Oblivious Polynomial Evaluation and Secure Set-Intersection from Algebraic PRFs

TCC 2015

Carmit Hazay

Bar-Ilan University

אוניברסיטת בר-אילן
Bar-Ilan University

Pseudorandom Functions (PRFs)

- PRF is indistinguishable from a random function

PRF: Let $k \leftarrow \{0,1\}^n$ Given x output $F_k(x)$	$\approx$	**Random function:** Let $R: \{0,1\}^n \rightarrow \{0,1\}^n$ Given x output $R(x)$

- Adversary makes polynomial number of queries

Algebraic PRFs [BGV11]

- PRFs with Abelian group range
- Support fast batch PRFs evaluation given the PRF key
- Originally introduced in the context of verifiable delegation of polynomial evaluation

Algebraic PRFs [BGV11]

Closed from efficiency:

$$\mathrm{CFEval}_{h,z}(x,k) = \prod_{i=0}^{l} \left[\mathrm{PRF}_k(z_i)\right]^{h_i(x)}$$

Running time of CFeval is **sublinear** in **l**

Special case: **z = (0,…,d)** and $\mathbf{h_i(x)=x^i}$

Algebraic PRFs [BGV11]

- Implementations in prime order groups:
  - **Strong DDH:**

$$\left|\Pr\left[D\left(G, p, g, g^{x}, g^{x^2}, \ldots, g^{x^d}\right)\right] - \left[D\left(G, p, g, g^{x_1}, g^{x_2}, \ldots, g^{x_d}\right)\right]\right| \leq \mathrm{negl}(\cdot)$$

$$\mathrm{PRF}_k(x) = g^{k_0 k_1^x}$$

  - **DDH** (Naor-Reingold):

$$\mathrm{PRF}_k(x_1,\ldots,x_m) = g^{k_0 \prod_{i=1}^{m} k_i^{x_i}}$$

Algebraic PRFs [BGV11]

- **Strong DDH**
  - CEEval with constant overhead follows from the identity:

$$\sum_{i=0}^{d} k_0 k_1^i x^i = \frac{k_0\left(k_1^{d+1} x^{d+1} - 1\right)}{k_1 x - 1}$$

- **DDH** [NR]
  - CEEval with log d overhead follows from the identity:

$$\sum_{j=0}^{d} k_0 \prod_{i=1}^{j+1} k_i^{x_i} = k_0(1+k_1 x)(1+k_2 x^2)\cdots(1+k_{\log d}\, x^{d})$$

Verifiable Polynomial Evaluation [BGV11]

<u>Setup:</u> The client stores $\mathbf{Q}(\cdot)=(\mathbf{q_0},\ldots\mathbf{q_d})$ together with $(\mathbf{F_k}(\mathbf{0})\mathbf{g}^{\mathbf{aq_0}},\ldots\mathbf{F_k}(\mathbf{d})\mathbf{g}^{\mathbf{aq_d}})=(\mathbf{g}^{\mathbf{r_0+aq_0}},\ldots,\mathbf{g}^{\mathbf{r_d+aq_d}})$

<u>Evaluation:</u> Given $\mathbf{x}$, the server returns $\mathbf{Q}(\mathbf{x})$ and $\mathbf{g}^{\sum_i (\mathbf{r_i}+\mathbf{aq_i})^{\mathbf{x}^{\mathbf{i}}}}$

<u>Verification:</u> Given $\mathbf{Q}(\mathbf{x})$ and proof $\mathbf{w}$ check that $\mathbf{w}=\mathbf{g}^{\sum_i \mathbf{r}^{\mathbf{x}^{\mathbf{i}}}+\mathbf{aQ}(\mathbf{x})}$

- Computing $\mathbf{g}^{\sum_i \mathbf{r_i}^{\mathbf{x}^{\mathbf{i}}}}$ requires sublinear time in **d** given the PRF key
- The protocol is not private

Oblivious Polynomial Evaluation (OPE)

Sender

Receiver

Inputs:	$Q(\cdot)=(q_0,\ldots q_d)$	x
Outputs:	—	Q(x)

Oblivious Polynomial Evaluation (OPE)

- Very useful building block:
  - RSA key generation [Gil99]
  - Approximation of Taylor series [LP02]
  - Set-intersection [FNP04]
  - Oblivious keyword search [FIPR05]
  - Secure equality of strings [NP06]
  - Data entanglement [ADDV12]

Prior maliciously secure work uses cut-and-choose or somewhat homomorphic SIMD approach

