Constrained Verifiable Random Functions from Indistinguishability Obfuscation

Bei Liang*, *Hongda Li, Jinyong Chang

State Key Laboratory of Information Security,
Institute of Information Engineering,
Chinese Academy of Sciences.
liangbei@iie.ac.cn

Constrained PRFs

- Constrained PRFs [BW13]
  - Functions $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$

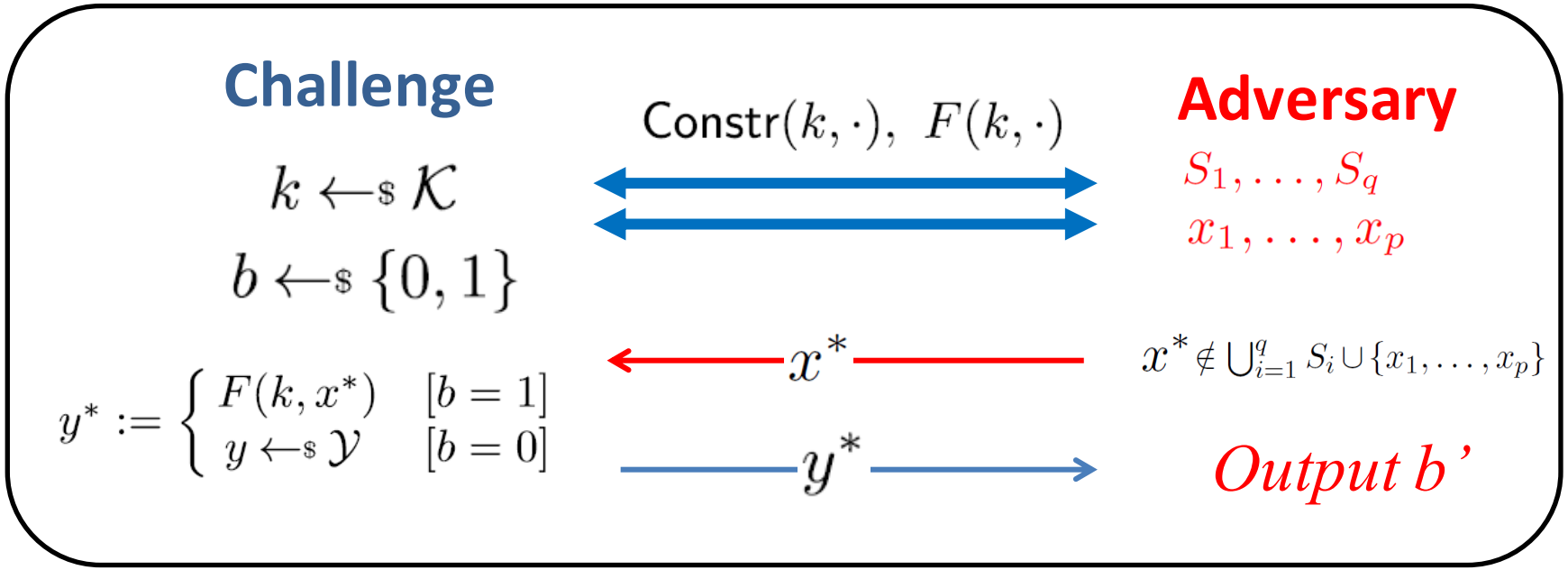
    with respect to a set systen $\mathcal{S} \subseteq 2^{\mathcal{X}}$
  - Algorithms:
    - $\mathsf{Constrain}(K, S \in \mathcal{S}) \to K_S$
    - $\mathsf{Eval}(K_S, x \in S) \to y$

$$K_S \leftarrow \mathsf{Constrain}(K, S)$$

$$\Rightarrow \mathsf{Eval}(K_S, x) = \begin{cases} F(K, x), & \text{if } x \in S; \\ \perp, & \text{otherwise.} \end{cases}$$

