Signatures with Flexible Public Key: Introducing Equivalence Classes for Public Keys

Michael Backes, Lucjan Hanzlik, Kamil Kluzcniak, **Jonas Schneider**

UNIVERSITÄT
DES
SAARLANDES

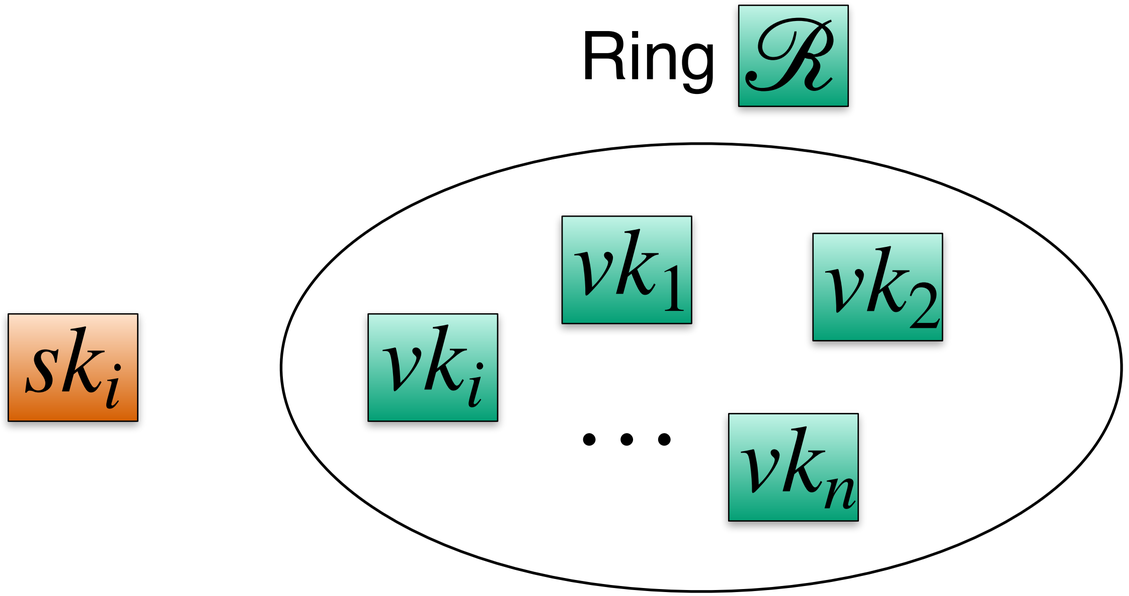
CISPA
HELMHOLTZ-ZENTRUM i. G.

This Talk

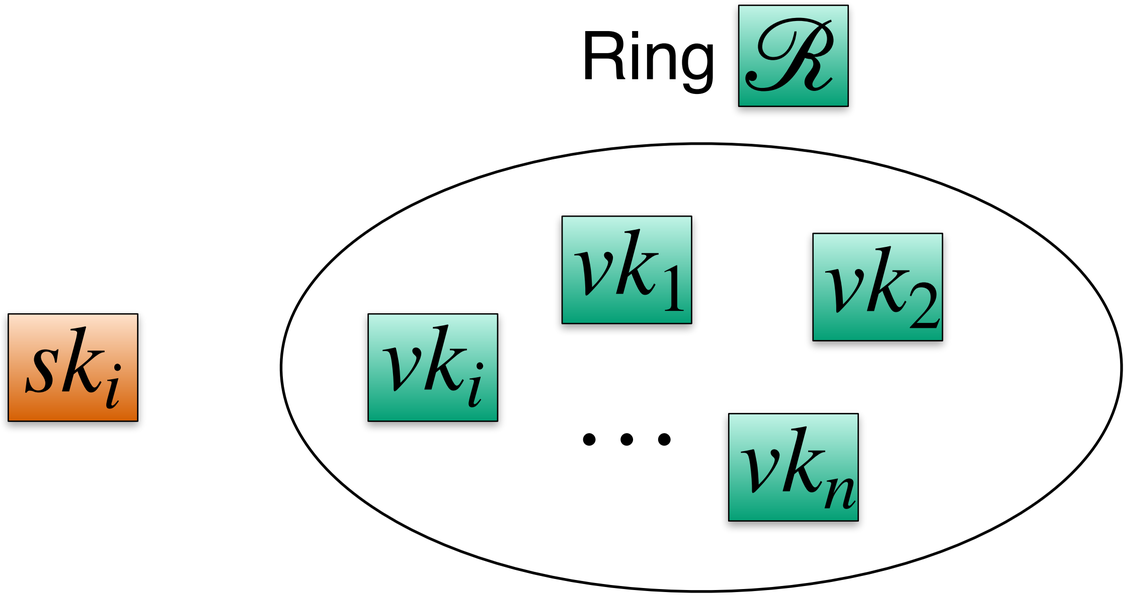
- New primitive! Signatures with Flexible Public Key
  - Applications to
    - sub-linear size Ring Signatures from *falsifiable assumptions without trusted setup*
    - efficient standard model Group Signatures in combination with SPS-EQ

Ring Signatures [Rivest-Shamir-Tauman, 2001]

Ring Signatures [Rivest-Shamir-Tauman, 2001]

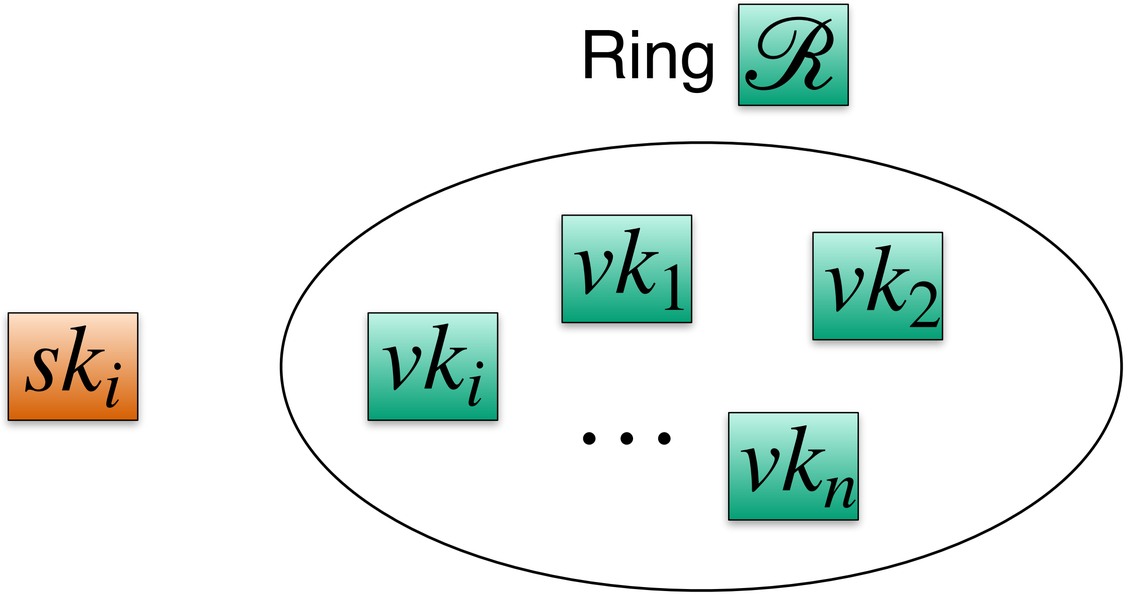


Ring Signatures [Rivest-Shamir-Tauman, 2001]



$$\sigma \leftarrow \mathrm{Sign}(sk_i, m, \mathscr{R})$$

Ring Signatures [Rivest-Shamir-Tauman, 2001]

