Constrained Verifiable Random Functions

Georg Fuchsbauer
IST Austria

SCN 2014, 3 September 2014

(Full version: eprint 2014/537)

Overview

Pseudorandom functions (PRF) $\overset{\subseteq}{\longrightarrow}$ Constrained PRF

$\subseteq$

Verifiable random functions (VRF)

Overview

$$
\begin{tikzcd}
\text{Pseudorandom functions (PRF)} \arrow[r, dash, "\subseteq"] \arrow[d, dash, "\subseteq"] & \text{Constrained PRF} \arrow[d, dash, "\subseteq"] \\
\text{Verifiable random functions (VRF)} \arrow[r, dash, "\subseteq"] & \boxed{\text{Constrained VRF}}
\end{tikzcd}
$$

- Formal definition
- Constructions

PRFs

- Pseudorandom function [GGM86]:
  - Function $F\colon \mathcal{K} \times \{0,1\}^n \to \mathcal{Y}$

$$\begin{array}{ccc} k \leftarrow_{\$} \mathcal{K} & & G \leftarrow_{\$} \{F \mid F\colon \{0,1\}^n \to \mathcal{Y}\} \\ F(k,\cdot) & \approx_c & G(\cdot) \end{array}$$

PRFs

- Pseudorandom function [GGM86]:
  - Function $F: \mathcal{K} \times \{0,1\}^n \to \mathcal{Y}$

A PRF key is a *compact description* of an exponentially long (pseudo) random string.

PRFs

- Pseudorandom function [GGM86]:
  - Function $F: \mathcal{K} \times \{0,1\}^n \to \mathcal{Y}$

A PRF key is a *compact description* of an exponentially long (pseudo) random string.

- Application:
  - Symmetric encryption: Key: $k \leftarrow_{\$} \mathcal{K}$

    Encryption: $r \leftarrow_{\$} \{0,1\}^n;\ C := (r, F(k,r) \oplus M)$

Constrained PRFs

- Constrained PRF [BW13, KPTZ13, BGI14]:
  - Function $F\colon \mathcal{K} \times \{0,1\}^n \to \mathcal{Y}$
    for set system $\mathcal{S} \subseteq \mathcal{P}(\{0,1\}^n)$
  - Algorithms:
    - $\mathsf{Constr}(k, S \in \mathcal{S}) \to k_S$
    - $\mathsf{Eval}(k_S, x \in \{0,1\}^n) \to y$

Constrained PRFs

- Constrained PRF [BW13, KPTZ13, BGI14]:
  - Function $F\colon \mathcal{K} \times \{0,1\}^n \to \mathcal{Y}$
    for set system $\mathcal{S} \subseteq \mathcal{P}(\{0,1\}^n)$
  - Algorithms:
    - $\mathsf{Constr}(k, S \in \mathcal{S}) \to k_S$
    - $\mathsf{Eval}(k_S, x \in \{0,1\}^n) \to y$