Security of Constrained PRFs

- *Pseudorandomness* of constrained PRFs:
  - *Function should look random where:*
    - *we have not seen its value*
    - *we cannot evaluate it using a constrained key*



Verifiable Random Functions

- VRFs [MRV99]
  - Functions: $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$
  - Algorithms:
    - $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$
    - $\mathsf{Prove}(\mathsf{sk}, x) \to (y, \pi)$
    - $\mathsf{Verify}(\mathsf{pk}, x, y, \pi) \to 0/1$

$\mathsf{Prove}(\mathsf{sk}, x)$ algorithm contains: $y = F(\mathsf{sk}, x)$ and $\pi = P(\mathsf{sk}, x)$

π is to prove the function value is computed correctly.

Security of VRFs

- Provability (Correctness):

$$y = F(\mathsf{sk}, x) \text{ and } \mathsf{Verify}(\mathsf{pk}, x, y, \pi) = 1$$

- Uniqueness:

For (x, y_0, π_0) , (x, y_1, π_1) :

$$y_0 \neq y_1 \Rightarrow \begin{cases} \mathsf{Verify}(\mathsf{pk}, x, y_0, \pi_0) = 0 \\ \quad \vee \ \mathsf{Verify}(\mathsf{pk}, x, y_1, \pi_1) = 0 \end{cases}$$

- Pseudorandomness:
  - Adversary gets Prove oracle.
  - submits x^* that has not been queried
  - receives either $F(\mathsf{sk}, x^*)$ or $y \leftarrow\!\!\$\ \mathcal{Y}$.

Constrained VRFs

- Constrained VRFs [Fuc14]:
  - Functions $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$

    with respect to a set system $\mathcal{S} \subseteq 2^{\mathcal{X}}$
  - Algorithms:
    - $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$
    - $\mathsf{Constr}(\mathsf{sk}, S) \to \mathsf{sk}_S$
    - $\mathsf{Prove}(\mathsf{sk}_S, x) \to (y, \pi)$
    - $\mathsf{Verify}(\mathsf{pk}, x, y, \pi) \to 0/1$
- Provability (correctness):

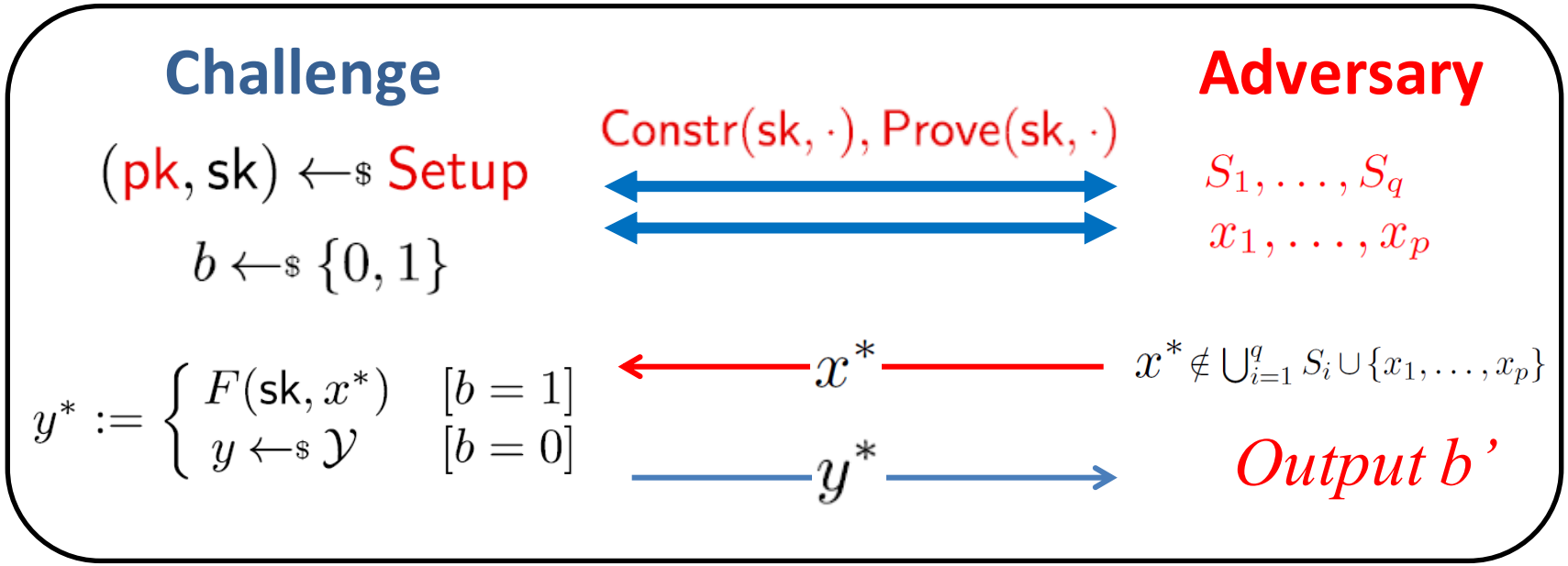
$$\left.\begin{array}{l} \mathsf{sk}_S \leftarrow \mathsf{Constr}(\mathsf{sk}, S) \\ (y, \pi) \leftarrow \mathsf{Prove}(\mathsf{sk}_S, x) \end{array}\right\} \Rightarrow \begin{cases} x \in S & \Rightarrow y = F(\mathsf{sk}, x) \text{ and } \mathsf{Verify}(\mathsf{pk}, x, y, \pi) = 1 \\ x \notin S & \Rightarrow (y, \pi) = (\bot, \bot) \end{cases}$$

