts risk and chooses action minutes. PCP compresses the result and projects it into residues. RGM produces a signed receipt. These parts work together: predict, decide, prove.

2.3 Structured Decision Arena

SDA uses a Hawkes process to model clustered events. The intensity with context is

$$\lambda = \mu + \omega \sum_k e^{-\delta t_k} + \boldsymbol{\zeta}^\top \boldsymbol{\varphi}(c_t).$$

The short-horizon probability of at least one event is

$$h = 1 - e^{-\lambda \tau}.$$

With impacts L_i set scores $a_i = -h\, L_i$. Solve the quadratic program

$$\min_{\boldsymbol{u}} \ \tfrac{1}{2}\, \boldsymbol{u}^\top \boldsymbol{u} + \boldsymbol{a}^\top \boldsymbol{u} \quad \text{s.t.} \quad \sum_i u_i \le B, \quad 0 \le u_i \le u_i^{\max}.$$

KKT gives water–filling:

$$u_i^* \ = \ \text{clip}\big(-a_i - \lambda^*,\, 0,\, u_i^{\max}\big), \qquad \sum_i u_i^* = \min\Big\{ B, \sum_i u_i^{\max} \Big\}.$$

The SDA output is $\boldsymbol{x} = [\, h, u_1^*, u_2^*, \dots]$.

2.4 Policy Computation Plane

PCP turns $\boldsymbol{x}$ into integers and residues.

Quantisation.

$$\tilde{\boldsymbol{x}} = \text{round}(\boldsymbol{x}/\Delta) \in \mathbb{Z}^d, \qquad \|\tilde{\boldsymbol{x}}\|_\infty \le B_{\max},$$

where d is the length of $\tilde{\boldsymbol{x}}$.

Residues. Let $\boldsymbol{a}_i \in \mathbb{Z}^d$ be short integer rows and p_i pairwise coprime primes. Define

$$r_i \equiv \langle \boldsymbol{a}_i, \tilde{\boldsymbol{x}} \rangle \pmod{p_i}, \qquad i = 1, \dots, k,$$

and let $A \in \mathbb{Z}^{k \times d}$ stack the rows $\boldsymbol{a}_i^\top$.

Uniqueness bound (CRT). Let $A \in \mathbb{Z}^{k \times d}$ stack the rows $\boldsymbol{a}_i^\top$ and let

$$\|A\|_\infty := \max_{i=1,\dots,k} \sum_{j=1}^{d} |A_{ij}|.$$

If $\|\tilde{\boldsymbol{x}}\|_\infty \le B_{\max}$, then

$$\max_i |\langle \boldsymbol{a}_i, \tilde{\boldsymbol{x}} \rangle| \le \ \|A\|_\infty \, \|\tilde{\boldsymbol{x}}\|_\infty \le \ \|A\|_\infty B_{\max}.$$

Thus the inner products are uniquely determined by residues modulo $M := \prod_{i=1}^k p_i$ provided

$$M \ > \ 2\, \|A\|_\infty \, B_{\max}.$$

If the inequality fails, append another prime to $\{p_i\}$ and recompute the residues.

2.5 Receipt Generation Module

Build Merkle leaves with a collision-resistant hash H and compute the root:

$$L_i \;=\; H\bigl(r_i \,\|\, \mathrm{meta}_i\bigr), \qquad C \;=\; \mathrm{MerkleRoot}\bigl(\{L_i\}_{i=1}^{k}\bigr).$$

Sign the root (and public parameters) to obtain

$$\sigma \;=\; \mathrm{Sign}_{\mathrm{sk}}(m, C, \{p_i\}, \mathrm{policy}), \qquad \mathcal{E} \;=\; \bigl(\{r_i\}, \{p_i\}, C, \sigma, \mathrm{policy}\bigr).$$

Anyone can check σ and the leaf proofs without seeing $\boldsymbol{x}$.

3 Cybersecurity Example

This example shows how to use the method for a security operations centre.