$$k_S \leftarrow \mathsf{Constr}(k, S)$$

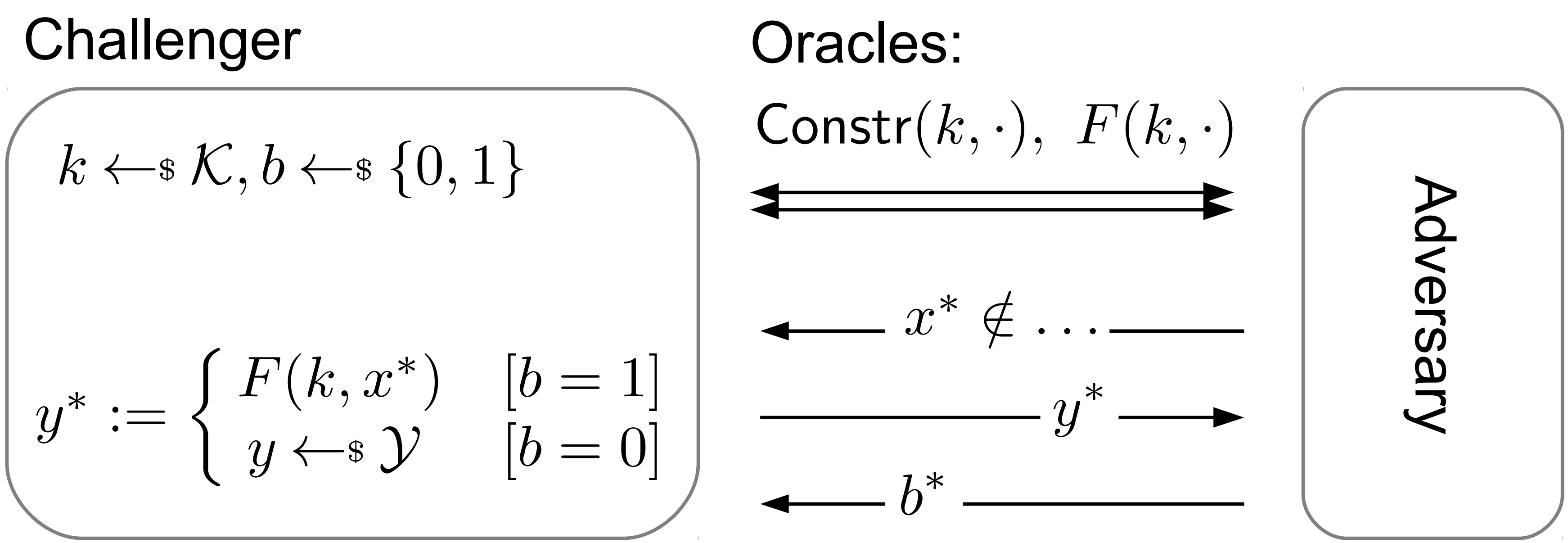
$$\Rightarrow \quad \mathsf{Eval}(k_S, x) = \begin{cases} F(k, x) & \text{if } x \in S \\ \perp & \text{otherwise} \end{cases}$$

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*

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*



Instantiations of Constrained PRFs

- Instantiations for set systems $\mathcal{S} \subseteq \mathcal{P}(\{0,1\}^n)$
  - prefix-constrained PRF [BW13, KPTZ13, BGI14]:
    keys for sets $\mathrm{S}_p = \left\{p\|z \;\middle|\; z \in \{0,1\}^{n-|p|}\right\}$

Instantiations of Constrained PRFs

- Instantiations for set systems $\mathcal{S} \subseteq \mathcal{P}(\{0,1\}^n)$
  - prefix-constrained PRF [BW13, KPTZ13, BGI14]:
    keys for sets $\mathrm{S}_p = \left\{ p \| z \;\middle|\; z \in \{0,1\}^{n-|p|} \right\}$
  - bit-fixing PRF [BW13]:
    keys for sets defined by $v \in \{0,1,?\}^n$ as

$$\mathrm{S}_v = \left\{ z \in \{0,1\}^n \;\middle|\; \forall i : z_i = v_i \vee v_i = ? \right\}$$

Instantiations of Constrained PRFs

- Instantiations for set systems $\mathcal{S} \subseteq \mathcal{P}(\{0,1\}^n)$
  - prefix-constrained PRF [BW13, KPTZ13, BGI14]:

    keys for sets $\mathrm{S}_p = \left\{ p \| z \;\middle|\; z \in \{0,1\}^{n-|p|} \right\}$

  - bit-fixing PRF [BW13]:

    keys for sets defined by $v \in \{0,1,?\}^n$ as

    $$\mathrm{S}_v = \left\{ z \in \{0,1\}^n \;\middle|\; \forall i : z_i = v_i \vee v_i = ? \right\}$$

  - circuit-constrained PRF [BW13]:

    keys defined by circuit *C*: $\mathrm{S}_C = \left\{ z \in \{0,1\}^n \;\middle|\; C(z) = 1 \right\}$

Applications of Constrained PRFs

- Identity-based non-interactive key exchange (ID-NIKE) from bit-fixing PRF [BW13]
- Broadcast encryption with optimal ciphertext length [BW13]

Applications of Constrained PRFs

- Identity-based non-interactive key exchange (ID-NIKE) from bit-fixing PRF [BW13]

- Broadcast encryption with optimal ciphertext length [BW13]

- Punctured PRFs [BW13, KPTZ13, BGI14]:
  - constr. keys for domain $\mathcal{X}^* := \{0,1\}^n \setminus \{x^*\}$
  - many applications in combination with indistinguishability obfuscation [GGH+13]

VRFs

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

VRFs

- Verifiable random function [MRV99]:
  - Function $F \colon \mathcal{K} \times \{0,1\}^n \to \mathcal{Y}$
  - Algorithms:
    - $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$
    - $\mathsf{Prove}(\mathsf{sk}, x) \to (y, \pi)$
    - $\mathsf{Verify}(\mathsf{pk}, x, y, \pi) \to 0/1$
- Provability

$$\begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda) \\ (y, \pi) \leftarrow \mathsf{Prove}(\mathsf{sk}, x) \end{array} \quad \Rightarrow \quad \begin{array}{l} y = F(\mathsf{sk}, x) \\ \mathsf{Verify}(\mathsf{pk}, x, y, \pi) = 1 \end{array}$$

Security of VRFs

- Uniqueness:
  - For all λ, pk, x, y_0, π_0, y_1, π_1 :

$$y_0 \neq y_1 \Rightarrow \left\{ \begin{array}{l} \mathsf{Verify}(\mathsf{pk}, x, y_0, \pi_0) = 0 \\ \quad \vee \ \mathsf{Verify}(\mathsf{pk}, x, y_1, \pi_1) = 0 \end{array} \right.$$

Security of VRFs

- Uniqueness:
  - For all λ, pk, x, y_0, π_0, 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:
  - Adv gets Prove oracle
  - submits x^* that has not been queried
  - receives either $F(\mathsf{sk}, x^*)$ or $y \xleftarrow{\$} \mathcal{Y}$

Security of VRFs

- Uniqueness:
  - For all $\lambda,\ \mathsf{pk},\ x,\ y_0, \pi_0,\ y_1, \pi_1$:

A VRF public key can be seen as
a *compact commitment* to an
exponential number of (pseudo) random bits.

- Ps
  - Adv gets Prove oracle
  - submits x^* that has not been queried
  - receives either $F(\mathsf{sk}, x^*)$ or $y \leftarrow_{\$} \mathcal{Y}$

Application of VRFs

- Micropayments:
  - e.g. many payments of 1¢, too expensive to process

$\Rightarrow$ "Rivest's lottery":

- User U pays merchant M with cheque:

$$C := \mathsf{Sign}(\mathsf{sk}_U, T)$$

- Rate s: with $Pr = s$, a cheque is "payable"
  - payable: M receives $\frac{1}{s}$ ¢
  - else cheque is discarded

Application of VRFs

- Micropayments:
  - e.g. many payment

$\Rightarrow$ "Rivest's lottery":

- User U pays me
  $C :=$
- Rate s: with $Pr = s$, a cheque is "payable"
  - payable: M receives $\frac{1}{s}$ ¢
  - else cheque is discarded

How do we decide which C should be payable (in fair way)?

- M publishes pk_M for VRF with $\mathcal{Y} := [0, 1]$
- C is payable if $F(\mathsf{sk}_M, C) < s$

Constrained
Verifiable Random Functions

Constrained VRFs

- Constrained VRF [This work]:
  - Function $F \colon \mathcal{K} \times \{0,1\}^n \to \mathcal{Y}$

    for set system $\mathcal{S} \subseteq \mathcal{P}(\{0,1\}^n)$
  - Algorithms:
    - $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$
    - $\mathsf{Constr}(\mathsf{sk}, S) \to \mathsf{sk}_S$
    - $\mathsf{Prove}(\mathsf{sk}_S, x) \to (y, \pi)$
    - $\mathsf{Verify}(\mathsf{pk}, x, y, \pi) \to 0/1$

Constrained VRFs

- Constrained VRF [This work]:
  - Function $F : \mathcal{K} \times \{0,1\}^n \to \mathcal{Y}$

    for set system $\mathcal{S} \subseteq \mathcal{P}(\{0,1\}^n)$
  - Algorithms:
    - $\mathsf{Setup}(1^\lambda) \to (\mathsf{pk}, \mathsf{sk})$
    - $\mathsf{Constr}(\mathsf{sk}, S) \to \mathsf{sk}_S$
    - $\mathsf{Prove}(\mathsf{sk}_S, x) \to (y, \pi)$
    - $\mathsf{Verify}(\mathsf{pk}, x, y, \pi) \to 0/1$

$$\mathsf{sk}_S \leftarrow \mathsf{Constr}(\mathsf{sk}, S)$$

$$(y, \pi) \leftarrow \mathsf{Prove}(\mathsf{sk}_S, x)$$

$$\Rightarrow \begin{cases} x \in S \;\Rightarrow\; y = F(\mathsf{sk}, x),\ \mathsf{Verify}(\mathsf{pk}, x, y, \pi) = 1 \\ x \notin S \;\Rightarrow\; (y, \pi) = (\bot, \bot) \end{cases}$$

Security of Constrained VRFs

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

Security of Constrained VRFs

- **Uniqueness:**

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

- **Pseudorandomness of constrained PRFs:**

Challenger

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{Setup}$$

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

$$y^* := \begin{cases} F(\mathsf{sk}, x^*) & [b = 1] \\ y \leftarrow_{\$} \mathcal{Y} & [b = 0] \end{cases}$$

Oracles:

$\mathsf{Constr}(\mathsf{sk}, \cdot), \mathsf{Prove}(\mathsf{sk}, \cdot)$

$\xleftarrow{\quad x^* \notin \ldots \quad}$

$\xrightarrow{\quad y^* \quad}$

$\xleftarrow{\quad b^* \quad}$

Adversary

Security of Constrained VRFs

- **Uniqueness:** $$y_0 \neq y_1 \Rightarrow \begin{cases} \mathsf{Verify}(\mathsf{pk}, x, y_0, \pi_0) = 0 \\ \quad \vee\ \mathsf{Verify}(\mathsf{pk}, x, y_1, \pi_1) = 0 \end{cases}$$
- **Constraint-hiding:** $\mathsf{Prove}(\mathsf{sk}, x) \approx_c \mathsf{Prove}(\mathsf{Constr}(\mathsf{sk}, S), x)$
- **Pseudorandomness of constrained PRFs:**

Challenger

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{Setup}$$
$$b \leftarrow_{\$} \{0, 1\}$$
$$y^* := \begin{cases} F(\mathsf{sk}, x^*) & [b = 1] \\ y \leftarrow_{\$} \mathcal{Y} & [b = 0] \end{cases}$$

Oracles:

$\mathsf{Constr}(\mathsf{sk}, \cdot), \mathsf{Prove}(\mathsf{sk}, \cdot)$ $\Longleftrightarrow$

$\xleftarrow{\ x^* \notin \ldots\ }$

$\xrightarrow{\ y^*\ }$

$\xleftarrow{\ b^*\ }$

Adversary

Possible Application of Constrained VRF

- Micropayments:

"Rivest's lottery"

- M publishes pk_M for VRF with $\mathcal{Y} := [0, 1]$
- C is payable if $F(\mathsf{sk}_M, C) < s$

Possible Application of Constrained VRF

- Micropayments:

"Rivest's lottery"

- M publishes pk_M for VRF with $\mathcal{Y} := [0,1]$
- C is payable if $F(\mathsf{sk}_M, C) < s$

Drawback

- need PKI for merchants' keys pk_M

Identity-based solution?