Security of Constrained VRFs

- Uniqueness:
- Constraint-hiding:

$$\mathsf{Prove}(\mathsf{sk}, x) = \mathsf{Prove}(\mathsf{Constr}(\mathsf{sk}, S), x)$$

- Pseudorandomness:

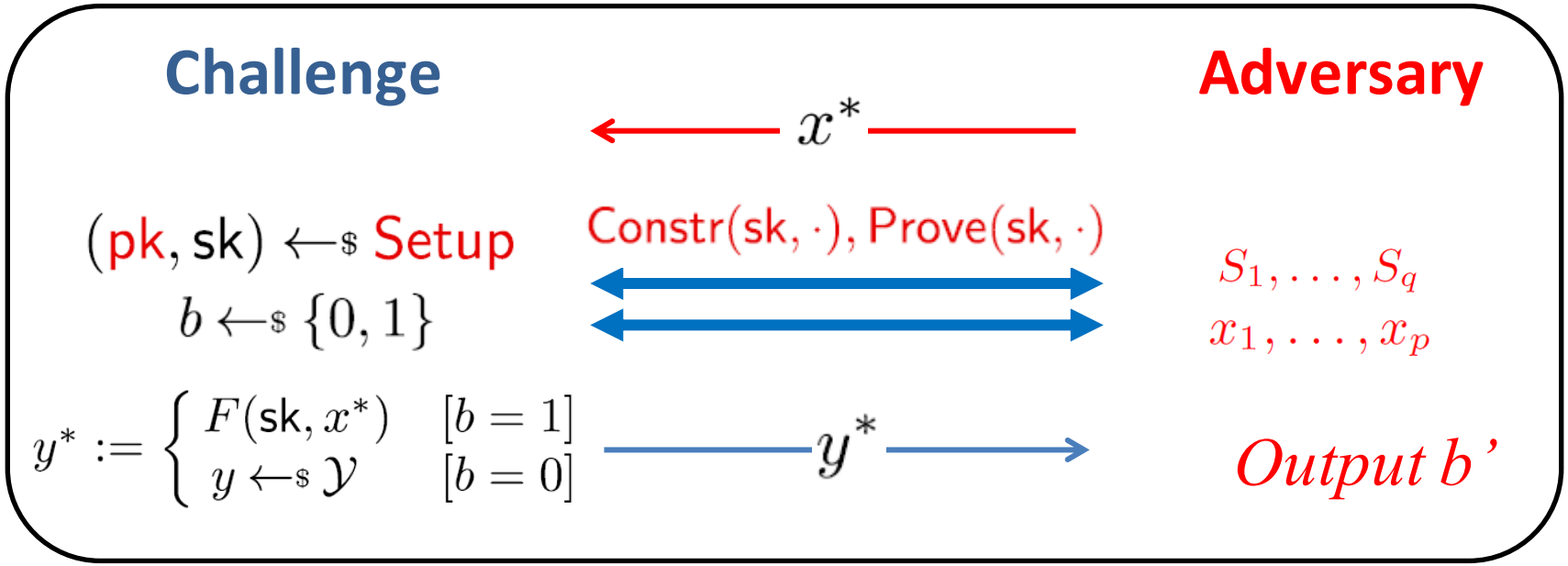


Security of Constrained VRFs

- Uniqueness:
- Constraint-hiding:

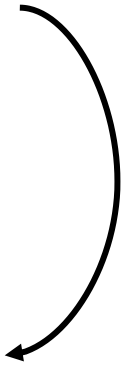
$$\mathsf{Prove}(\mathsf{sk}, x) = \mathsf{Prove}(\mathsf{Constr}(\mathsf{sk}, S), x)$$

- Pseudorandomness (*selective*):



Construction of VRFs

- Constrained PRFs [BW13]
  - bit-fixing, circuit-constr:
    - from multilinear Maps
- Constrained VRFs [Fuc14]
  - bit-fixing, circuit-constr:
    - from multilinear Maps



Construction of VRFs

- Constrained PRFs [BW13]
  - bit-fixing, circuit-constr:
    - from multilin. Maps
- Constrained VRFs [Fuc14]
  - bit-fixing, circuit-constr:
    - from multilin. Maps