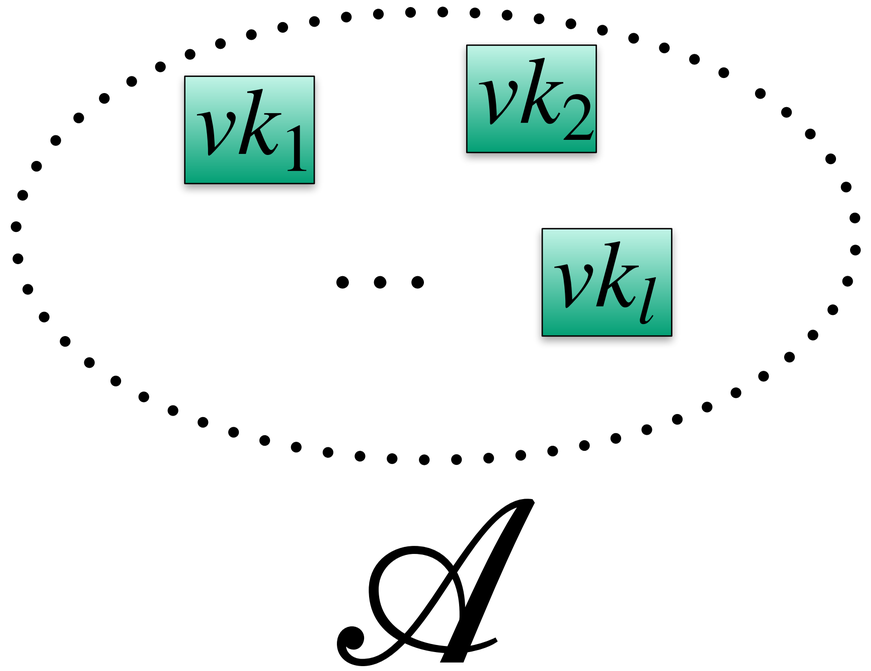


$$\sigma \leftarrow \mathsf{Sign}(sk_i, m, \mathscr{R})$$

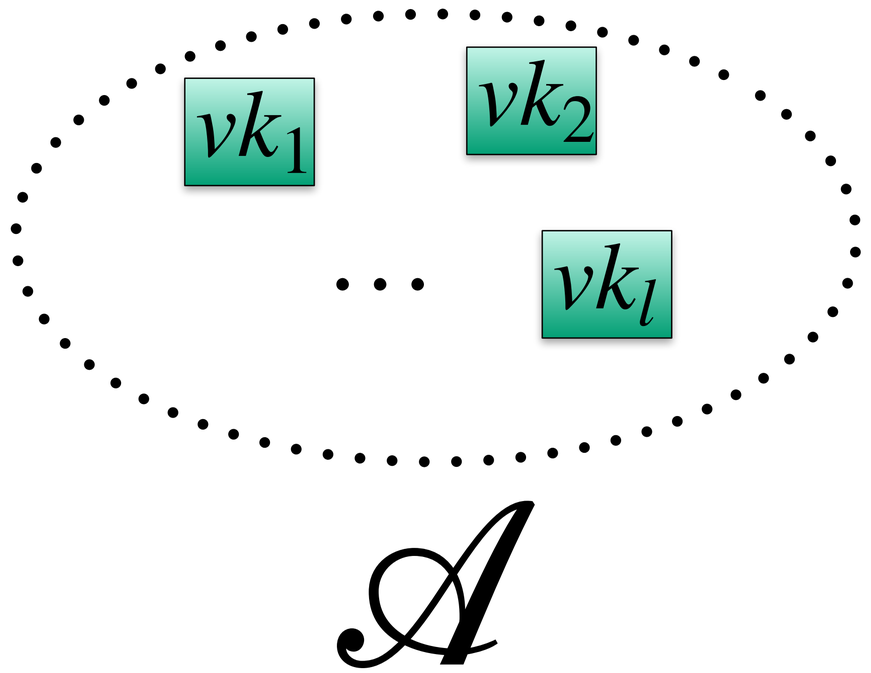
$$\texttt{accept/reject} \leftarrow \mathsf{Verify}(\sigma, m, \mathscr{R})$$

Ring Signatures: Unforgeability [Bender-Katz-Morselli, 2006]

Ring Signatures: Unforgeability [Bender-Katz-Morselli, 2006]



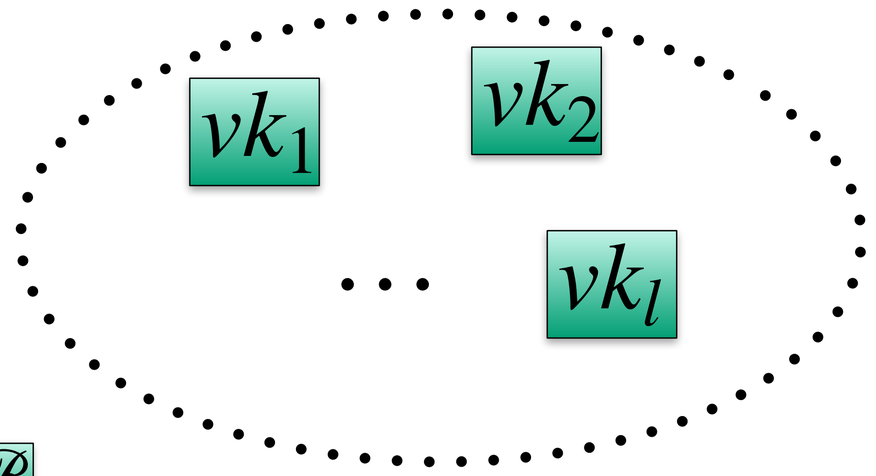
Ring Signatures: Unforgeability [Bender-Katz-Morselli, 2006]



Signing Oracle

Corruption Oracle

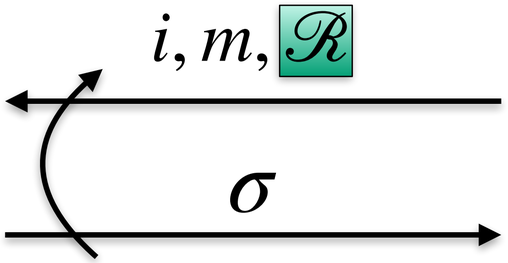
Ring Signatures: Unforgeability [Bender-Katz-Morselli, 2006]



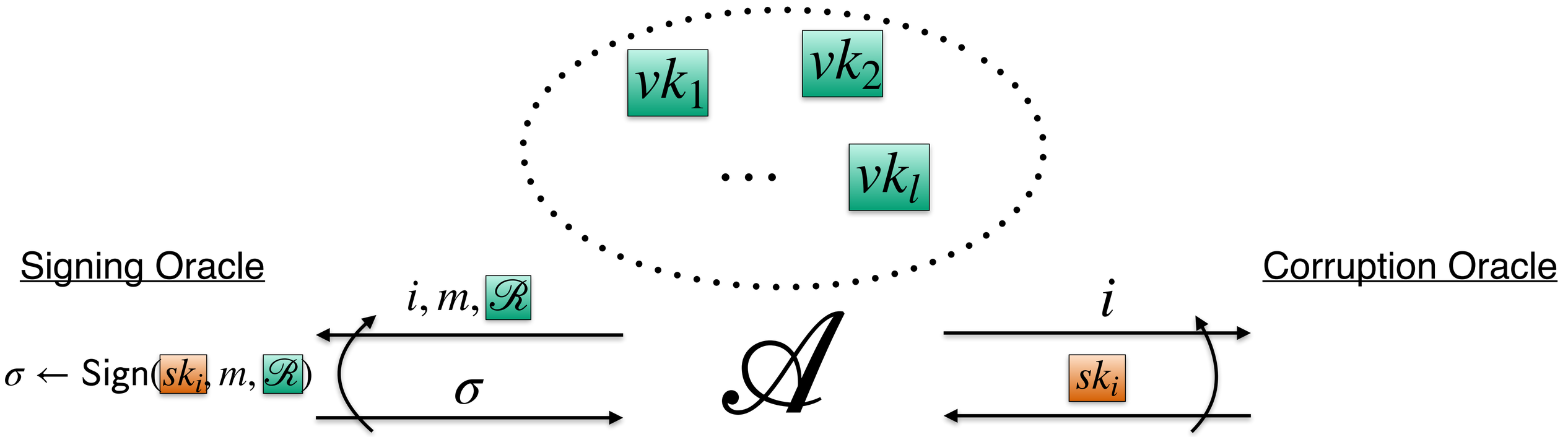
Signing Oracle

Corruption Oracle

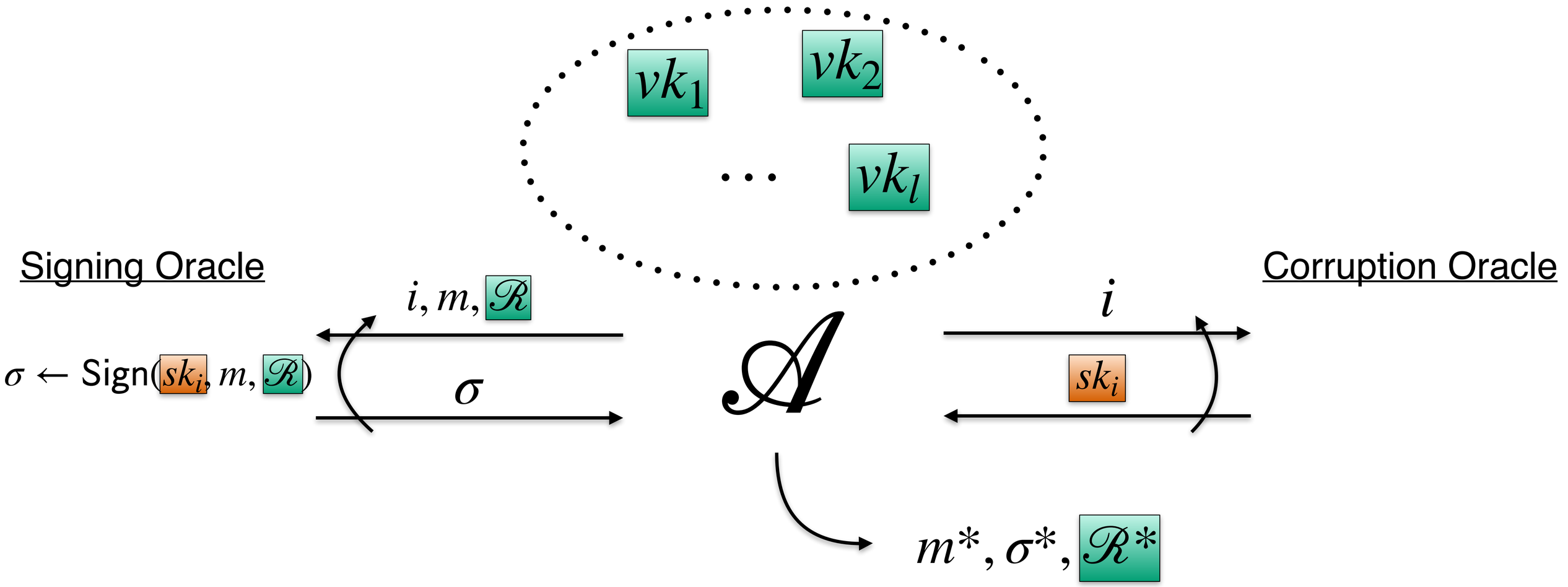
$\sigma \leftarrow \text{Sign}(sk_i, m, \mathcal{R})$



