On the (in)security of ROS

Student Seminar : Security Protocols and Applications

Max Duparc, Christophe Marciot

Based on the paper of:
Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, Mariana Raykova

04.04.2022

EPFL

What is ROS ?

ROS is the game of **R**andom inhomogeneities in an **O**verdetermined **S**olvable linear system.

Game: $\mathsf{ROS}_l(\lambda)$:
$p \leftarrow_\$ \mathbf{Pgen}(1^\lambda)$
$\left((\hat{\boldsymbol{\rho}}_i)_{i\in[l+1]}, \mathbf{c}\right) \leftarrow_\$ \mathcal{A}^{\mathsf{H}_{\mathsf{ROS}}}(p)$
return $\left(\forall i \neq j, \hat{\boldsymbol{\rho}}_i \neq \hat{\boldsymbol{\rho}}_j \ \wedge \langle \hat{\boldsymbol{\rho}}_i, \mathbf{c}\rangle = \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_i)\right)$

- $\mathbf{Pgen}$ a prime generator with $\lceil \log_2(p) \rceil = \lambda$
- $\hat{\boldsymbol{\rho}}_i, \mathbf{c} \in \mathbb{Z}_p^l$
- $\mathsf{H}_{\mathsf{ROS}}$ a random oracle with image in $\mathbb{Z}_p$
- $\mathcal{A}^{\mathsf{H}_{\mathsf{ROS}}}$ a probabilistic poly(λ) time adversary.

EPFL

Table of Contents

EPFL

Table of Contents

ROS Attack

Theorem [2020] (ROS-attack)

for $l > \lambda$

$$\text{ROS}_l(\lambda) \text{ is easy}$$

Where *hard* means that for every adversary in poly(λ) time

$$\mathbb{P}[\text{ROS}_l(\lambda) = 1] = \lambda^{-\omega(1)}$$

- Let $\boldsymbol{\rho} = \rho_0 + \sum_{i=1}^{l} \rho_i x_i \in \mathbb{Z}_p[x_1, \cdots, x_l]$ and $\hat{\boldsymbol{\rho}} = (\rho_1, \cdots, \rho_l) \in \mathbb{Z}_p^l$
  See that $\mathbf{c} \in \mathbb{Z}_p^l$

$$\boldsymbol{\rho}(\mathbf{c}) = \langle \hat{\boldsymbol{\rho}}, \mathbf{c} \rangle - \rho_0$$

EPFL

ROS Adversary (1)

- For $i = 1, \cdots, l$, $b = \{0, 1\}$

$$\boldsymbol{\rho}_i^b = 2^b x_i$$

$$c_i^b = 2^{-b} \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_i^b)$$

- If $\exists i^*$ such that $c_{i^*}^0 = c_{i^*}^1$

$$\textbf{return } (\hat{\boldsymbol{\rho}}_1^0, \cdots, \hat{\boldsymbol{\rho}}_l^0, \hat{\boldsymbol{\rho}}_{i^*}^1) \text{ and } \mathbf{c} = (c_1^0, \cdots, c_l^0)$$

- Otherwise, define

$$\mathbf{f}_i = \frac{x_i - c_i^0}{c_i^1 - c_i^0}$$

we have that $\mathbf{f}_i(c_i^b) = b$.

ROS Adversary (2)

$$\text{Let } \boldsymbol{\rho}_{l+1} = \sum_{i=1}^{l} 2^{i-1}\mathbf{f}_i \qquad y = \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_{l+1}) + \boldsymbol{\rho}_{l+1}(0).$$

- See y in binary as

$$y = \sum_{i=1}^{l} 2^{i-1} b_i \mod p$$

- **return** $(\hat{\boldsymbol{\rho}}_1^{b_1}, \cdots, \hat{\boldsymbol{\rho}}_l^{b_l}, \hat{\boldsymbol{\rho}}_{l+1})$ and $\mathbf{c} = (c_1^{b_1}, \cdots, c_l^{b_l})$
- Thoses are valid solutions:
  - for $i = 1, \cdots, l$, $\langle \hat{\boldsymbol{\rho}}_i, \mathbf{c} \rangle = 2^{b_i - b_i} \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_i^b) = \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_i^b)$.
  - $\langle \hat{\boldsymbol{\rho}}_{l+1}, \mathbf{c} \rangle = \boldsymbol{\rho}_{l+1}(\mathbf{c}) - \boldsymbol{\rho}_{l+1}(0) = \sum_{i=1}^{l} 2^{i-1} \mathbf{f}_i(c_i^{b_i}) - \boldsymbol{\rho}_{l+1}(0) =$
    $= \sum_{i=1}^{l} 2^{i-1} b_i - \boldsymbol{\rho}_{l+1}(0) = y - \boldsymbol{\rho}_{l+1}(0) = \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_{l+1})$.