- VRFs [HW10]
  - *under q-type assumptions*

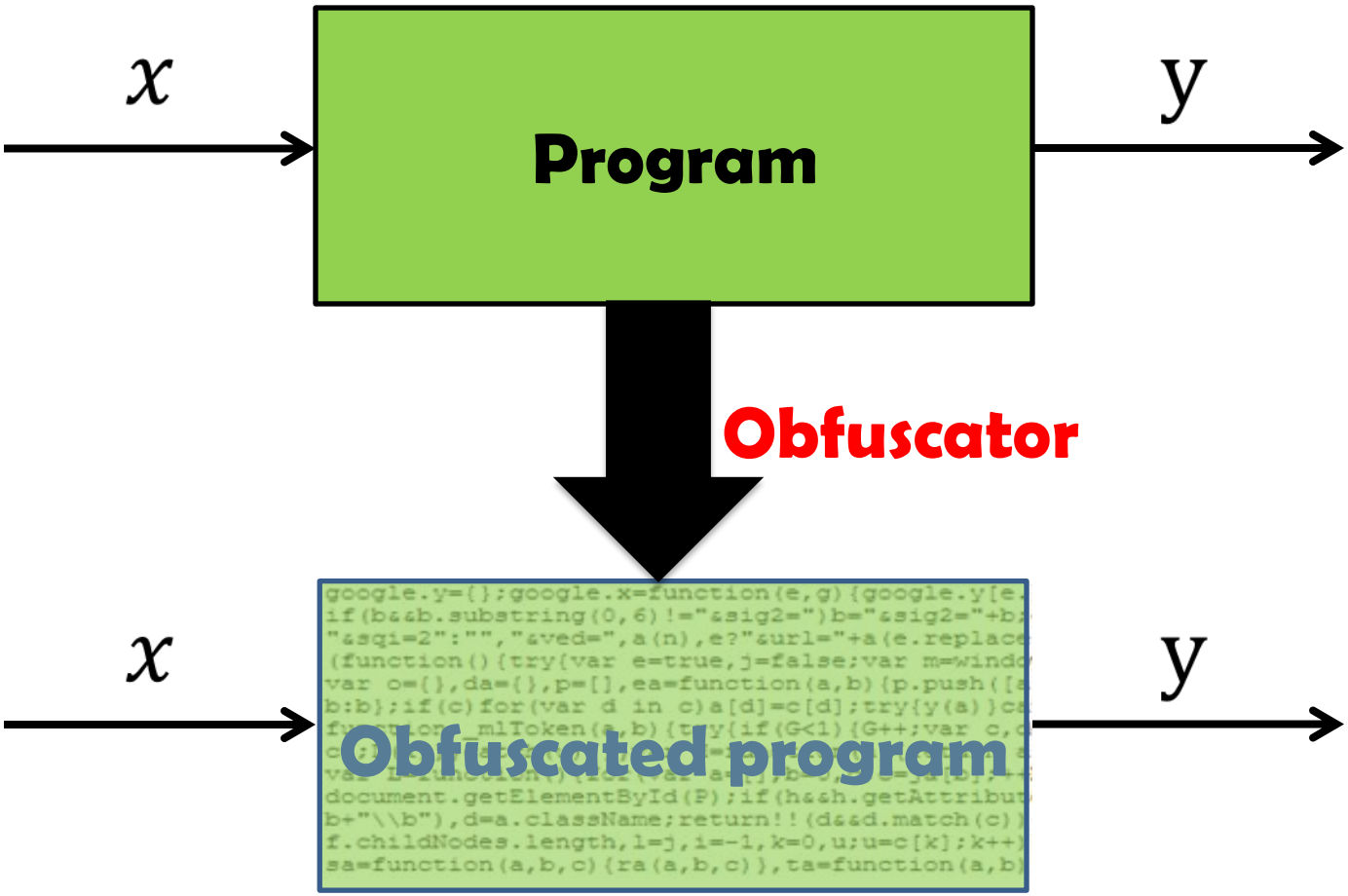
Negative results of VRFs

cannot be constructed in black-box way:

- *Based on one-way permutations* [BG09]
- *Based on trapdoor permutations* [FS12]

Is it possible to construct VRFs from one way functions, combing with other assumption (indistinguishability obfuscation)?

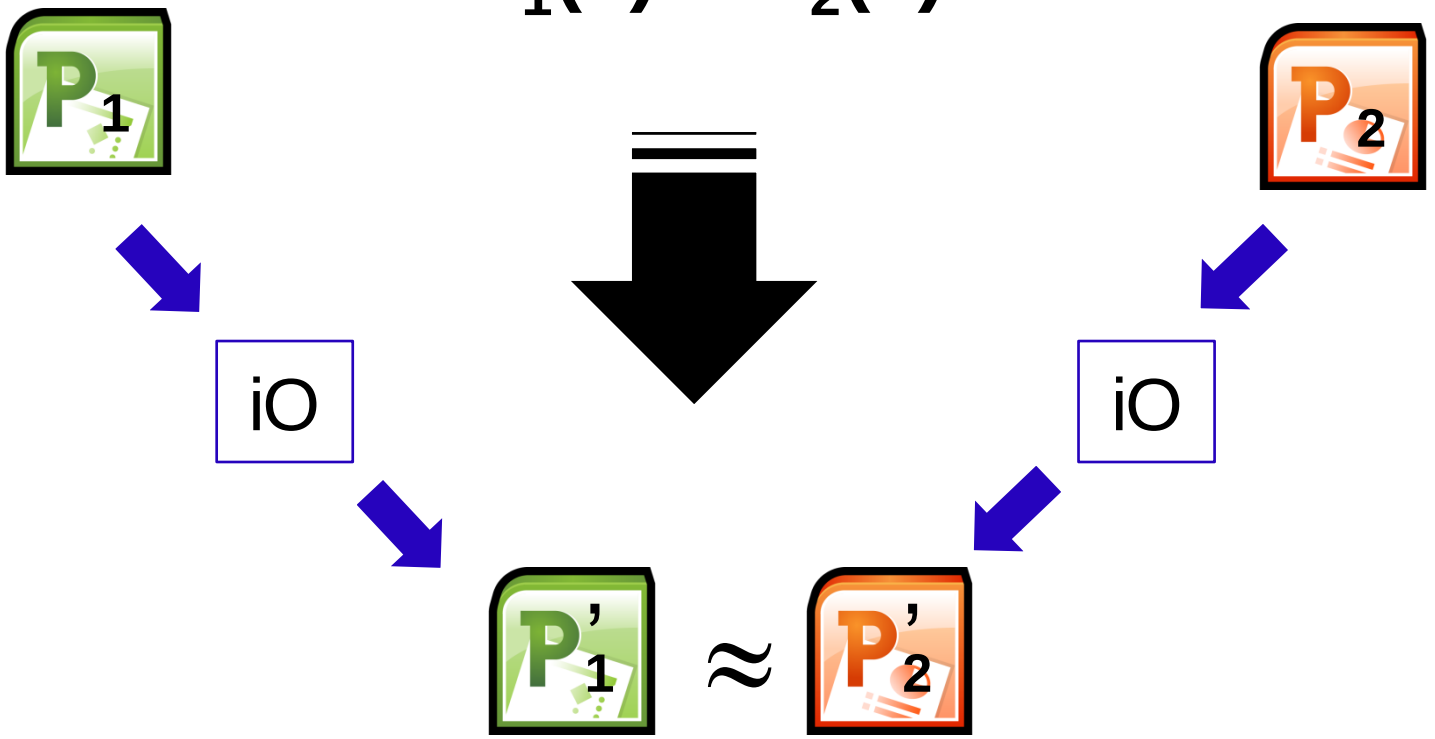
Program Obfuscation[BGI+01]



Indistinguishability Obfuscation[BGI+01]

If two programs have same functionality, obfuscations are indistinguishable.

$$P_1(x) = P_2(x) \quad \forall x$$



Punctured PRFs [BW13]

- Punctured PRF key $K\{x^*\}$:
  - $K\{x^*\}$ evaluate $\mathsf{PRF}(K, x)$ on all points, but x^*;
- $\mathsf{PRF}(K, x)$ define for all x;
- $K\{x^*\}$ can evaluate $\mathsf{PRF}(K, x)$ for $\forall x \neq x^*$;
- Security: given $K\{x^*\}$,

  cannot distinguish $\mathsf{PRF}(K, x^*)$ and random;

Special case of constrained PRFs [BW13]

Build from [GGM84]

Our Construction of CVRFs [this work]

- **Setup:** samples a PRF key K
  - the secret key $SK = K$
  - *Define* $F(K,x) = y = b$, $P(K,x) = \pi = r$
  - the public key *PK*= obfuscation of the program

$\mathsf{PRF}(K,x) = b\|r \in \{0,1\} \times \{0,1\}^{\ell}$

Constants: punctured PRF key K

Inputs: $x \in \{0,1\}^n$

Algorithm:

(1) compute $t = \mathsf{PRF}(K,x)$
and set $t = b\|r \in \{0,1\} \times \{0,1\}^{\ell(\lambda)}$

(2) output $c = \mathsf{Com}(b;r)$

$\mathbf{Prog_1}$: $\mathbf{Verify}_K$

Our Construction of CVRFs [this work]

- **Constrain(SK=K, S):** SK_S=obfuscated program

Constants: punctured PRF key K, set S

Inputs: $x \in \{0,1\}^n$

Algorithm:

(1) If $x \in S$,

- compute $\mathsf{PRF}(K, x) = b \| r \in \{0,1\} \times \{0,1\}^\ell$
- outputs $F(K, x) = y = b$ and $P(K, x) = \pi = r$

(2) Otherwise, output $(\perp, \perp)$

$\mathbf{Prog_2}$: $\mathsf{ConstrainedKey}_{SK,S}$

Our Construction of CVRFs [this work]

- **Prove(SK_S, x):** run $SK_S(x)$
  - *the functionality of* SK_S *is equal to* $Prog_2(x)$
  - **Provability:**

    for all x, $SK_S(x) = Prog_2(x)$
    - **if $x \in S$, $SK_S(x) = (b, r)$**
    - **If $x \notin S$, $SK_S(x) = (\perp, \perp)$**
- **Verify(PK, x, y, π):**
  - Run PK(x) and obtain c;
  - Check if c= Com(y; π);
  - Output 1 if true; else output 0.

The functionality of PK(x) is equal to $Prog_1(x)$

Proof of Security [this work]

- **Uniqueness:** perfectly binding property of Com;
  - if $y_0 \neq y_1 \wedge \mathsf{VRF.Verify}(PK, x, y_0, \pi_0) = 1 \wedge \mathsf{VRF.Verify}(PK, x, y_1, \pi_1) = 1$
  - That is $y_0 \neq y_1 \wedge \mathsf{Com}(y_0; \pi_0) = \mathsf{Com}(y_1; \pi_1)$
  - It contradicts with the perfectly binding property of Com.

Proof of Security [this work]

- **Selective pseudorandomness:**

Game $0'$

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$b^* \| r^* = \mathsf{PRF}(K, x^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_K])$

$y^* = b^*$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K,\cdot),\mathsf{C}(K,\cdot)}(y^*)$

output 1 iff $b' = 0$

Po(K, ·) Stands for Prove(K, ·)

C(K, ·) Stands for Constrain(K, ·)

Proof of Security [this work]

- **Selective pseudorandomness:** $i\mathcal{O}$

Game 1

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$K(x^*) \leftarrow \mathsf{Punctured}(K, x^*)$

$b^* \| r^* = \mathsf{PRF}(K, x^*)$

$c^* = \mathsf{Com}(b^*; r^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_{K(x^*),c^*}])$

$y^* = b^*$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K,\cdot),\mathsf{C}(K,\cdot)}(y^*)$

output 1 iff $b' = 0$

Constants: punctured key $K(x^*)$ and c^*

Inputs: $x \in \{0,1\}^n$

Algorithm:

(1) If $x = x^*$, outputs c^*.

(2) Else, do as follows:

- compute $t = \mathsf{PRF}(K, x)$ and set $t = b \| r \in \{0,1\} \times \{0,1\}^{\ell(\lambda)}$
- output $c = \mathsf{Com}(b; r)$

Prog'$_1$: $\mathsf{Verify}_{K(x^*),c^*}$

Proof of Security [this work]

- **Selective pseudorandomness:** *functionality preserved under puncturing*

Game 2

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$K(x^*) \leftarrow \mathsf{Punctured}(K, x^*)$

$b^* \| r^* = \mathsf{PRF}(K, x^*)$

$c^* = \mathsf{Com}(b^*; r^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_{K(x^*),c^*}])$

$y^* = b^*$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K(x^*),\cdot),\mathsf{C}(K,\cdot)}(y^*)$

output 1 iff $b' = 0$

For $x \neq x^*$, it always holds that

$$\mathsf{PRF}(K, x) = \mathsf{PRF}(K\{x^*\}, x)$$

Proof of Security [this work]

- **Selective pseudorandomness:** $i\mathcal{O}$

Game 3

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$K(x^*) \leftarrow \mathsf{Punctured}(K, x^*)$

$b^* \| r^* = \mathsf{PRF}(K, x^*)$

$c^* = \mathsf{Com}(b^*; r^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_{K(x^*),c^*}])$