OPE from Algebraic PRFs

- **Our results:**

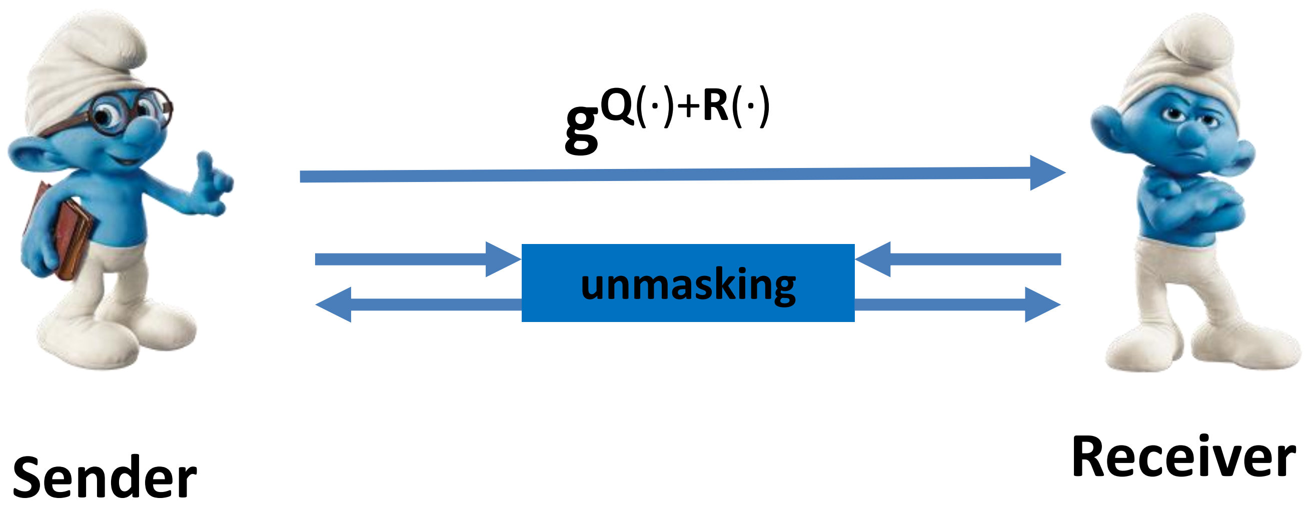
  OPE in the exponent with malicious security

- **Two phases protocol:**
  1. Use algebraic PRFs to mask the sender's polynomial
  2. Unmask the evaluated masked polynomial

OPE from Algebraic PRFs

Masking: Given input $\mathbf{g}^{Q(\cdot)}=(\mathbf{g}^{q_0},\ldots\mathbf{g}^{q_d})$ the sender sends the masked polynomial $\mathbf{g}^{Q(\cdot)+R(\cdot)} = (\mathbf{g}^{q_0+r_0},\ldots,\mathbf{g}^{q_d+r_d})$

Unmasking: The receiver computes $\mathbf{g}^{Q(x)+R(x)}$ and the parties compute $\mathbf{g}^{R(x)}$



OPE from Algebraic PRFs

- Unmasking requires computing $\mathbf{g}^{\sum_i \mathbf{r_i} \mathbf{x}^{\mathbf{i}}}$ which can be carried out in **sublinear time in d**

- <u>Overhead:</u>
  - **d+1** exponentiations in masking phase
  - O(**1**)/O(**log d**) exponentiations in unmasking phase

Secure Set-Intersection

Sender

Receiver

Inputs:	X	Y
Outputs:	—	X∩Y

Secure Set-Intersection

- Intensively studied due to many applications [FNP04,KS05,DSMRY09,**JL09**,JL10,HL10,**HN12**]

- Two common approaches in the plain model:
  - Oblivious polynomial evaluation [FNP04]
  - Oblivious PRF evaluation [FIPR05,LP10]

Set-Intersection - The OPE Approach

1. On input ($\mathbf{x_1},\ldots,\mathbf{x_m}$), sender masks polynomial $\mathbf{Q}(\mathbf{t})=(\mathbf{x_1}-\mathbf{t}) \cdots (\mathbf{x_m}-\mathbf{t})$ and sends $\mathbf{g}^{\mathbf{Q}(\mathbf{t})+\mathbf{R}(\mathbf{t})}$
2. For each $\mathbf{y}\in\mathbf{Y}$
   - Receiver evaluates $\mathbf{g}^{\mathbf{Q}(\mathbf{y})+\mathbf{R}(\mathbf{y})}$
   - The parties "unmask" the result by verifying whether $\mathbf{g}^{\mathbf{Q}(\mathbf{y})+\mathbf{R}(\mathbf{y})} = \mathbf{g}^{\mathbf{R}(\mathbf{y})}$

- Overhead is $|\mathbf{X}|\cdot|\mathbf{Y}|$
  - Reduce overhead using hash functions

Set-Intersection - The OPE Approach and Hash Functions

- Split the set into bins
  - Receiver evaluates a small degree poly each time
  - Easy to evaluate a polynomial of a particular bin

- Correctness:
  - Sender proves that the polynomials are correct, potentially can use zero polynomials or too many elements
  - Receiver proves that it uses the same input when more than one hash is used

Set-Intersection - The OPE Approach and Hash Functions

- <u>Overhead:</u>

  - **O**($\mathbf{m_X + m_Y loglog\ m_X}$) under **d-strong** assumption

  - **O**($\mathbf{m_X + m_Y log\ m_X}$) under **DDH** assumption

    Better than [HN12]!

Set-Intersection - The Oblivious PRF Approach

- The **oblivious PRF** functionality:

Sender **Receiver**

Inputs:	**k**	**x**
Outputs:	—	$F_k(x)$

Set-Intersection - The Oblivious PRF Approach

- **First phase:** sender picks PRF key **k** and sends $\mathbf{F_K}(\mathbf{x})$ for all $\mathbf{x} \in \mathbf{X}$
- **Second phase:** parties run oblivious PRF, receiver learns $\mathbf{F_K}(\mathbf{y})$ for all $\mathbf{y} \in \mathbf{Y}$

Requires committed oblivious PRF!

Set-Intersection - The Committed Oblivious PRF Approach

The **committed oblivious PRF** functionality:

Sender **Receiver**

Inputs:	k	$x_1,\ldots,x_n$
Outputs:	—	$F_k(x_1),\ldots, F_k(x_n)$

Set-Intersection - The Committed Oblivious PRF Approach

- Algebraic PRFs easily imply committed PRFs
  - