Modeling Preprocessing Attacks

Auxiliary-Input Model/Bit-Fixing Model

14

Bit-Fixing Model (ROM)

- ▷ $\mathcal{A}_{\mathsf{pre}}$ fixes random oracle $\mathsf{H}(\cdot)$ at P locations
- ▷ $\mathcal{A}_{\mathsf{on}}$ initially knows nothing about remaining unfixed values (picked uniformly at random)

multi-user security of **standardized implementations** of **short** Schnorr signatures against preprocessing attacks **in ROM+GGM [New Result]**

- Much easier to prove security
- Not a compelling model for preprocessing attacks!
- **Usage:** lower bound in Bit-Fixing Model $\Rightarrow$ lower bound in Auxiliary-Input Model [Coretti et al., EUROCRYPT 2018]

$$\varepsilon_{\mathsf{AI}}(S,q) \leq \varepsilon_{\mathsf{BF}}(P,q) + \mathcal{O}(Sq/P)$$

[Coretti et al., EUROCRYPT 2018]

Bit-Fixing ROM $\Rightarrow$ Auxiliary-Input ROM

[Coretti et al., CRYPTO 2018]

Bit-Fixing GGM $\Rightarrow$ Auxiliary-Input GGM

Bit-Fixing ICM $\Rightarrow$ Auxiliary-Input ICM

Bit-Fixing RPM $\Rightarrow$ Auxiliary-Input RPM only showed in a single idealized model!

We extend the result to work in **multiple** idealized models!

- **Not** a black-box extension (the hint may simultaneously depend on **all** of the idealized primitives)

$$\varepsilon_{\mathsf{AI}}(S,q) \le \varepsilon_{\mathsf{BF}}(P,q) + \mathcal{O}(Sq/P)$$

Modeling Preprocessing Attacks

Auxiliary-Input Model/Bit-Fixing Model

Bit-Fixing Model (ROM)

- ▷ $\mathcal{A}_{\text{pre}}$ fixes random oracle $\mathsf{H}(\cdot)$ at P locations
- ▷ $\mathcal{A}_{\text{on}}$ initially knows nothing about remaining unfixed values (picked uniformly at random)

➕ Much easier to prove security

➖ Not a compelling model for preprocessing attacks!

- **Usage:** lower bound in Bit-Fixing Model ⇒ lower bound in Auxiliary-Input Model [Coretti et al., EUROCRYPT 2018]

$$\varepsilon_{\mathsf{AI}}(S,q) \le \varepsilon_{\mathsf{BF}}(P,q) + \mathcal{O}(Sq/P)$$

multi-user security of **standardized implementations** of **short** Schnorr signatures against preprocessing attacks **in** ROM+GGM [New Result]

[Coretti et al., EUROCRYPT 2018]
Bit-Fixing ROM ⇒ Auxiliary-Input ROM

[Coretti et al., CRYPTO 2018]
Bit-Fixing GGM ⇒ Auxiliary-Input GGM
Bit-Fixing ICM ⇒ Auxiliary-Input ICM
Bit-Fixing RPM ⇒ Auxiliary-Input RPM

only showed in a single idealized model!

We extend the result to work in **multiple** idealized models!

- **Not** a black-box extension (the hint may simultaneously depend on **all** of the idealized primitives)

$$\varepsilon_{\mathsf{AI}}(S,q) \le \varepsilon_{\mathsf{BF}}(P,q) + \mathcal{O}(Sq/P)$$

Bridge-Finding Game

In the Bit-Fixing GGM

Fix $(t_1, \tau(t_1)), \ldots, (t_P, \tau(t_P))$

preprocessing phase

online phase

$\tau(1), \tau(x_1), \tau(x_2), \tau(x_3)$ ←

$\mathtt{Mult}(\tau(x_1), \tau(x_2))$ →

$\tau(x_1 + x_2)$ ←

$\mathtt{Inv}(\tau(x_1 + x_2))$ →

$\tau(-x_1 - x_2)$ ←

$\mathtt{Mult}(\tau(x_1), \tau(1))$ →

$\tau(x_1 + 1)$ ←

$\mathtt{Mult}(\tau(x_2), \mathfrak{y})$ →

$\tau(x_2 + 7)$ ←

Restricted Discrete-Log Oracle

(x_1, x_2, x_3)

$\tau(y) = \vec{a} \cdot \vec{x} + b$

$\mathcal{L}$

$\tau(y)$	$\vec{a}$	b
$\tau(t_1)$	$(0,0,0)$	t_1
$\cdots$	$\cdots$	$\cdots$
$\tau(t_P)$	$(0,0,0)$	t_P
$\tau(1)$	$(0,0,0)$	1
$\tau(x_1)$	$(1,0,0)$	0
$\tau(x_2)$	$(0,1,0)$	0
$\tau(x_3)$	$(0,0,1)$	0
$\tau(x_1 + x_2)$	$(1,1,0)$	0
$\tau(-x_1 - x_2)$	$(-1,-1,0)$	0
$\tau(x_1 + 1)$	$(1,0,0)$	1
$\mathfrak{y}$	$(0,0,0)$	7
$\tau(x_2 + 7)$	$(0,1,0)$	7
$\cdots$	$\cdots$	$\cdots$