EPFL

Table of Contents

Wagner's ROS Attack

Theorem [2002] (Wagner's ROS Attack)

for any l, $\exists$ $\mathcal{A}$ an adversary that wins $\mathsf{ROS}_l(\lambda)$ using:

$$\text{time} : \mathcal{O}\big((l+1)2^{\lambda/(1+\lfloor \log_2(l+1) \rfloor)}\big)$$
$$\text{memory} : \mathcal{O}\big(\log_2(l+1)2^{\lambda/(1+\lfloor \log_2(l+1) \rfloor)}\big)$$

This is sub exponential *but* slowly distantiates itself from $\mathcal{O}(2^{\lambda})$. For example, taking $l = 2^{\sqrt{\lambda}} - 1$, it is in time $\mathcal{O}(2^{2\sqrt{\lambda}})$.
This adversary relies on another math problem: *the k-sum problem.*

k-list problem

Definition (k-list problem in a group G)

Let $\mathcal{L}_1, \cdots, \mathcal{L}_k$ be random lists of element in G and let $H \subseteq G$. The k-list problem consists in finding $x_i \in \mathcal{L}_i$ such that

$$x_1 + x_2 + \cdots + x_k \in H$$

If $|H| = 1$, this is called the k-sum problem. This is a generalisation of the birthday paradox problem.

It is a fundamental problem in cryptography

Theorem [2001] (Wei Dai)

If the k-sum problem over a cyclic group $G = \langle g \rangle$ can be solved in time $\mathcal{O}(t)$, then the discrete log with respect to g can be found in time $\mathcal{O}(t)$.

Wagner's ROS-Attack

Consider

$$M_i = \left\{ \boldsymbol{\rho}_i = \rho_i x_i \middle| \rho_i \in \mathbb{Z}_p^\times \right\} \text{ and corresponding lists } \mathcal{L}_i = \left\{ \mathbf{c}_i = \rho_i^{-1} \mathsf{H}_{\mathsf{ROS}}(\boldsymbol{\rho}_i) | \boldsymbol{\rho}_i \in M_i \right\}$$

- Let $\hat{\boldsymbol{\rho}}_{l+1} = (1, \cdots, 1)$. Solve the k-sum problem for

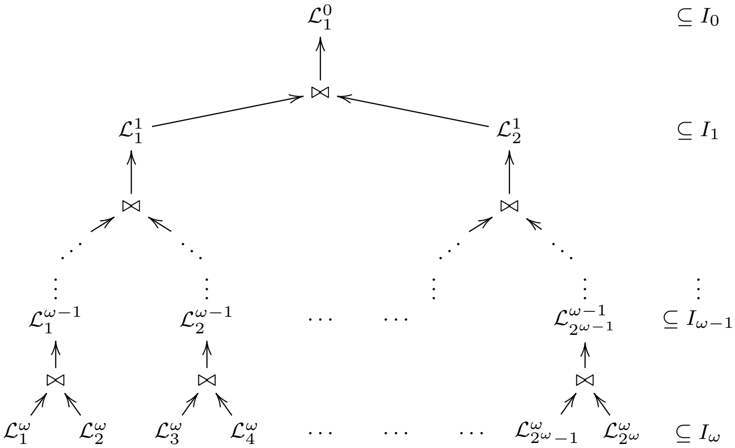
$$\langle \hat{\boldsymbol{\rho}}_{l+1}, (c_1, \cdots, c_l) \rangle = c_1 + c_2 + \cdots + c_l = \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_{l+1}), c_i \in \mathcal{L}_i$$

- **return** $(\hat{\boldsymbol{\rho}}_1, \cdots, \hat{\boldsymbol{\rho}}_l, \hat{\boldsymbol{\rho}}_{l+1})$ and $\mathbf{c} = (c_1, \cdots, c_l)$.