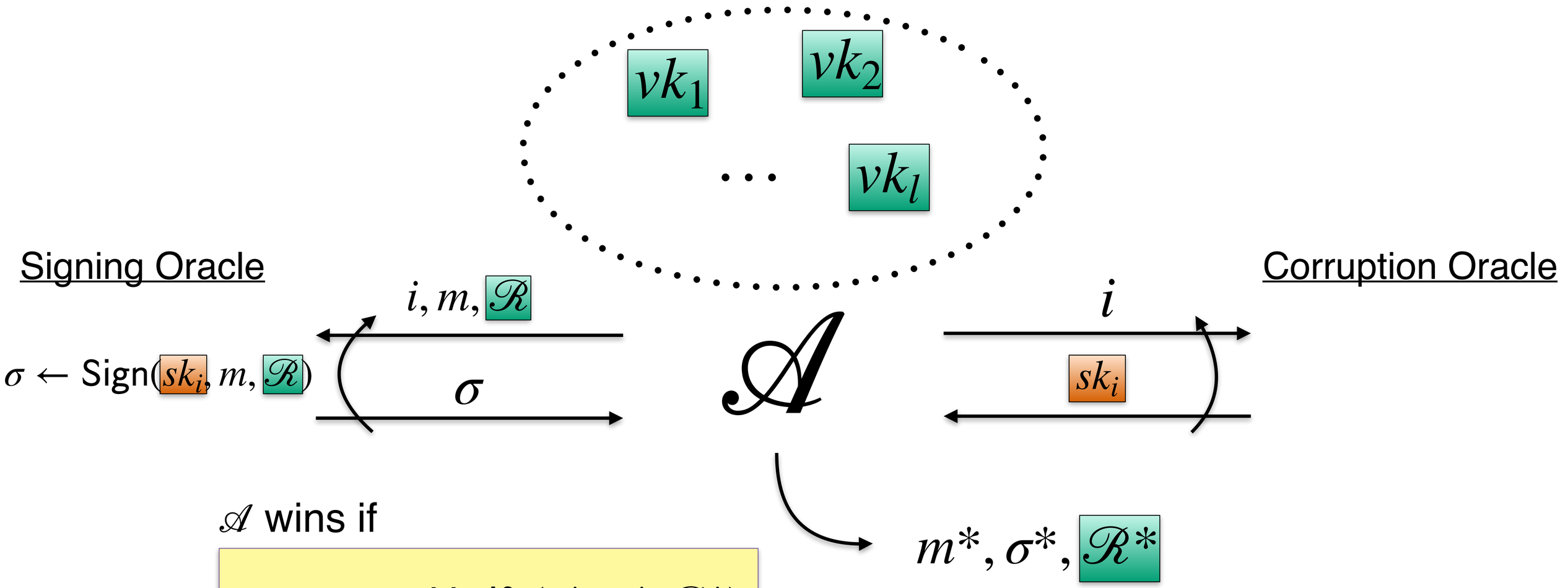
Ring Signatures: Unforgeability [Bender-Katz-Morselli, 2006]



Ring Signatures: Unforgeability [Bender-Katz-Morselli, 2006]



Ring Signatures: Unforgeability [Bender-Katz-Morselli, 2006]



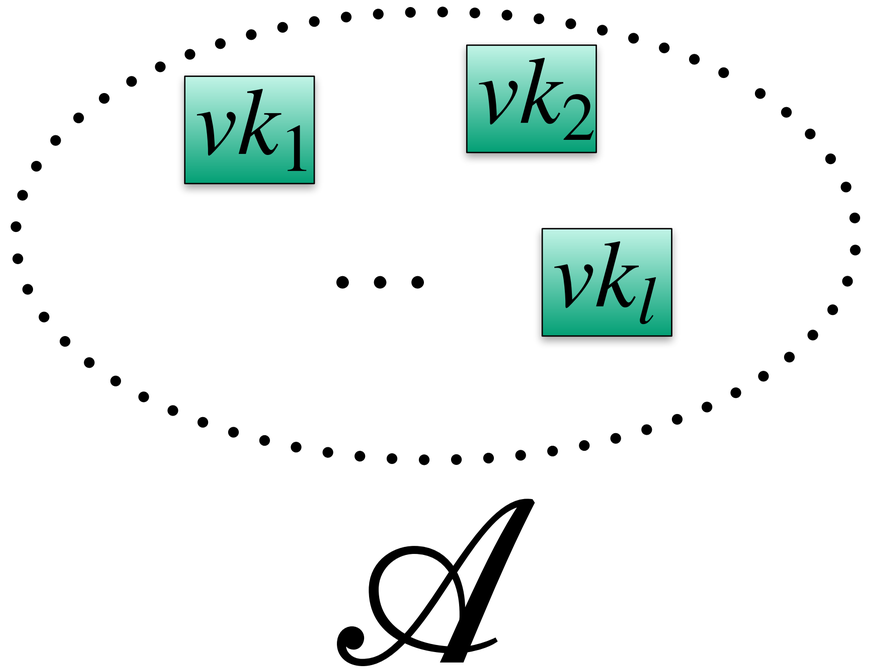
$\mathcal{A}$ wins if

$\texttt{accept} \leftarrow \text{Verify}(\sigma^*, m^*, \mathcal{R}^*)$

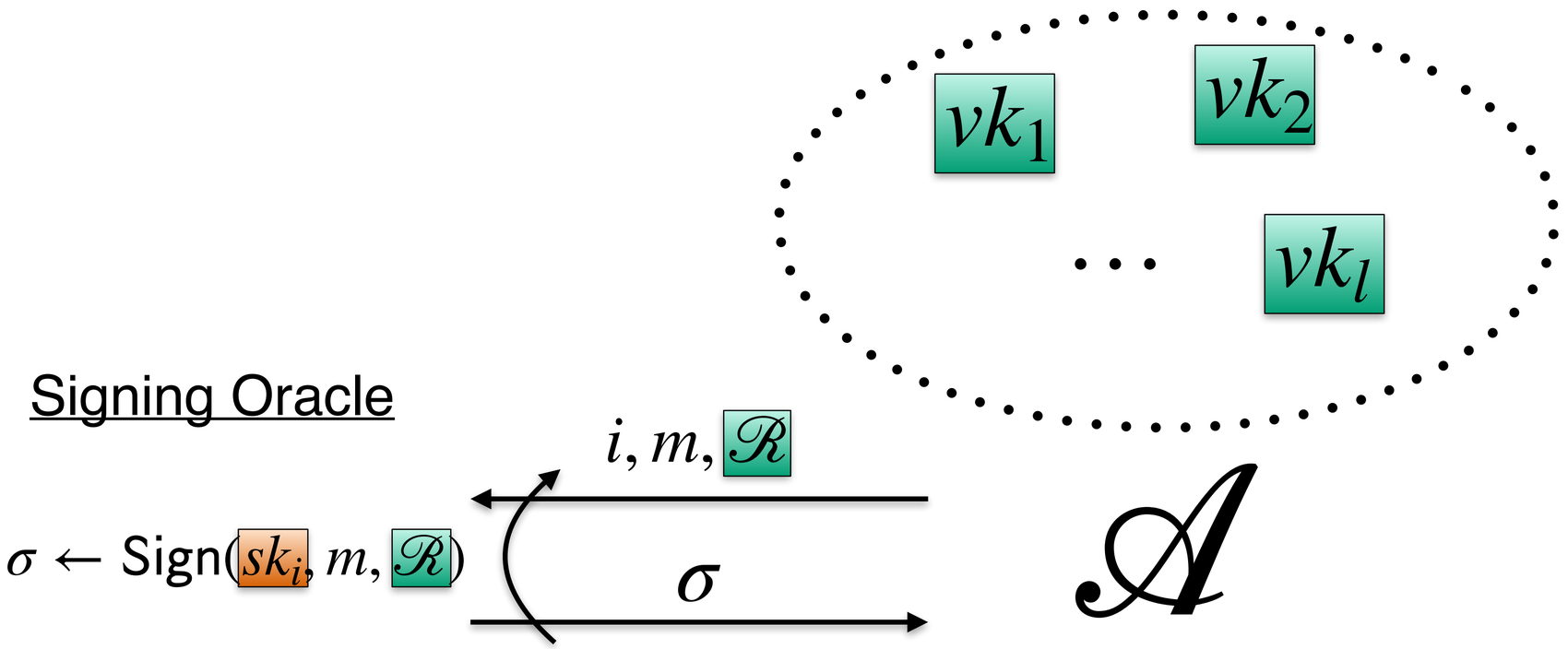
only honest keys in $\mathcal{R}^*$

$(m^*, \mathcal{R}^*)$ never queried

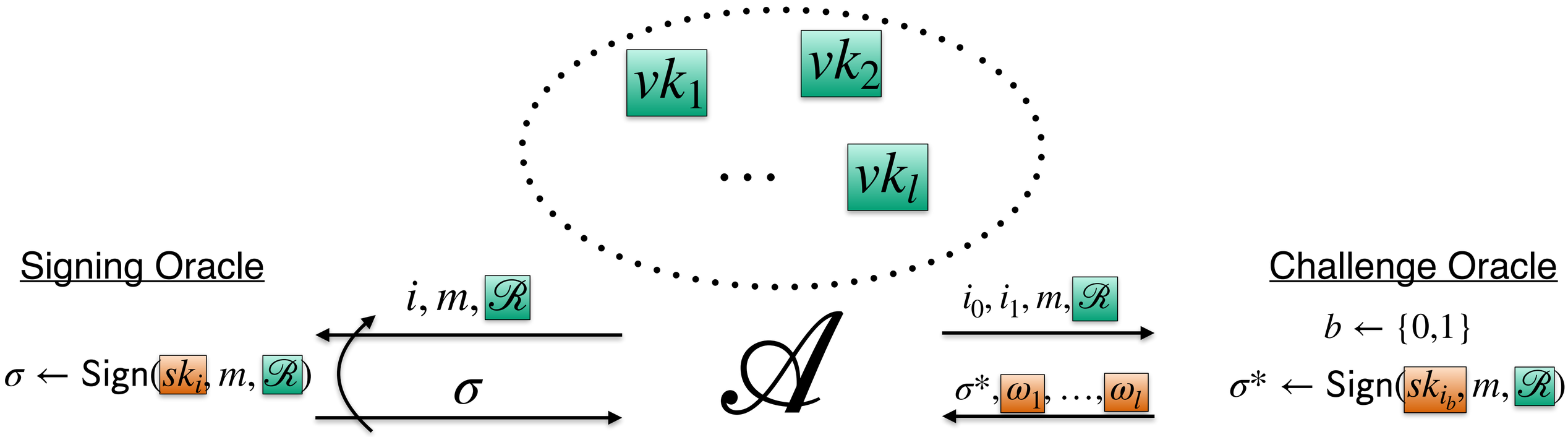
Ring Signatures: Anonymity [Bender-Katz-Morselli, 2006]



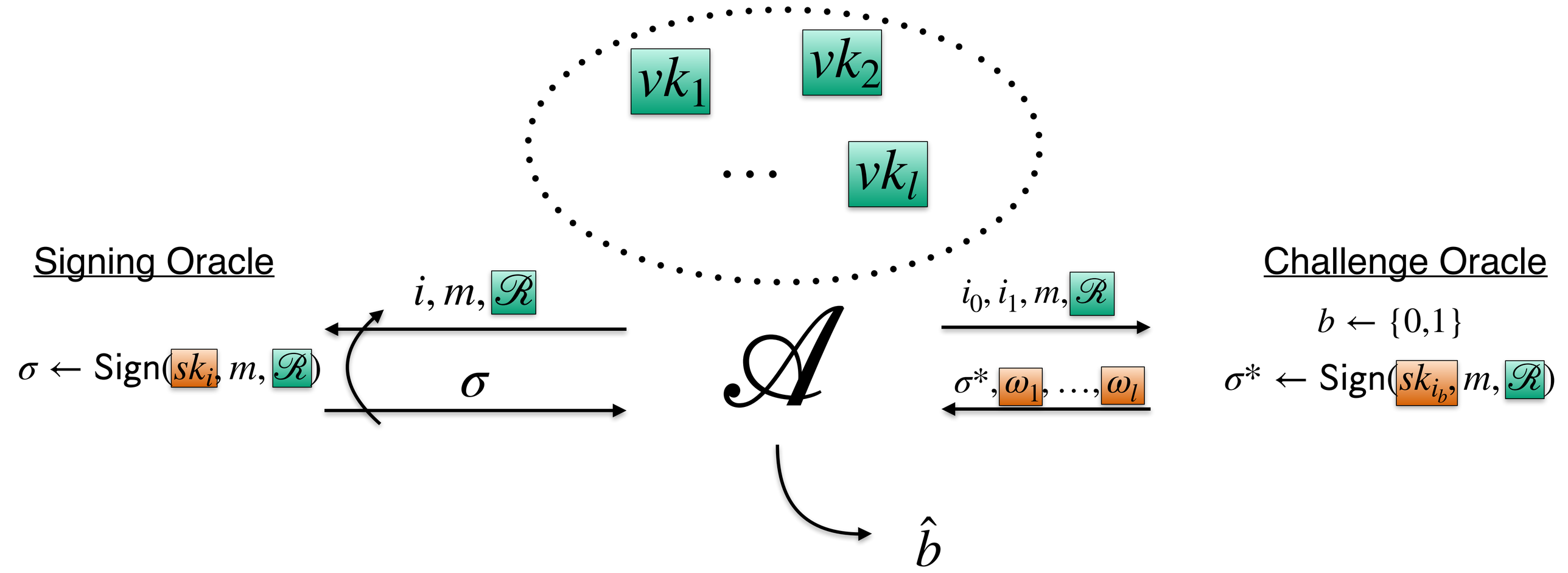
Ring Signatures: Anonymity [Bender-Katz-Morselli, 2006]



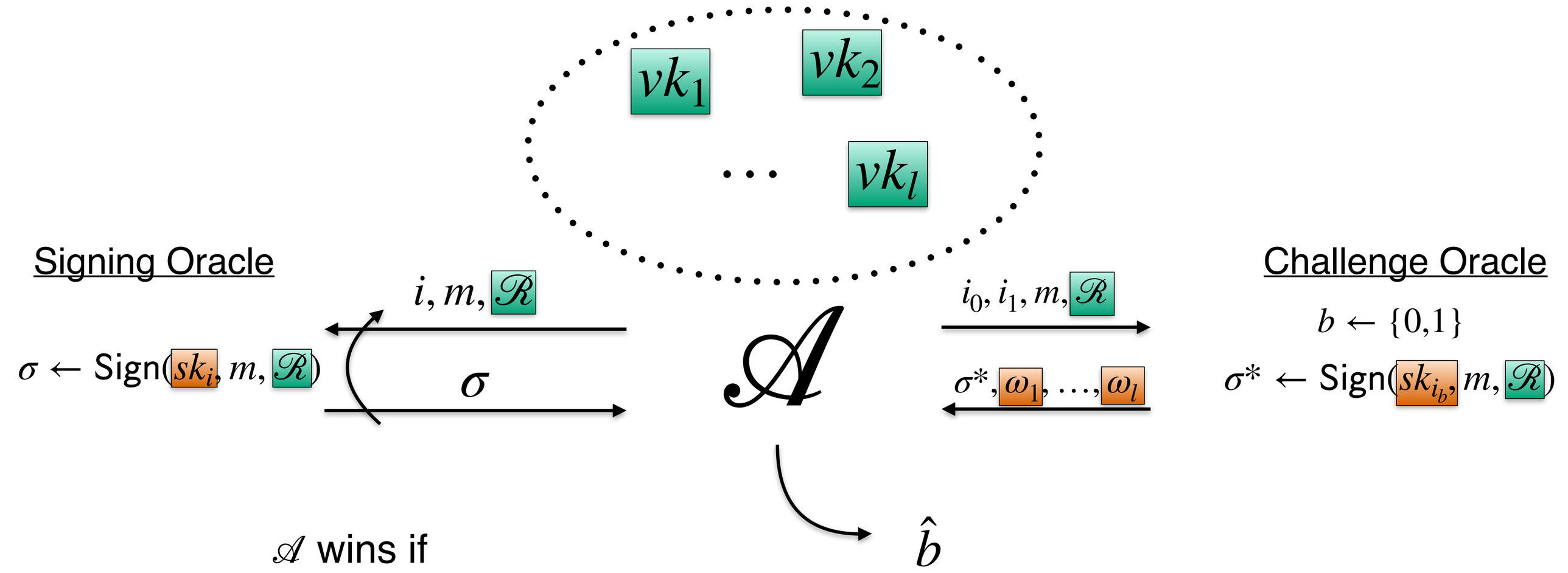
Ring Signatures: Anonymity [Bender-Katz-Morselli, 2006]



Ring Signatures: Anonymity [Bender-Katz-Morselli, 2006]



Ring Signatures: Anonymity [Bender-Katz-Morselli, 2006]



$\mathcal{A}$ wins if

$$b = \hat{b}$$

Ring Signatures: Desiderata

- Standard Model security, falsifiable assumptions, no ROM
- Small signatures for efficiency
- No form of trusted setup!

Ring Signatures: Generic Approach

Sign the Message

Prove Ring Membership

Ring Signatures: Generic Approach

Ring Signatures: [Malavolta-Schröder, 2017]

CISPA HELMHOLTZ-ZENTRUM i.G.

UNIVERSITÄT DES SAARLANDES

Ring Signatures: [Malavolta-Schröder, 2017]

$$sk' \leftarrow \Sigma.\mathsf{RerandSK}(sk, r)$$

$$\sigma_m \leftarrow \Sigma.\mathsf{Sign}(sk', m \,||\, \mathscr{R})$$

Σ: Signature Scheme with Rerandomizable Keys
[Fleischhacker-Krupp-Malavolta-Simkin-S-Schröder, 2016]

Ring Signatures: [Malavolta-Schröder, 2017]

$$sk' \leftarrow \Sigma.\mathsf{RerandSK}(sk, r)$$

$$\sigma_m \leftarrow \Sigma.\mathsf{Sign}(sk', m \,||\, \mathscr{R})$$

$$\pi \leftarrow \Pi.\mathsf{Prove}\left(\begin{array}{c} vk' = \Sigma.\mathsf{RerandVK}(vk, r) \\ vk \text{ in } \mathscr{R} \end{array}\right)$$

$$\sigma = (vk', \sigma_m, \pi)$$

Σ: Signature Scheme with Rerandomizable Keys
[Fleischhacker-Krupp-Malavolta-Simkin-S-Schröder, 2016]

Π: tailored NIZK-PoK with shared setup

Ring Signatures: [Malavolta-Schröder, 2017]

$$sk' \leftarrow \Sigma.\mathsf{RerandSK}(sk, r)$$

$$\sigma_m \leftarrow \Sigma.\mathsf{Sign}(sk', m \,||\, \mathscr{R})$$

$$\pi \leftarrow \Pi.\mathsf{Prove}\left(\begin{array}{c} vk' = \Sigma.\mathsf{RerandVK}(vk, r) \\ vk \text{ in } \mathscr{R} \end{array}\right)$$

$$\sigma = (vk', \sigma_m, \pi)$$

Σ: Signature Scheme with Rerandomizable Keys
[Fleischhacker-Krupp-Malavolta-Simkin-S-Schröder, 2016]

Π: tailored NIZK-PoK with shared setup

- No Setup
- Security under q-Strong DH assumption + Linear-KEA (GGM)
- O(n) signature size

Signatures with Flexible Public Key (SFPK)

CISPA HELMHOLTZ-ZENTRUM i.G.