3.1 Inputs

- Recent failed logins occurred at 25, 12 and 4 minutes.
- Parameters: $\mu = 0.012$, $\omega = 0.18$, $\delta = 0.12$, context bump 0.04.
- Horizon: $\tau = 8$ minutes.
- Actions: forced reset ($L_1 = 3, u_1^{\max} = 8$); soft lock and MFA ($L_2 = 5, u_2^{\max} = 10$); Tier-2 escalation ($L_3 = 7, u_3^{\max} = 12$).
- Budget $B = 24$ minutes.
- Quantisation $\Delta = 0.01$.
- Primes $(101, 103, 107)$ and rows $\boldsymbol{a}_1 = (2, 1, 0, 3)$, $\boldsymbol{a}_2 = (1, 4, 1, 2)$, $\boldsymbol{a}_3 = (3, 0, 5, 1)$.

3.2 Compute Hazard and Allocation

$$\begin{aligned}\sum e^{-\delta t_k} &= e^{-0.12\times 25} + e^{-0.12\times 12} + e^{-0.12\times 4} \approx 0.90550,\\ \lambda &= 0.012 + 0.18 \times 0.90550 + 0.04 = 0.21499,\\ h &= 1 - e^{-0.21499\times 8} = 0.82092.\end{aligned}$$

Set scores $a = -hL = (-2.46276, -4.10460, -5.74643)$. Solve the clip rule with $\sum_i u_i^* = 24$ to get $u^* \approx (6.3582, 8.0000, 9.6418)$.

3.3 Quantisation and Residues

Quantise $\boldsymbol{x} = [0.82092, 6.3582, 8.0000, 9.6418]$ by dividing by 0.01 and rounding to integers: $\tilde{\boldsymbol{x}} = (82, 636, 800, 964)$. Compute residues:

$$\begin{aligned}\langle \boldsymbol{a}_1, \tilde{\boldsymbol{x}} \rangle &= 2 \times 82 + 1 \times 636 + 3 \times 964 = 3692 \equiv 56 \bmod 101,\\ \langle \boldsymbol{a}_2, \tilde{\boldsymbol{x}} \rangle &= 5354 \equiv 101 \bmod 103,\\ \langle \boldsymbol{a}_3, \tilde{\boldsymbol{x}} \rangle &= 5210 \equiv 74 \bmod 107.\end{aligned}$$

Build the Merkle root and sign it. The receipt includes the residues $(56, 101, 74)$, the primes, the root, and the signature.

3.4 Action Plan

You start Tier-2 escalation at once. You apply a soft lock and multi-factor authentication. You queue a forced reset. You send the signed receipt to incident response and audit. Everyone can trust the plan.

4 Education Example

This example shows how to use the method for a university help-desk surge.

4.1 Inputs

- Help-desk tickets at ages $30, 15, 5$ minutes.
- Parameters: $\mu = 0.015$, $\omega = 0.20$, $\delta = 0.09$, context bump 0.05.
- Horizon: $\tau = 12$ minutes.
- Actions: ticket triage ($L_1 = 4, u_1^{\max} = 8$); call TA ($L_2 = 6, u_2^{\max} = 10$); extend lab hours ($L_3 = 7, u_3^{\max} = 12$).
- Budget $B = 15.5$ minutes.
- Quantisation $\Delta = 0.01$.
- Primes $(101, 103, 107)$; rows $\boldsymbol{a}_1 = (2, 1, 0, 3)$, $\boldsymbol{a}_2 = (1, 4, 1, 2)$, $\boldsymbol{a}_3 = (3, 0, 5, 1)$.