So, the question is: "*do we have a quick algorithm for k-sum in $\mathbb{Z}_p$?*"

- Sadly k-sum is in time $\Omega(2^{\frac{|G|}{k}})$,
- however, fascinating algorithms exist.

Wagner's k-list algorithm (1)



Wagner's k-list algorithm (2)

Let H be any interval of $\mathbb{Z}_p$. w.l.o.g, we see $\mathbb{Z}_p = [-\frac{p-1}{2}, \frac{p-1}{2}]$ and $H \subseteq [-\lfloor \frac{p-1}{2^{\omega L+1}} \rfloor, \lfloor \frac{p-1}{2^{\omega L+1}} \rfloor]$

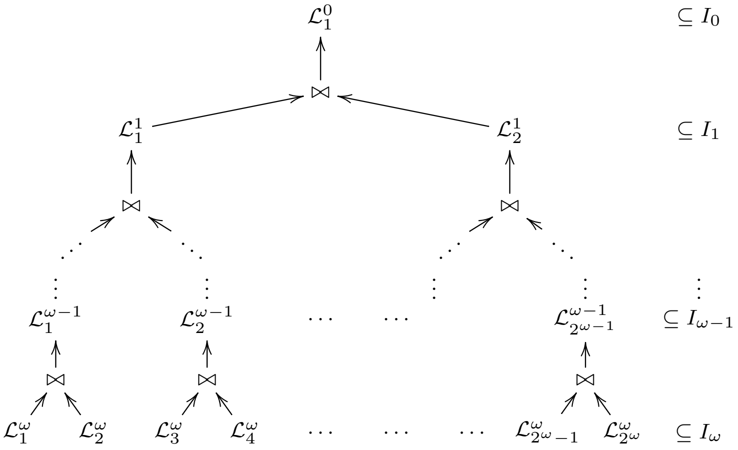
$$\text{Let } I_{-1} = H,\ I_i = \left[-\left\lfloor \frac{p-1}{2^{(\omega-i)L+1}} \right\rfloor, \left\lfloor \frac{p-1}{2^{(\omega-i)L+1}} \right\rfloor \right], i = 0, \cdots, \omega$$

Algorithm: k-list$\big(\{\mathcal{L}^\omega\}_{[2^\omega]}\big)$: with $|\mathcal{L}_i^\omega| = 2^L$
for $i = \omega$ **downto** 1 **do**
  for $j \in [2^{i-1}]$ **do**
    $\mathcal{L}_j^{i-1} = \{a + b \mid a \in \mathcal{L}_{2j-1}^i, b \in \mathcal{L}_{2j}^i, a + b \in I_{i-1}\}$
  end
end
if $\mathcal{L}^0 \cap I_{-1} = \emptyset$ **then**
  return $\perp$
end
return $(l_1, \cdots, l_n)$, $l_1 + l_2 + \cdots + l_k = s \in I_{-1}$

Wagner's conjecture: Provided $\frac{p}{|H|} \leqslant 2^{\omega L}$ with ω, L optimal approximation of H, this k-list algorithm on 2^ω lists of 2^L uniformly random elements in $\mathbb{Z}_p$ has constant failure probability.

Wagner's k-list algorithm (3)

- $\bowtie$ denote the merging of the two lists, using a Hash-joint / Merge-sort.



time : $\mathcal{O}(2^{\omega+L})$

memory : $\mathcal{O}(\omega 2^L)$

EPFL

ROS Generalised Attack

Theorem [2020] (ROS Generalised attack)

For $l \leqslant \lambda$, $\exists \mathcal{A}$ an adversary that wins $\mathrm{ROS}_l(\lambda)$ in an efficient sub exponential.

For $l \geqslant \max\left\{2^\omega - 1, \lceil 2^\omega - 1 + \lambda - (\omega+1)L \rceil\right\}$, the adversary runs in :

$$\text{time} : \mathcal{O}(2^{\omega+L})$$
$$\text{memory} : \mathcal{O}(\omega 2^L)$$

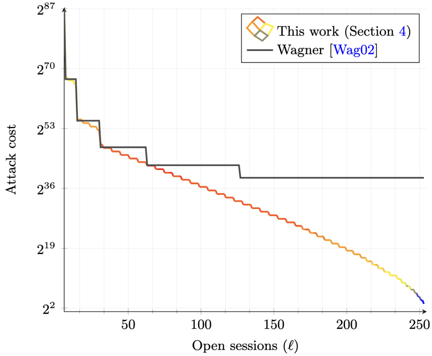
EPFL

ROS Generalised Attack idea

1. let $k_1 = 2^\omega - 1$, $k_2 = \max(0, \lceil \lambda - (\omega + 1)L \rceil)$, set $k = k_1 + k_2$.
2. Run ROS-attack on $\mathbb{Z}_{2^{k_2}} \subseteq \mathbb{Z}_p$.
3. Run Wagner's k-list attack on $k_1 + 1 = 2^\omega$ with lists of size 2^L to find a 2^ω-list solution in $\mathbb{Z}_{2^{k_2}}$.
4. Merge both solutions. (See details in appendix).

EPFL

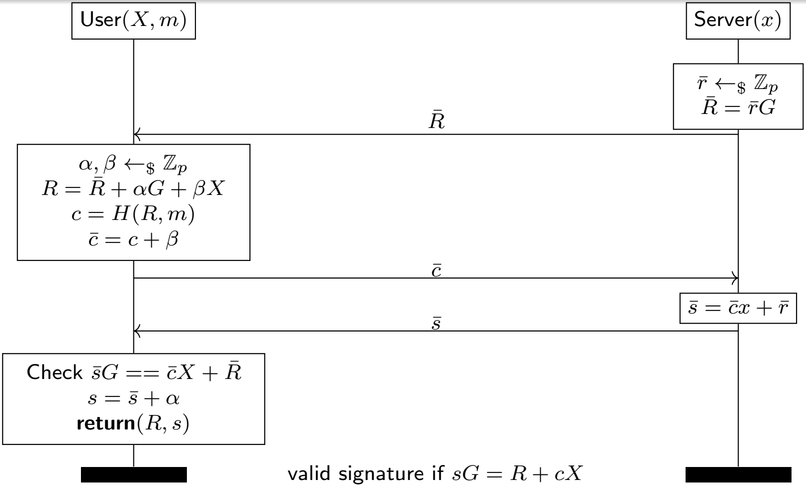
ROS Generalised Attack in action



	λ	l	time	memory
Brute force	256	197	2^{128}	2^{128}
WROSA	256	197	2^{39}	$7{\cdot}2^{32}$
ROSGA	256	197	2^{20}	$5{\cdot}2^{15}$
WROSA	512	253	2^{71}	$7{\cdot}2^{64}$
ROSGA	512	253	2^{53}	$6{\cdot}2^{46}$
WROSA	512	513	2^{60}	$7 \cdot 2^{53}$
ROSGA	512	513	poly(λ)	poly(λ)

Table of Contents

Schnorr blind signature (SBS) protocol [2001]



- $X = xG$
- G generator of $\mathbb{G}$, group of order p
- H a hash fonction.

SBS attack using ROS

Theorem [2001] (SBS attack using ROS)

Given l the number of parallel section doable using SBS.
Given $\mathcal{A}$ an adversary of $\text{ROS}_l(\lambda)$ that wins in time $\mathcal{O}(t)$.

- We can construct an adversary $\mathcal{B}$ that breaks UFKMA(SBS) in time $\mathcal{O}(t)$.

Corollary [2020]

If $l > \log_2(p)$

UFKMA(SBS) is insecure

If $l \leqslant \log_2(p)$, it is sub exponential breakable.

EPFL

SBS attack using ROS

Let $m_1, \cdots, m_l$ be arbitrary messages, m_{l+1} be the desired forged message.

- Get $\overline{\mathbf{R}} = (\overline{R}_1, \cdots, \overline{R}_l)$ by opening l parallel sessions with the server (fixed x).
- Using $\mathcal{A}$, get $\boldsymbol{\rho}_1, \ldots, \boldsymbol{\rho}_{l+1}, \mathbf{c} \in \mathbb{Z}_p^l$, such that

$$\forall i = 1, \cdots, l+1, \ \langle \boldsymbol{\rho}_i, \mathbf{c} \rangle = H(R_i, m_i) \qquad \text{with } R_i = \sum_{j=1}^{l} \boldsymbol{\rho}_{i,j} \overline{R}_j$$

- Send $\overline{c_i} = c_i$ as an answer to $\overline{R}_i$ to the server and get $\bar{\mathbf{s}} = (\overline{s_1}, \cdots, \overline{s_l})$.
- For $i = 1, \cdots, l+1$ define $s_i = \sum_{j=1}^{l} \boldsymbol{\rho}_{i,j} \bar{\mathbf{s}}_j$
- For $i = 1, \cdots, l+1$ **return** (R_i, s_i) as signatures for m_i. Those are valid. Indeed

$$s_i G = \sum_{j=1}^{l} \boldsymbol{\rho}_{i,j} \bar{\mathbf{s}}_j G = \Big(\sum_{j=1}^{l} \boldsymbol{\rho}_{i,j} (\bar{c_j} x + r_j) \Big) G = \langle \boldsymbol{\rho}_i, \mathbf{c} \rangle x G + \sum_{j=1}^{l} \boldsymbol{\rho}_{i,j} r_j G = c_i X + R_i$$

Other signature schemes affected (1)

Okamoto-Schnorr blind signatures

Okamoto-Schnorr blind signatures are of the form (R, s, t) such that $sG + tH - cX = R$. G, H generators of $\mathbb{G}$.

It was proven that for $l < \log_Q(p)$, UFKMA(OSBS) is secure[a]

Now, for $l > \log_2(p)$, UFKMA(OSBS) is insecure

[a] where Q is the number of queries to H_{ROS}

Other signature schemes affected (2)

- CoSi is a multi-signature scheme with signatures (c, s) such that $c = H(sG - c\text{pk}, m)$.

  If $l > \log_2(p)$, UFKMA-(CoSi) is unsecure

- Threshold signature scheme like GJKR07 was[1] also insecure for $l > \log_2(p)$.
- Partially blind signatures like Abe-Okamoto.
- **Every cryptosystem whose security is based on** ROS **is potentially at risk!**

[1] this attack has now been fixed

EPFL

Conclusion

- We have a polytime attack on $\mathsf{ROS}_l(\lambda)$ for $l > \lambda$
- A good subexponential attack on $\mathsf{ROS}_l(\lambda)$ for $l \leqslant \lambda$
- Many signature schemes are no longer secure.
- Always be cautious about parallel sessions !

Bibliography

Fabrice Benhamouda, Tancrède Lepoint, Julian Loss, Michele Orrù, and Mariana Raykova, *On the (in) security of ros*, Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2021, pp. 33–53.

Claus Peter Schnorr, *Security of blind discrete log signatures against interactive attacks*, International Conference on Information and Communications Security, Springer, 2001, pp. 1–12.

David Wagner, *A generalized birthday problem*, Annual International Cryptology Conference, Springer, 2002, pp. 288–304.

Appendix: ROS Generalised Attack proof (1)

- Let $k_1 = 2^\omega - 1$, $k_2 = \max(0, \lceil \lambda - (\omega+1)L \rceil)$, set $l = k_1 + k_2$.
- $\forall i \in [k_2]$, $b = 0, 1$ we define

$$\boldsymbol{\rho}_i = 2^b x_i \qquad\qquad c_i^b = 2^{-b} \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_i^0)$$

- If $\exists i^*$ such that $c_{i^*}^0 = c_{i^*}^1$, set $\boldsymbol{\rho}_i = x_i$, $c_i = \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_i)$ for $i \in [k_2+1, l]$ **return** $(\boldsymbol{\rho}_1^0, \cdots, \boldsymbol{\rho}_{k_2}^0, \boldsymbol{\rho}_{k_2+1}, \cdots, \boldsymbol{\rho}_l^0, \boldsymbol{\rho}_{i^*}^1)$ and $(c_1^0, \cdots, c_l)$
- Otherwise, define

$$\mathbf{f}_i = \frac{x_i - c_i^0}{c_i^1 - c_i^0}$$

$$\bar{\boldsymbol{\rho}}_{l+1} = \sum_{i=1}^{k_2} 2^{i-1}\mathbf{f}_i + \left\lfloor \frac{p-1}{2^{(\omega+1)L+1}} \right\rfloor - \sum_{i=k_2+1}^{l} x_i$$

EPFL

Appendix: ROS Generalised Attack proof (2)

- For $i = k_2 + 1, \cdots, l+1$

$$H_i(\alpha) = \begin{cases} \alpha^{-1} H_{\mathsf{ROS}}(\boldsymbol{\rho}) & \text{with } \boldsymbol{\rho} = \alpha x_i \text{ if } i \in [k_2+1, l] \\ \alpha^{-1} H_{\mathsf{ROS}}(\boldsymbol{\rho}) - \bar{\boldsymbol{\rho}}_{l+1} & \text{with } \boldsymbol{\rho} = \alpha \bar{\boldsymbol{\rho}}_{l+1} \text{ if } i = l+1 \end{cases}$$