UNIVERSITÄT DES SAARLANDES

Signatures with Flexible Public Key (SFPK)

$$(sk, vk) \leftarrow \mathsf{KeyGen}(1^\lambda)$$

$$\sigma \leftarrow \mathsf{Sign}(sk, m)$$

$$\begin{matrix}\texttt{accept}\\ \texttt{reject}\end{matrix} \leftarrow \mathsf{Verify}(vk, \sigma, m)$$

Signatures with Flexible Public Key (SFPK)

$$(sk, vk) \leftarrow \mathsf{KeyGen}(1^\lambda)$$

$$\sigma \leftarrow \mathsf{Sign}(sk, m)$$

$$\begin{matrix}\texttt{accept}\\ \texttt{reject}\end{matrix} \leftarrow \mathsf{Verify}(vk, \sigma, m)$$

$$vk' \leftarrow \mathsf{ChangeRepVK}(vk, r)$$

$$sk' \leftarrow \mathsf{ChangeRepSK}(sk, r)$$

$$vk' \sim vk$$

Signatures with Flexible Public Key (SFPK)

$(sk, vk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\sigma \leftarrow \mathsf{Sign}(sk, m)$

$\texttt{accept}/\texttt{reject} \leftarrow \mathsf{Verify}(vk, \sigma, m)$

$(\tau, sk, vk) \leftarrow \mathsf{TKeyGen}(1^\lambda)$

$\texttt{yes}/\texttt{no} \leftarrow \mathsf{CheckRep}(\tau, vk, vk')$

$vk' \leftarrow \mathsf{ChangeRepVK}(vk, r)$

$sk' \leftarrow \mathsf{ChangeRepSK}(sk, r)$

$vk' \sim vk$

Signatures with Flexible Public Key (SFPK)

$(sk, vk) \leftarrow \mathsf{KeyGen}(1^\lambda)$

$\sigma \leftarrow \mathsf{Sign}(sk, m)$

$\texttt{accept}/\texttt{reject} \leftarrow \mathsf{Verify}(vk, \sigma, m)$

$(\tau, sk, vk) \leftarrow \mathsf{TKeyGen}(1^\lambda)$

$\texttt{yes}/\texttt{no} \leftarrow \mathsf{CheckRep}(\tau, vk, vk')$

$vk' \leftarrow \mathsf{ChangeRepVK}(vk, r)$

$sk' \leftarrow \mathsf{ChangeRepSK}(sk, r)$

$vk' \sim vk$

Example

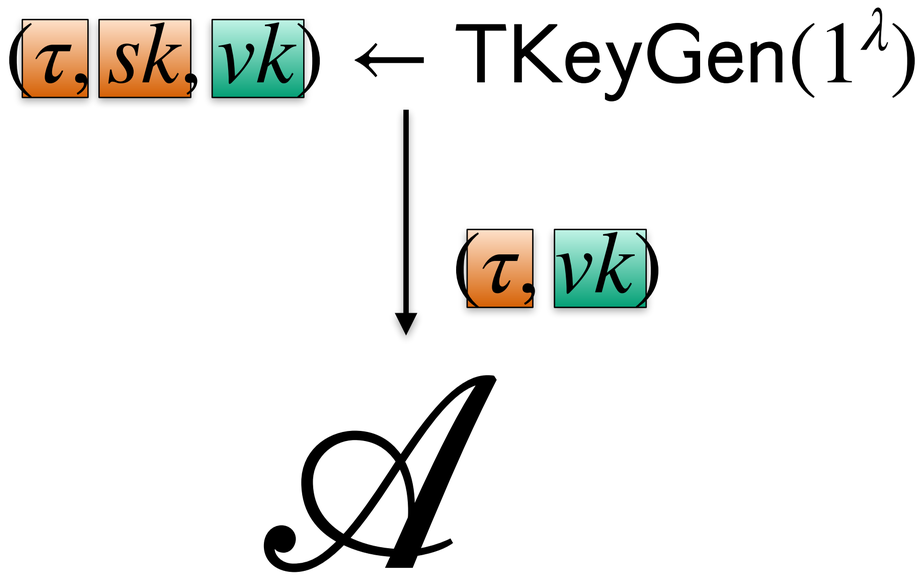
$vk, vk' \in \mathbb{G}^\ell$

$vk \sim vk'$ if there is $r \in \mathbb{Z}_p^*$

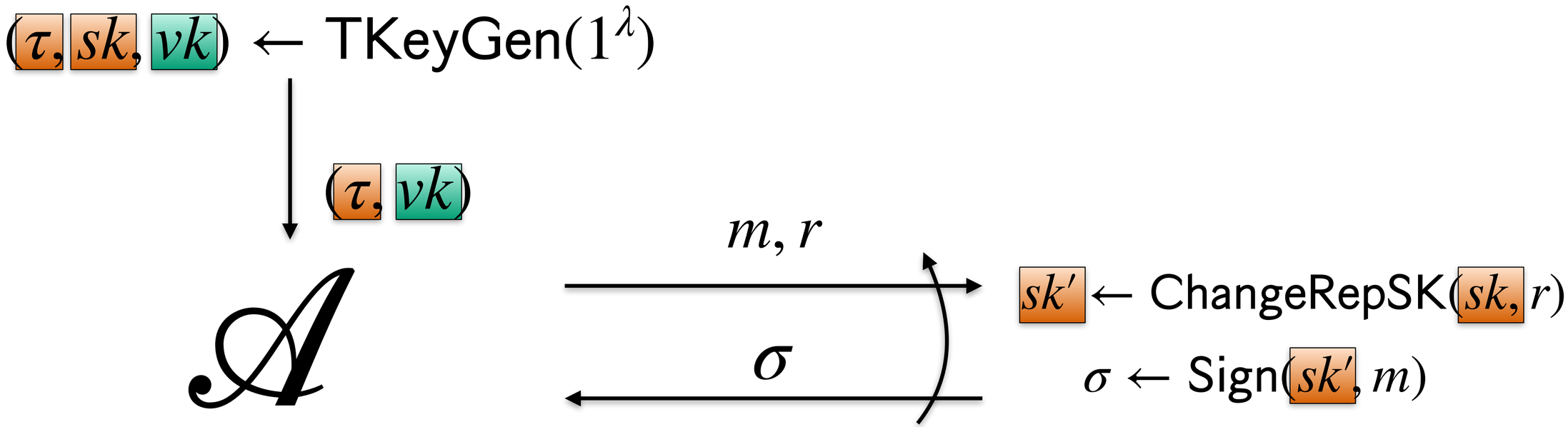
$(vk_1^r, \ldots vk_\ell^r) = (vk'_1, \ldots vk'_\ell)$