4.2 Compute Hazard and Allocation

$$\begin{aligned}\sum e^{-\delta t_k} &= e^{-2.7} + e^{-1.35} + e^{-0.45} \approx 0.96407,\\ \lambda &= 0.015 + 0.20 \cdot 0.96407 + 0.05 = 0.25781,\\ h &= 1 - e^{-0.25781 \cdot 12} = 0.95467.\end{aligned}$$

Set scores $a = (-3.81868, -5.72802, -6.68269)$. Solve the clip rule with $\sum_i u_i^* = 15.5$ to get

$$\boldsymbol{u}^* \approx (3.5756, 5.4849, 6.4396).$$

4.3 Quantisation and Residues

Quantise $\boldsymbol{x} = [0.95467, 3.5756, 5.4849, 6.4396]$: $\tilde{\boldsymbol{x}} = (95, 358, 548, 644)$. Compute residues:

$$\begin{aligned}\langle \boldsymbol{a}_1, \tilde{\boldsymbol{x}} \rangle &= 3698 \equiv 56 \bmod 101,\\ \langle \boldsymbol{a}_2, \tilde{\boldsymbol{x}} \rangle &= 6655 \equiv 67 \bmod 103,\\ \langle \boldsymbol{a}_3, \tilde{\boldsymbol{x}} \rangle &= 3857 \equiv 31 \bmod 107.\end{aligned}$$

Make the Merkle root and sign it. The receipt includes residues $(56, 67, 31)$.

4.4 Action Plan

Run ticket triage. Call one TA. Extend lab hours. Send the signed receipt to IT and Student Services.

5 Construction Example

This example shows how to use the method for a construction site safety surge.

5.1 Inputs

- Incident ages $50, 25, 8$ minutes.
- Parameters: $\mu = 0.010$, $\omega = 0.22$, $\delta = 0.07$, context bump 0.03.
- Horizon: $\tau = 20$ minutes.
- Actions: safety audit $(L_1 = 3, u_1^{\max} = 10)$; equipment check $(L_2 = 6, u_2^{\max} = 10)$; crew reassign $(L_3 = 8, u_3^{\max} = 10)$.
- Budget $B = 16.0$ minutes.
- Quantisation $\Delta = 0.01$.
- Primes $(101, 103, 107)$; rows $\boldsymbol{a}_1 = (2, 1, 0, 3)$, $\boldsymbol{a}_2 = (1, 4, 1, 2)$, $\boldsymbol{a}_3 = (3, 0, 5, 1)$.

5.2 Compute Hazard and Allocation

$$\begin{aligned}\sum e^{-\delta t_k} &= e^{-3.5} + e^{-1.75} + e^{-0.56} \approx 0.77518,\\ \lambda &= 0.010 + 0.22 \cdot 0.77518 + 0.03 = 0.21054,\\ h &= 1 - e^{-0.21054 \cdot 20} = 0.98517.\end{aligned}$$

Set scores $a = (-2.95550, -5.91099, -7.88132)$. Solve the clip rule with $\sum_i u_i^* = 16.0$ to get

$$\boldsymbol{u}^* \approx (2.7062, 5.6617, 7.6321).$$

5.3 Quantisation and Residues

Quantise $\boldsymbol{x} = [0.98517, 2.7062, 5.6617, 7.6321]$: $\tilde{\boldsymbol{x}} = (99, 271, 566, 763)$. Compute residues:

$$\begin{aligned}\langle \boldsymbol{a}_1, \tilde{\boldsymbol{x}} \rangle &= 2758 \equiv 31 \bmod 101,\\ \langle \boldsymbol{a}_2, \tilde{\boldsymbol{x}} \rangle &= 3275 \equiv 82 \bmod 103,\\ \langle \boldsymbol{a}_3, \tilde{\boldsymbol{x}} \rangle &= 3890 \equiv 38 \bmod 107.\end{aligned}$$

Make the Merkle root and sign it. The receipt includes residues $(31, 82, 38)$.

5.4 Action Plan

Run the safety audit. Check equipment. Reassign crew. Share the signed receipt with the site lead and QA.

Cryptographic note. We model H in the random oracle heuristic and use an EUF-CMA secure signature; thus $\mathcal{E} = (\{r_i\}, \{p_i\}, C, \sigma, \text{policy})$ is binding to $(A, \{p_i\})$ and unforgeable.