- Get $\rho^*_{k_2+1}, \cdots, \rho^*_{l+1}$ by running k-list$\Big(\big\{H_i([2^L])\big\}_{i \in [k_1+1]}\Big)$.

$$\text{define } \boldsymbol{\rho}^*_i = \begin{cases} \rho^*_i x_i & i \in [k_2+1, l] \\ \rho^*_{l+1} \bar{\boldsymbol{\rho}}_{l+1} & i = l+1 \end{cases}$$

$$y^*_i = H_i(\rho^*_i) = \begin{cases} (\rho^*_i)^{-1} H_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}^*_i) & i \in [k_2+1, l] \\ (\rho^*_{l+1})^{-1} H_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}^*_{l+1}) - \bar{\boldsymbol{\rho}}_{l+1} & i = l+1 \end{cases}$$

$$s = \sum_{k_2+1}^{l} y^*_i \in \left[-\left\lfloor \frac{p-1}{2^{(\omega+1)L+1}} \right\rfloor, \left\lfloor \frac{p-1}{2^{(\omega+1)L+1}} \right\rfloor \right]$$

$$\text{See } s + \left\lfloor \frac{p-1}{2^{(\omega+1)L+1}} \right\rfloor = \sum_{i=1}^{k_2} 2^{i-1} b_i$$

EPFL

Appendix: ROS Generalised Attack proof (3)

$$\text{define } \hat{\boldsymbol{\rho}}_i = \begin{cases} \hat{\boldsymbol{\rho}}_i^{b_i} & i \in [1, k_2] \\ \hat{\boldsymbol{\rho}}_i^{*} & i \in [k_2+1, l+1] \end{cases}$$

$$c_i = \begin{cases} c_i^{b_i} & i \in [1, k_2] \\ y_i^{*} & i \in [k_2+1, l] \end{cases}$$

- **return** $(\hat{\boldsymbol{\rho}}_1, \cdots, \hat{\boldsymbol{\rho}}_{l+1})$ and $(c_1, \cdots, c_l)$.

Thoses are valid solutions.

Appendix: ROS Generalised Attack proof (4)

$$\langle \hat{\boldsymbol{\rho}}_i, \mathbf{c} \rangle = \begin{cases} \boldsymbol{\rho}_i^{b_i}(\mathbf{c}) = 2^{b_i} \mathbf{c}_i^{b_i} = \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_{\mathsf{i}}^{\mathsf{b_i}}) & i \in [1, k_2] \\ \boldsymbol{\rho}_i^*(\mathbf{c}) = \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_{\mathsf{i}}^*) & i \in [k_2+1, l] \end{cases}$$

$$\begin{aligned} \langle \hat{\boldsymbol{\rho}}_{l+1}, \mathbf{c} \rangle &= \boldsymbol{\rho}_{l+1}(\mathbf{c}) - \boldsymbol{\rho}_{l+1}(0) \\ &= \rho_{l+1}^* \left(\sum_{i=1}^{k_2} 2^{i-1} \mathbf{f}_i(\mathbf{c}) - \left\lfloor \frac{p-1}{2^{(\omega+1)L+1}} \right\rfloor - \sum_{i=k_2+1}^{l} c_i - \overline{\boldsymbol{\rho}}_{l+1}(0) \right) \\ &= \rho_{l+1}^* \left(\sum_{i=1}^{k_2} 2^{i-1} b_i - \left\lfloor \frac{p-1}{2^{(\omega+1)L+1}} \right\rfloor - \sum_{i=k_2+1}^{l} y_i^* - \overline{\boldsymbol{\rho}}_{l+1}(0) \right) \\ &= \rho_{l+1}^* \left(s - \sum_{i=k_2+1}^{l} y_i^* - \overline{\boldsymbol{\rho}}_{l+1}(0) \right) \\ &= \rho_{l+1}^* \left(y_{l+1}^* - \overline{\boldsymbol{\rho}}_{l+1}(0) \right) \\ &= \mathsf{H}_{\mathsf{ROS}}(\hat{\boldsymbol{\rho}}_{l+1}^*) \end{aligned}$$

EPFL