Possible Application of Constrained VRF

- Micropayments:

"Rivest's lottery"

- M publishes pk_M for VRF with $\mathcal{Y} := [0,1]$
- C is payable if $F(\mathsf{sk}_M, C) < s$

Drawback

- need PKI for merchants' keys pk_M

$\Rightarrow$ constrained VRFs

- every M uses same key sk
- C is payable if $F(\mathsf{sk}, \mathsf{id}_M \| C) < s$
- Merchant M gets constr. key sk_M for set $(\mathsf{id}_M, ?\ldots?)$

Constructions

PRFs

- from PRG [GGM86]
- under DDH [NR97]

VRFs

- under q-DDHI [DY05]
  but: poly-size domain only!
- under q-type assumptions,
  value in target group
  large proofs [HW10,ACF13]

Constructions

PRFs

- from PRG [GGM86]
- under DDH [NR97]

constrained PRFs

- prefix-fixing:
  - from PRG [BW13, KPTZ13, BGI14]
- bit-fixing, circuit-constr:
  - from multilin. maps
  - under MDDH [BW13]

VRFs

- under q-DDHI [DY05]
  but: poly-size domain only!
- under q-type assumptions, value in target group large proofs [HW10,ACF13]

Constructions

PRFs

- from PRG [GGM86]
- under DDH [NR97]

constrained PRFs

- prefix-fixing:
  - from PRG [BW13, KPTZ13, BGI14]
- bit-fixing, circuit-constr:
  - from multilin. maps
  - under MDDH [BW13]

VRFs

- under q-DDHI [DY05] but: poly-size domain only!
- under q-type assumptions, value in target group large proofs [HW10,ACF13]

constrained VRFs

- bit-fixing, circuit-constr:
  - from multilin. maps
  - under MDDH [this work]

Constructions

PRFs

- from PRG
- under

constrained

- prefix-fixing:
  - from PRG [BW13, KPTZ13, BGI14]
- bit-fixing, circuit-constr:
  - from multilin. maps
  - under MDDH [BW13]

VRFs

DDHI [DY05]
ze domain only!
e assumptions,
s [HW10,ACF13]

constrained VRFs

- bit-fixing, circuit-constr:
  - from multilin. maps
  - under MDDH [this work]

- based on same assumption
- same function values

→ verifiability "for free"

Construction of Constrained VRFs

- Multilinear maps:
  - Groups $(\mathbb{G}_1, \ldots, \mathbb{G}_\kappa)$, each generated by g_i
  - Maps $e_{i,j}$ s.t. $\mathrm{e}(\mathrm{g}_i^a, g_j^b) = g_{i+j}^{ab}$

- $\kappa-$MDDH assumption:

  given $g_1, g_1^{c_1}, \ldots, g_1^{c_{\kappa+1}}$ then $(g_\kappa)^{\prod_{j=1}^{\kappa+1} c_j}$ looks random

Construction of Constrained VRFs

- Boneh-Waters cPRF:

  (bit-fixing)

$$k := \left\{ \begin{array}{cc} \alpha, & d_{1,0}, \ldots, d_{n,0} \\ & d_{1,1}, \ldots, d_{n,1} \end{array} \right\}$$

$$F(k, x) := (g_{n+1})^{\alpha \prod_{i=1}^{n} d_{i,x_i}}$$

$\boxed{\text{secure under } (n+1)\text{-MDDH}}$

Construction of Constrained VRFs

- Boneh-Waters cPRF: (bit-fixing)

$$k := \left\{ \begin{array}{ll} \alpha, & d_{1,0}, \ldots, d_{n,0} \\ & d_{1,1}, \ldots, d_{n,1} \end{array} \right\}$$

$$F(k, x) := (g_{n+1})^{\alpha \prod_{i=1}^{n} d_{i,x_i}}$$

- Observation: Even when $D_{i,j} := (g_1)^{d_{i,j}}$ and $(g_2)^{\alpha}$ are public, F still pseudorandom