SFPK: Unforgeability

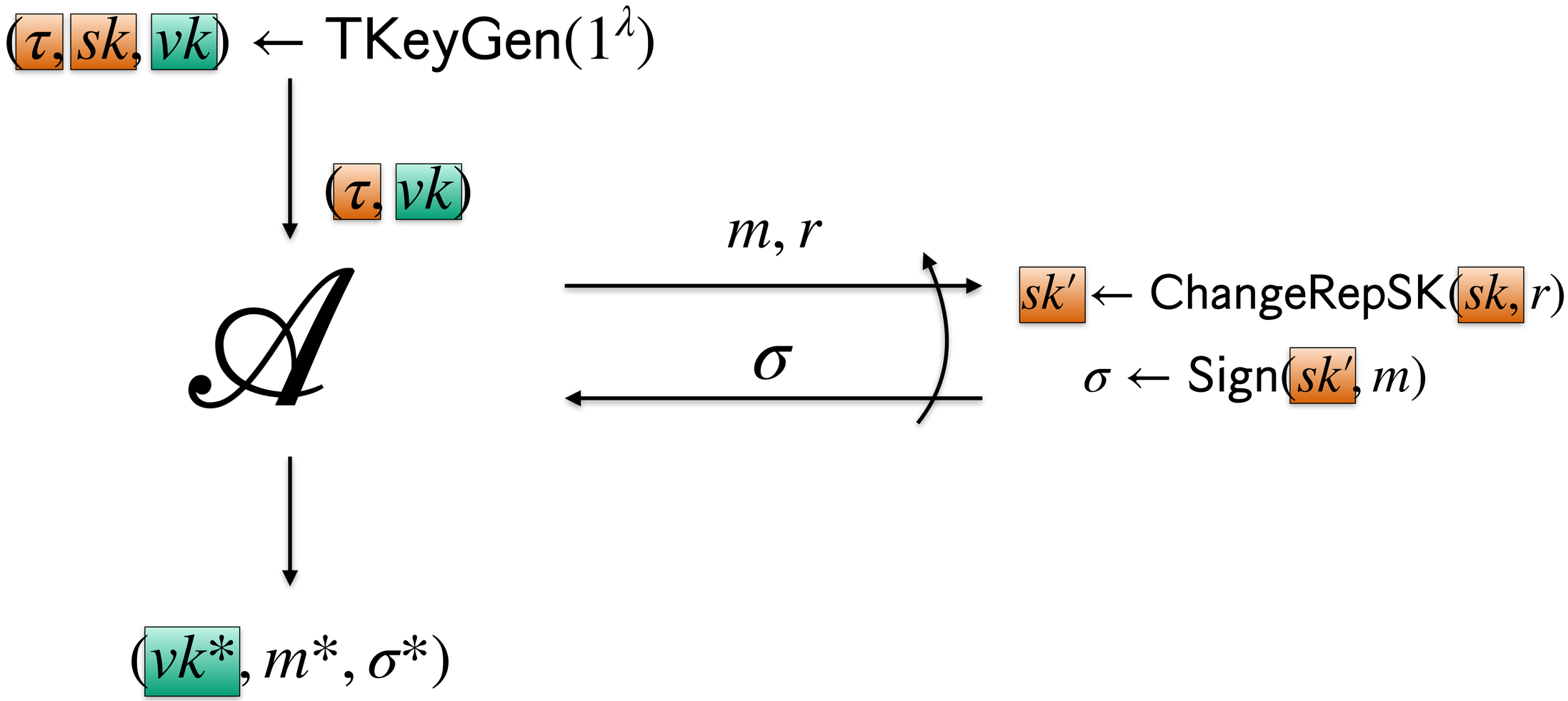
SFPK: Unforgeability



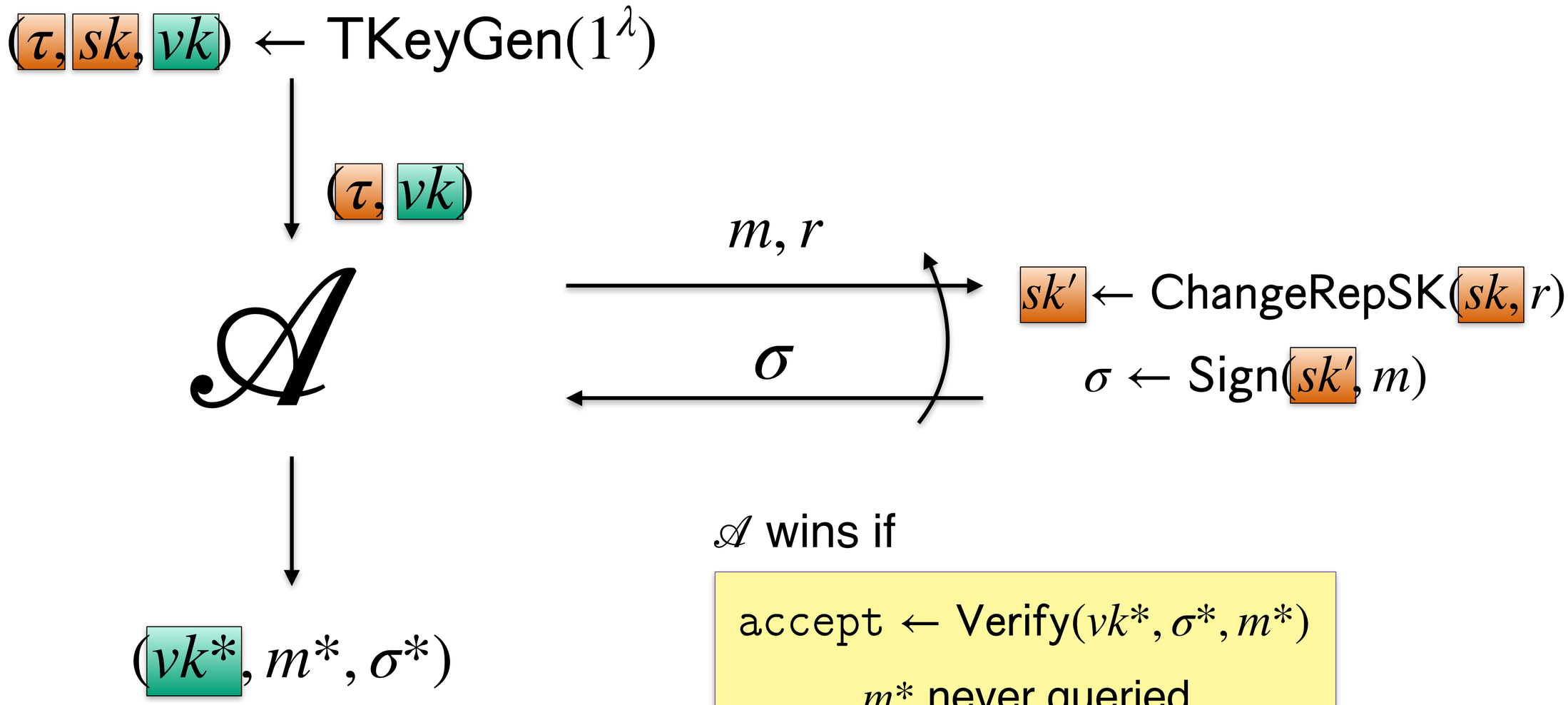
SFPK: Unforgeability



SFPK: Unforgeability



SFPK: Unforgeability



$\mathcal{A}$ wins if

$\mathtt{accept} \leftarrow \mathsf{Verify}(vk^*, \sigma^*, m^*)$

m^* never queried

$\mathtt{yes} \leftarrow \mathsf{CheckRep}(\tau, \mathsf{vk}, \mathsf{vk}^*)$

SFPK: Class-Hiding

SFPK: Class-Hiding

$$(sk_0, vk_0) \leftarrow \mathsf{KeyGen}(1^\lambda; \omega_0)$$
$$(sk_1, vk_1) \leftarrow \mathsf{KeyGen}(1^\lambda; \omega_1)$$