6 Government Example

This example shows how to use the method for dispatch surge management.

6.1 Inputs

- Dispatch call ages $45, 22, 9$ minutes.
- Parameters: $\mu = 0.009$, $\omega = 0.17$, $\delta = 0.08$, context bump 0.04.
- Horizon: $\tau = 10$ minutes.
- Actions: public info alert $(L_1 = 4, u_1^{\max} = 8)$; EOC liaison $(L_2 = 5, u_2^{\max} = 10)$; extra unit $(L_3 = 7, u_3^{\max} = 10)$.
- Budget $B = 12.2$ minutes.
- Quantisation $\Delta = 0.01$.
- Primes $(101, 103, 107)$; rows $\boldsymbol{a}_1 = (2, 1, 0, 3)$, $\boldsymbol{a}_2 = (1, 4, 1, 2)$, $\boldsymbol{a}_3 = (3, 0, 5, 1)$.

6.2 Compute Hazard and Allocation

$$\begin{aligned}\sum e^{-\delta t_k} &= e^{-3.6} + e^{-1.76} + e^{-0.72} \approx 0.68610,\\ \lambda &= 0.009 + 0.17 \cdot 0.68610 + 0.04 = 0.16564,\\ h &= 1 - e^{-0.16564 \cdot 10} = 0.80918.\end{aligned}$$

Set scores $a = (-3.23671, -4.04589, -5.66424)$. Solve the clip rule with $\sum_i u_i^* = 12.2$ to get

$$\boldsymbol{u}^* \approx (2.9878, 3.7967, 5.4153).$$

6.3 Quantisation and Residues

Quantise $\boldsymbol{x} = [0.80918, 2.9878, 3.7967, 5.4153]$: $\tilde{\boldsymbol{x}} = (81, 299, 380, 542)$. Compute residues:

$$\begin{aligned}\langle \boldsymbol{a}_1, \tilde{\boldsymbol{x}} \rangle &= 2087 \equiv 67 \bmod 101,\\ \langle \boldsymbol{a}_2, \tilde{\boldsymbol{x}} \rangle &= 2741 \equiv 63 \bmod 103,\\ \langle \boldsymbol{a}_3, \tilde{\boldsymbol{x}} \rangle &= 2685 \equiv 10 \bmod 107.\end{aligned}$$

Make the Merkle root and sign it. The receipt includes residues $(67, 63, 10)$.

6.4 Action Plan

Publish a public-info alert. Activate the EOC liaison. Dispatch one extra unit. Send the signed receipt to emergency management.

7 Software Engineering Example

This example shows how to use the method for a production incident.

7.1 Inputs

- Error ages $35, 14, 3$ minutes.
- Parameters: $\mu = 0.011$, $\omega = 0.19$, $\delta = 0.11$, context bump 0.05.
- Horizon: $\tau = 6$ minutes.
- Actions: rollback ($L_1 = 8, u_1^{\max} = 10$); feature flag off ($L_2 = 5, u_2^{\max} = 8$); on-call page ($L_3 = 3, u_3^{\max} = 6$).
- Budget $B = 11.5$ minutes.
- Quantisation $\Delta = 0.01$.
- Primes $(101, 103, 107)$; rows $\boldsymbol{a}_1 = (2, 1, 0, 3)$, $\boldsymbol{a}_2 = (1, 4, 1, 2)$, $\boldsymbol{a}_3 = (3, 0, 5, 1)$.