Construction of Constrained VRFs

- Boneh-Waters cPRF:
  (bit-fixing)

$$k := \left\{ \begin{array}{cc} \alpha, & d_{1,0}, \ldots, d_{n,0} \\ & d_{1,1}, \ldots, d_{n,1} \end{array} \right\}$$

$$F(k,x) := (g_{n+1})^{\alpha \prod_{i=1}^{n} d_{i,x_i}}$$

- Observation: Even when $D_{i,j} := (g_1)^{d_{i,j}}$ and $(g_2)^{\alpha}$ are public, F still pseudorandom
- Split $\alpha = \beta \cdot \gamma$, publish $B := (g_1)^{\beta}$ and $C := (g_1)^{\gamma}$

Construction of Constrained VRFs

- Boneh-Waters cPRF:
  (bit-fixing)

$$k := \left\{ \begin{array}{ll} \alpha, & d_{1,0}, \ldots, d_{n,0} \\ & d_{1,1}, \ldots, d_{n,1} \end{array} \right\}$$

$$F(k,x) := (g_{n+1})^{\alpha \prod_{i=1}^{n} d_{i,x_i}}$$

- Observation: Even when $D_{i,j} := (g_1)^{d_{i,j}}$ and $(g_2)^{\alpha}$ are public, F still pseudorandom
- Split $\alpha = \beta \cdot \gamma$, publish $B := (g_1)^{\beta}$ and $C := (g_1)^{\gamma}$
- Observation: $D := (g_{n+1})^{\beta \prod_{i=1}^{n} d_{i,x_i}}$ publicly computable

Construction of Constrained VRFs

- Boneh-Waters cPRF:

  (bit-fixing)

$$k := \left\{ \begin{array}{ll} \alpha, & d_{1,0}, \ldots, d_{n,0} \\ & d_{1,1}, \ldots, d_{n,1} \end{array} \right\}$$

$$F(k,x) := (g_{n+1})^{\alpha \prod_{i=1}^{n} d_{i,x_i}}$$

- Observation: Even when $D_{i,j} := (g_1)^{d_{i,j}}$ and $(g_2)^{\alpha}$ are public, F still pseudorandom
- Split $\alpha = \beta \cdot \gamma$, publish $B := (g_1)^{\beta}$ and $C := (g_1)^{\gamma}$
- Observation: $D := (g_{n+1})^{\beta \prod_{i=1}^{n} d_{i,x_i}}$ publicly computable
- Define proof: $P := (g_n)^{\beta \prod_{i=1}^{n} d_{i,x_i}}$

Construction of Constrained VRFs

- Boneh-Waters cPRF:
  (bit-fixing)

$$k := \left\{ \begin{matrix} \alpha, & d_{1,0}, \ldots, d_{n,0} \\ & d_{1,1}, \ldots, d_{n,1} \end{matrix} \right\}$$

$$F(k,x) := (g_{n+1})^{\alpha \prod_{i=1}^{n} d_{i,x_i}}$$

- Observation: Even when $D_{i,j} := (g_1)^{d_{i,j}}$ and $(g_2)^{\alpha}$ are public, F still pseudorandom
- Split $\alpha = \beta \cdot \gamma$, publish $B := (g_1)^{\beta}$ and $C := (g_1)^{\gamma}$
- Observation: $D := (g_{n+1})^{\beta \prod_{i=1}^{n} d_{i,x_i}}$ publicly computable
- Define proof: $P := (g_n)^{\beta \prod_{i=1}^{n} d_{i,x_i}}$
- Verification of y: $e(P, g_1) \stackrel{?}{=} D \quad \wedge \quad e(P, C) \stackrel{?}{=} y$

Thank you