SFPK: Class-Hiding

$(sk_0, vk_0) \leftarrow \mathsf{KeyGen}(1^\lambda; \omega_0)$

$(sk_1, vk_1) \leftarrow \mathsf{KeyGen}(1^\lambda; \omega_1)$

$b \leftarrow \{0,1\}$

$\mathscr{A}$

SFPK: Class-Hiding

$(sk_0, vk_0) \leftarrow \text{KeyGen}(1^\lambda; \omega_0)$

$(sk_1, vk_1) \leftarrow \text{KeyGen}(1^\lambda; \omega_1)$

$b \leftarrow \{0,1\}$

$vk' \leftarrow \text{ChangeRepPK}(vk_b, r)$

$sk' \leftarrow \text{ChangeRepSK}(sk_b, r)$

$r \leftarrow R$

$\mathcal{A}$

SFPK: Class-Hiding

$(sk_0, vk_0) \leftarrow \text{KeyGen}(1^\lambda; \omega_0)$

$(sk_1, vk_1) \leftarrow \text{KeyGen}(1^\lambda; \omega_1)$

$b \leftarrow \{0,1\}$

$vk' \leftarrow \text{ChangeRepPK}(vk_b, r)$

$sk' \leftarrow \text{ChangeRepSK}(sk_b, r)$

$r \leftarrow R$

$(vk', \omega_0, \omega_1)$

$\mathcal{A}$

SFPK: Class-Hiding

$(sk_0, vk_0) \leftarrow \text{KeyGen}(1^\lambda; \omega_0)$

$(sk_1, vk_1) \leftarrow \text{KeyGen}(1^\lambda; \omega_1)$

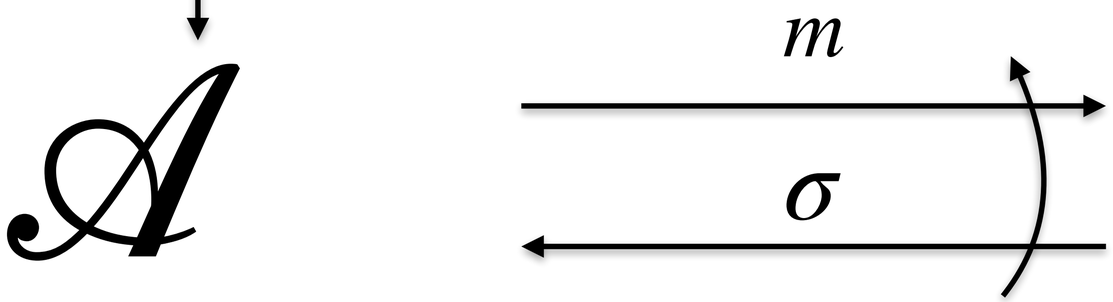
$b \leftarrow \{0,1\}$

$vk' \leftarrow \text{ChangeRepPK}(vk_b, r)$

$sk' \leftarrow \text{ChangeRepSK}(sk_b, r)$

$r \leftarrow R$

$(vk', \omega_0, \omega_1)$



$\sigma \leftarrow \text{Sign}(sk', m)$

SFPK: Class-Hiding

$(sk_0, vk_0) \leftarrow \mathsf{KeyGen}(1^\lambda; \omega_0)$

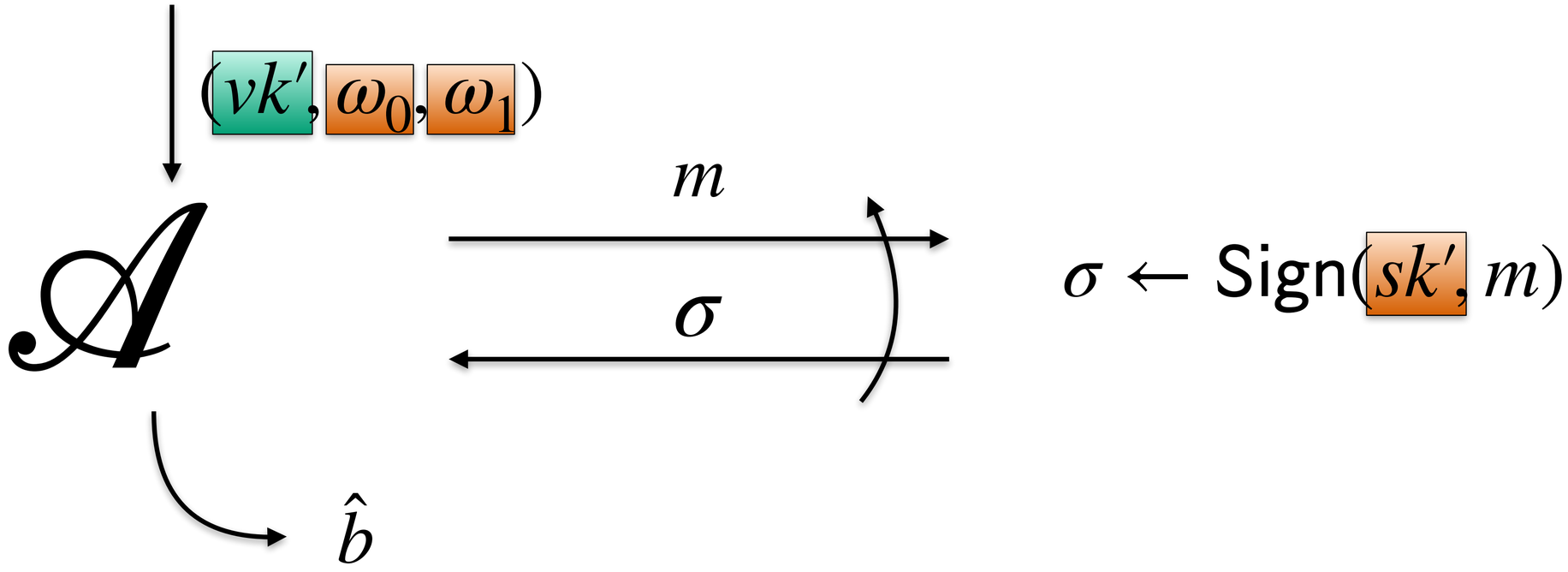
$(sk_1, vk_1) \leftarrow \mathsf{KeyGen}(1^\lambda; \omega_1)$

$b \leftarrow \{0,1\}$

$vk' \leftarrow \mathsf{ChangeRepPK}(vk_b, r)$

$sk' \leftarrow \mathsf{ChangeRepSK}(sk_b, r)$

$r \leftarrow R$



$\sigma \leftarrow \mathsf{Sign}(sk', m)$

SFPK: Class-Hiding

$(sk_0, vk_0) \leftarrow \text{KeyGen}(1^\lambda; \omega_0)$

$(sk_1, vk_1) \leftarrow \text{KeyGen}(1^\lambda; \omega_1)$

$b \leftarrow \{0,1\}$

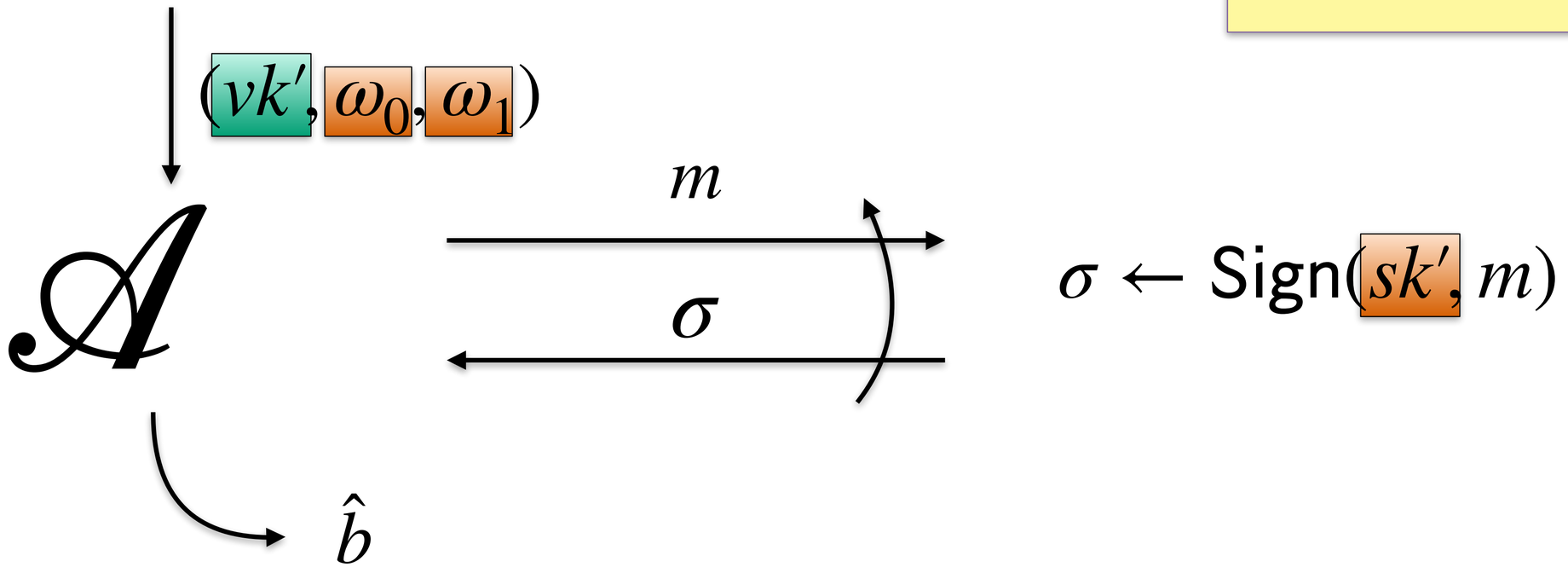
$vk' \leftarrow \text{ChangeRepPK}(vk_b, r)$

$sk' \leftarrow \text{ChangeRepSK}(sk_b, r)$

$r \leftarrow R$

$\mathcal{A}$ wins if

$$b = \hat{b}$$



$\sigma \leftarrow \text{Sign}(sk', m)$

Instantiation

$$sk = (y, X) \in \mathbb{Z}_p \times \mathbb{G}_1 \qquad \in \quad \mathbb{Z}_p \times \mathbb{G}_1$$

$$vk = (A, B, C, D, t := e(X^y, g_2), K_{PHF}) \quad \in \quad \mathbb{G}_1^4 \times \mathbb{G}_T \times \mathbb{G}_1^{\lambda+1}$$

Sign

$$r \leftarrow \mathbb{Z}_p^*$$

$$\sigma_1 := X^y \cdot H_{K_{PHF}}(m)^r$$

$$\sigma_2 := g_1^r \quad \sigma_3 := g_2^r$$

Verify

$$e(\sigma_2, g_2) =^? e(g_1, \sigma_3)$$

$$e(\sigma_1, g_2) =^? e(t, g_2) \cdot e(H_{K_{PHF}}(m), \sigma_3)$$

Instantiation

$$sk = (y, X) \in \mathbb{Z}_p \times \mathbb{G}_1 \qquad \in \quad \mathbb{Z}_p \times \mathbb{G}_1$$

$$vk = (A, B, C, D, t := e(X^y, g_2), K_{PHF}) \quad \in \quad \mathbb{G}_1^4 \times \mathbb{G}_T \times \mathbb{G}_1^{\lambda+1}$$

Sign

$$r \leftarrow \mathbb{Z}_p^*$$

$$\sigma_1 := X^y \cdot H_{K_{PHF}}(m)^r$$

$$\sigma_2 := g_1^r \quad \sigma_3 := g_2^r$$

Verify

$$e(\sigma_2, g_2) =^? e(g_1, \sigma_3)$$

$$e(\sigma_1, g_2) =^? e(t, g_2) \cdot e(H_{K_{PHF}}(m), \sigma_3)$$

- Unforgeable under DLIN assumption and security of PHF (e.g. CDH)
- Class Hiding under DDH assumption in $\mathbb{G}_1$

Ring Signatures: [Malavolta-Schröder, 2017]

$$sk' \leftarrow \Sigma.\mathsf{RerandSK}(sk, r)$$

$$\sigma_m \leftarrow \Sigma.\mathsf{Sign}(sk', m \,||\, \mathscr{R})$$

$$\pi \leftarrow \Pi.\mathsf{Prove}\left(\begin{array}{c} vk' = \Sigma.\mathsf{RerandVK}(vk, r) \\ vk \text{ in } \mathscr{R} \end{array}\right)$$

$$\sigma = (vk', \sigma_m, \pi)$$

Σ: Signature Scheme with Rerandomizable Keys
[Fleischhacker-Krupp-Malavolta-Simkin-S-Schröder, 2016]

Π: tailored NIZK-PoK with shared setup

- No Setup
- Security under q-Strong DH assumption + Linear-KEA (GGM)
- O(n) signature size

Ring Signatures: Our Construction

Ring Signatures: Our Construction

$$sk' \leftarrow \Sigma.\mathsf{ChangeRepSK}(sk, r)$$

$$\sigma_m \leftarrow \Sigma.\mathsf{Sign}(sk', m \,||\, \mathscr{R})$$

Σ: Signature Scheme with Flexible Public Keys

Ring Signatures: Our Construction

$$sk' \leftarrow \Sigma.\mathsf{ChangeRepSK}(sk, r)$$

$$\sigma_m \leftarrow \Sigma.\mathsf{Sign}(sk', m \,||\, \mathscr{R})$$

$$\pi \leftarrow \Pi.\mathsf{Prove}\left(\begin{matrix} vk' = \Sigma.\mathsf{ChangeRepVK}(vk, r) \\ vk \text{ in } \mathscr{R} \end{matrix}\right)$$

$$\sigma = (vk', \sigma_m, \pi)$$

Σ: Signature Scheme with Flexible Public Keys

Π: perfectly sound NIWI [Groth-Ostrovsky-Sahai, 2006]

Ring Signatures: Our Construction

$sk' \leftarrow \Sigma.\mathsf{ChangeRepSK}(sk, r)$

$\sigma_m \leftarrow \Sigma.\mathsf{Sign}(sk', m \,||\, \mathcal{R})$

$\pi \leftarrow \Pi.\mathsf{Prove}\left(vk' = \Sigma.\mathsf{ChangeRepVK}(vk, r) \text{ in } \mathcal{R}\right)$

$$\sigma = (vk', \sigma_m, \pi)$$

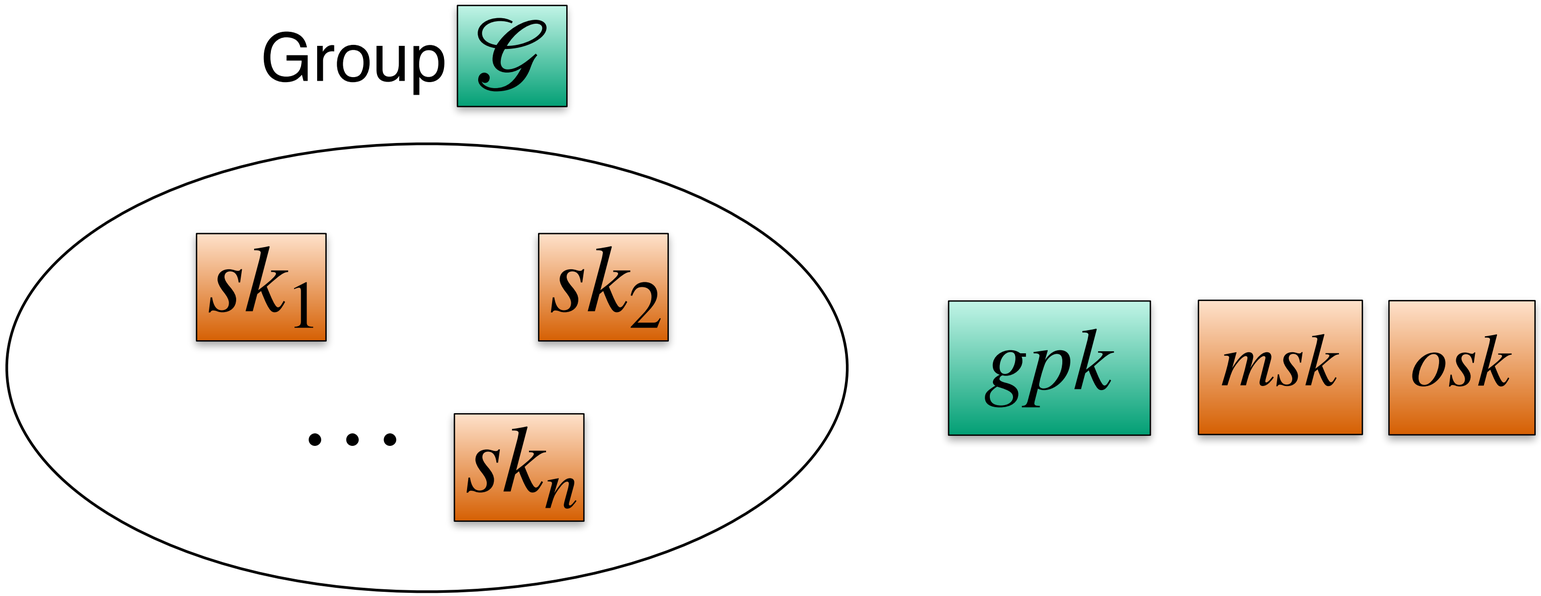
Σ: Signature Scheme with Flexible Public Keys

Π: perfectly sound NIWI [Groth-Ostrovsky-Sahai, 2006]

- No need to extract randomizer from SFPK in unforgeability proof!
- Perfectly sound NIWI without Setup + square root technique [Chandran-Groth-Sahai, 2007]

  ➡ **O(√n) size Ring Signatures without Setup from Standard assumptions!**

Group Signatures [Bellare-Micciancio-Warinschi, 2003]



$$\sigma \leftarrow \mathsf{Sign}(gpk, sk_i, m)$$

$$\texttt{accept/reject} \leftarrow \mathsf{Verify}(gpk, \sigma, m)$$

$$i \leftarrow \mathsf{Open}(osk, \sigma, m)$$

Anonymity

- Signature hides signer identity

Traceability

- Opening cannot be evaded or subverted

Group Signatures: Our construction

Setup :

Sign :

Σ_{flex}: SFPK

Σ_{EQ}: SPS-EQ

[Fuchsbauer-Gay, 2018]

}

$vk, vk' \in \mathbb{G}^{\ell}$

$vk \sim vk'$ if there is $r \in \mathbb{Z}_p^*$

$(vk_1^r, \ldots vk_{\ell}^r) = (vk'_1, \ldots vk'_{\ell})$

Group Signatures: Our construction

Setup :

$$(msk, gpk[0]) \leftarrow \Sigma_{SPS}.\mathsf{KeyGen}(1^\lambda)$$

$$\text{For } i \in [n] : (osk[i], sk_i, vk_i) \leftarrow \Sigma_{flex}.\mathsf{TKeyGen}(1^\lambda)$$

$$(gpk[1], \ldots, gpk[n]) \leftarrow \left(\Sigma_{SPS}.\mathsf{Sign}(msk, vk_1), \ldots, \Sigma_{SPS}.\mathsf{Sign}(msk, vk_n)\right)$$

Sign :

Σ_{flex}: SFPK

Σ_{EQ}: SPS-EQ

[Fuchsbauer-Gay, 2018]

}

$$vk, vk' \in \mathbb{G}^\ell$$

$vk \sim vk'$ if there is $r \in \mathbb{Z}_p^*$

$$(vk_1^r, \ldots vk_\ell^r) = (vk'_1, \ldots vk'_\ell)$$

Group Signatures: Our construction

CISPA HELMHOLTZ-ZENTRUM i.G.

Setup :

$$(msk, gpk[0]) \leftarrow \Sigma_{SPS}.\mathsf{KeyGen}(1^\lambda)$$

$$\text{For } i \in [n] : (osk[i], sk_i, vk_i) \leftarrow \Sigma_{flex}.\mathsf{TKeyGen}(1^\lambda)$$

$$(gpk[1], \ldots, gpk[n]) \leftarrow \left(\Sigma_{SPS}.\mathsf{Sign}(msk, vk_1), \ldots, \Sigma_{SPS}.\mathsf{Sign}(msk, vk_n)\right)$$

Sign :

$$sk' \leftarrow \Sigma_{flex}.\mathsf{ChangeRepSK}(sk, r)$$

$$\sigma_m \leftarrow \Sigma_{flex}.\mathsf{Sign}(sk', m)$$

$$\sigma'_{vk} \leftarrow \Sigma_{\mathsf{EQ}}.\mathsf{ChangeRep}(gpk[0], vk', \sigma_{vk})$$

$$\sigma = (vk', \sigma_m, \sigma_{vk'})$$

Σ_{flex}: SFPK

Σ_{EQ}: SPS-EQ

[Fuchsbauer-Gay, 2018]

$$vk, vk' \in \mathbb{G}^\ell$$

$vk \sim vk'$ if there is $r \in \mathbb{Z}_p^*$

$$(vk_1^r, \ldots vk_\ell^r) = (vk'_1, \ldots vk'_\ell)$$

UNIVERSITÄT DES SAARLANDES

Group Signatures: Our construction

CISPA HELMHOLTZ-ZENTRUM i.G.

Setup :

$$(msk, gpk[0]) \leftarrow \Sigma_{SPS}.\mathsf{KeyGen}(1^\lambda)$$

$$\text{For } i \in [n] : (osk[i], sk_i, vk_i) \leftarrow \Sigma_{flex}.\mathsf{TKeyGen}(1^\lambda)$$

$$(gpk[1], \ldots, gpk[n]) \leftarrow \left(\Sigma_{SPS}.\mathsf{Sign}(msk, vk_1), \ldots, \Sigma_{SPS}.\mathsf{Sign}(msk, vk_n)\right)$$

Sign :

$$sk' \leftarrow \Sigma_{flex}.\mathsf{ChangeRepSK}(sk, r)$$

$$\sigma_m \leftarrow \Sigma_{flex}.\mathsf{Sign}(sk', m)$$

$$\sigma'_{vk} \leftarrow \Sigma_{\mathsf{EQ}}.\mathsf{ChangeRep}(gpk[0], vk', \sigma_{vk})$$

$$\sigma = (vk', \sigma_m, \sigma_{vk'})$$

Σ_{flex}: SFPK

Σ_{EQ}: SPS-EQ

[Fuchsbauer-Gay, 2018]

$vk, vk' \in \mathbb{G}^\ell$

$vk \sim vk'$ if there is $r \in \mathbb{Z}_p^*$

$(vk_1^r, \ldots vk_\ell^r) = (vk'_1, \ldots vk'_\ell)$

- DDH + co-Flex-DH $\Rightarrow 20\mathbb{G}_1 + 5\mathbb{G}_2$
- GGM + co-Flex-DH $\Rightarrow 9\mathbb{G}_1 + 2\mathbb{G}_2$

  [Fuchsbauer-Hanser-Slamanig, 2014]

UNIVERSITÄT DES SAARLANDES

Summary

Signatures with flexible public keys:

- ➡ short ring signatures from weak assumptions without setup
- ➡ short group signatures because of compatibility with SPS-EQ
- ➡ generic privacy preserving building block

Thanks!

ia.cr/2018/191

jonas.schneider@cispa-helmholtz.de