$y^* = b^*$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K(x^*),\cdot),\mathsf{C}(K(x^*),\cdot)}(y^*)$

output 1 iff $b' = 0$

For S s.t. $x^* \notin S$, it always holds that for $x \in S$

$\mathsf{PRF}(K, x) = \mathsf{PRF}(K\{x^*\}, x)$

Proof of Security [this work]

- **Selective pseudorandomness:** *Pseudorandomness of punctured PRFs*

Game 4

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$K(x^*) \leftarrow \mathsf{Punctured}(K, x^*)$

$t^* \leftarrow \{0,1\}^{\ell+1}, t^* = b^* \| r^*$

$c^* = \mathsf{Com}(b^*; r^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_{K(x^*),c^*}])$

$y^* = b^*$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K(x^*),\cdot),\mathsf{C}(K(x^*),\cdot)}(y^*)$

output 1 iff $b' = 0$

Proof of Security [this work]

- **Selective pseudorandomness:** *Computational hiding property of Com*

Game 5

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$K(x^*) \leftarrow \mathsf{Punctured}(K, x^*)$

$t^* \leftarrow \{0,1\}^{\ell+1}, t^* = b^* \| r^*$

$c^* = \mathsf{Com}(b^*; r^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_{K(x^*),c^*}])$

$\tilde{b} \leftarrow \{0,1\}, y^* = \tilde{b}$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K(x^*),\cdot)}, \mathsf{C}(K(x^*),\cdot)(y^*)$

output 1 iff $b' = 0$

Proof of Security [this work]

- **Selective pseudorandomness:** *Pseudorandomness of punctured PRFs*

Game 6

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$K(x^*) \leftarrow \mathsf{Punctured}(K, x^*)$

$b^* \| r^* = \mathsf{PRF}(K, x^*)$

$c^* = \mathsf{Com}(b^*; r^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_{K(x^*), c^*}])$

$\tilde{b} \leftarrow \{0,1\}, y^* = \tilde{b}$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K(x^*),\cdot),\mathsf{C}(K(x^*),\cdot)}(y^*)$

output 1 iff $b' = 0$

Proof of Security [this work]

- **Selective pseudorandomness:** *functionality preserved under puncturing*

Game 7

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$K(x^*) \leftarrow \mathsf{Punctured}(K, x^*)$

$b^* \| r^* = \mathsf{PRF}(K, x^*)$

$c^* = \mathsf{Com}(b^*; r^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_{K(x^*),c^*}])$

$\tilde{b} \leftarrow \{0,1\}, y^* = \tilde{b}$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K,\cdot),\mathsf{C}(K(x^*),\cdot)}(y^*)$

output 1 iff $b' = 0$

Proof of Security [this work]

- **Selective pseudorandomness:** $i\mathcal{O}$

Game 8

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$K(x^*) \leftarrow \mathsf{Punctured}(K, x^*)$

$b^* \| r^* = \mathsf{PRF}(K, x^*)$

$c^* = \mathsf{Com}(b^*; r^*)$

$PK = i\mathcal{O}([\mathsf{Verify}_{K(x^*), c^*}])$

$\tilde{b} \leftarrow \{0,1\}, y^* = \tilde{b}$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K,\cdot),\mathsf{C}(K,\cdot)}(y^*)$

output 1 iff $b' = 0$

Proof of Security [this work]

- **Selective pseudorandomness:** $i\mathcal{O}$

Game 9

$x^* \leftarrow \mathcal{A}_1(1^\lambda)$

$K \leftarrow \mathsf{Key}_{\mathsf{PRF}}(1^\lambda)$

$PK = i\mathcal{O}([\mathsf{Verify}_K])$

$\tilde{b} \leftarrow \{0,1\}, y^* = \tilde{b}$

$b' \leftarrow \mathcal{A}_2^{\mathsf{Po}(K,\cdot),\mathsf{C}(K,\cdot)}(y^*)$

output 1 iff $b' = 0$

Thank You!