7.2 Compute Hazard and Allocation

$$\begin{aligned}\sum e^{-\delta t_k} &= e^{-3.85} + e^{-1.54} + e^{-0.33} \approx 0.95330,\\ \lambda &= 0.011 + 0.19 \cdot 0.95330 + 0.05 = 0.24213,\\ h &= 1 - e^{-0.24213 \cdot 6} = 0.76610.\end{aligned}$$

Set scores $a = (-6.12880, -3.83050, -2.29830)$. Solve the clip rule with $\sum_i u_i^* = 11.5$ to get

$$\boldsymbol{u}^* \approx (5.8763, 3.5780, 2.0457).$$

7.3 Quantisation and Residues

Quantise $\boldsymbol{x} = [0.76610, 5.8763, 3.5780, 2.0457]$: $\tilde{\boldsymbol{x}} = (77, 588, 358, 205)$. Compute residues:

$$\begin{aligned}\langle \boldsymbol{a}_1, \tilde{\boldsymbol{x}} \rangle &= 1911 \equiv 44 \bmod 101,\\ \langle \boldsymbol{a}_2, \tilde{\boldsymbol{x}} \rangle &= 1422 \equiv 4 \bmod 103,\\ \langle \boldsymbol{a}_3, \tilde{\boldsymbol{x}} \rangle &= 1962 \equiv 86 \bmod 107.\end{aligned}$$

Make the Merkle root and sign it. The receipt includes residues $(44, 4, 86)$.

7.4 Action Plan

Rollback immediately. Disable the feature flag. Page on-call. Send the signed receipt to SRE and Product.

8 Healthcare Example

This example shows how to use the method for emergency room staffing.

8.1 Inputs

- Patient arrivals spiked 40, 20 and 10 minutes ago.
- Parameters: $\mu = 0.008$, $\omega = 0.15$, $\delta = 0.08$, context bump 0.03.
- Horizon: $\tau = 15$ minutes.
- Actions: quick triage ($L_1 = 2, u_1^{\max} = 10$); call float nurse ($L_2 = 6, u_2^{\max} = 15$); prep overflow bay ($L_3 = 5, u_3^{\max} = 10$).
- Budget $B = 25$ minutes.
- Quantisation $\Delta = 0.01$.
- Same primes and rows as in the first example.

8.2 Compute Hazard and Allocation

$$\begin{aligned}
\sum e^{-\delta t_k} &= e^{-3.2} + e^{-1.6} + e^{-0.8} \approx 0.69200, \\
\lambda &= 0.008 + 0.15 \times 0.69200 + 0.03 = 0.14180, \\
h &= 1 - e^{-0.14180 \times 15} = 0.88080.
\end{aligned}$$

Set scores $a = -hL = (-1.76160, -5.28481, -4.40401)$. Solve the clip rule with $\sum_i u_i^* = 25$ to get $u^* \approx (6.2781, 9.8013, 8.9205)$.

8.3 Quantisation and Residues

Quantise $\boldsymbol{x} = [0.88080, 6.2781, 9.8013, 8.9205]$: $\tilde{\boldsymbol{x}} = (88, 628, 980, 892)$. Compute residues:

$$\begin{aligned}
\langle \boldsymbol{a}_1, \tilde{\boldsymbol{x}} \rangle &= 3480 \equiv 46 \bmod 101, \\
\langle \boldsymbol{a}_2, \tilde{\boldsymbol{x}} \rangle &= 5364 \equiv 8 \bmod 103, \\
\langle \boldsymbol{a}_3, \tilde{\boldsymbol{x}} \rangle &= 6056 \equiv 64 \bmod 107.
\end{aligned}$$

Make the Merkle root and sign it. The receipt includes residues $(46, 8, 64)$.

8.4 Action Plan

You open the overflow bay. You call one float nurse. You run a quick triage surge. You send the signed receipt to the shift lead and compliance.

9 Protecting and Licensing the Work

The algorithm is intellectual property. Patents give an inventor exclusive rights to make, use and sell an invention if it is novel, not obvious and useful. A license is a contract that grants rights and sets obligations for the licensor and licensee. You protect your work before licensing it.

9.1 Protection Options

- File a provisional patent to establish priority. Follow with a non-provisional or international filing within a year.
- Keep the method as a trade secret. Control access and enforce confidentiality.
- Publish defensively to block later patents.

9.2 Terms to Negotiate

When licensing, you negotiate costs, term, territory, exclusivity, rights granted and dispute rules. Costs include up-front fees, running royalties or per-decision fees. Term and territory define duration and geography. Rights define how the licensee uses the work. The contract also covers assignment, indemnity, and governing law.

9.3 Licensing Process

Licensing follows several steps:

- Find and vet potential licensees.
- Negotiate a term sheet that sets field of use, royalties and fees.
- Draft and refine the agreement.
- Sign the contract and then manage performance.

Best practice includes market research, a provisional patent, a polished pitch sheet, networking and legal counsel.

9.4 Value Stages

The value of the algorithm grows with proof and traction:

- Concept: clear equations and examples. Worth little more than reputation.
- Prototype: code that runs on test data. Worth tens of thousands in pilot fees.
- Proven pilot: a 90-day trial with a measurable lift. Worth hundreds of thousands to small contracts.
- Product: repeatable deployments with evidence. Worth millions in recurring revenue.
- Cross-sector licensing: templates for multiple domains. Worth tens of millions if growth continues.

10 Algorithms (Pseudocode)

Algorithm 1 SDA_Hazard_And_Allocate($\mathcal{D}, \mu, \omega, \delta, \zeta, \varphi, \tau, \{L_i, u_i^{\max}\}, B$)

```
$\lambda \leftarrow \mu + \omega \sum_k e^{-\delta t_k} + \zeta^\top \varphi(c_t)$
$h \leftarrow 1 - e^{-\lambda\tau}$
$a_i \leftarrow -h\, L_i \quad \forall i$
if $B \geq \sum_i u_i^{\max}$ then
    $u_i^{\leftarrow u_i^{\max}}$ for all $i$; return $h, \{u_i^{\}}$
end if
$\lambda_{\min} \leftarrow \min_i\{-a_i - u_i^{\max}\}$;   $\lambda_{\max} \leftarrow \max_i\{-a_i\}$
while $\lambda_{\max} - \lambda_{\min} > \varepsilon$ do
    $\lambda \leftarrow (\lambda_{\min} + \lambda_{\max})/2$
    $u_i(\lambda) \leftarrow clip(-a_i - \lambda, 0, u_i^{\max})$
    if $\sum_i u_i(\lambda) > B$ then
        $\lambda_{\min} \leftarrow \lambda$
    else
        $\lambda_{\max} \leftarrow \lambda$
    end if
end while
$\lambda^{\leftarrow(\lambda_{\min}+\lambda_{\max})/2}$;   $u_i^{\leftarrow clip(-a_i-\lambda^{,0,u_i^{\max})}}$
return $h, \{u_i^{\}}$
```

Algorithm 2 PCP_RGM_Receipt($\boldsymbol{x}, \Delta, \{\boldsymbol{a}_j\}, \{p_j\}, \text{meta}$)

```
$\tilde{\boldsymbol{x}} \leftarrow \text{round}(\boldsymbol{x}/\Delta)$
for $j = 1$ to $k$ do
    $r_j \leftarrow \langle \boldsymbol{a}_j, \tilde{\boldsymbol{x}} \rangle \bmod p_j$
end for
$A \leftarrow$ matrix stacking $\boldsymbol{a}_j^\top$
assert $\prod_{j=1}^{k} p_j > 2\,\|A\|_\infty\, B_{\max}$          ▷ CRT uniqueness bound
$L_j \leftarrow H(r_j \| \text{meta}_j)$;   $C \leftarrow \text{MerkleRoot}(\{L_j\})$
$\sigma \leftarrow \text{Sign}_{\text{sk}}(m, C, \{p_j\}, \text{policy})$
return $\tilde{\boldsymbol{x}}, \{r_j\}, C, \sigma$
```

Licenses

Docs License: Creative Commons Attribution 4.0 International (CC BY 4.0)

Code License: Apache License 2.0

The code and pseudocode are licensed under Apache-2.0. Full text follows.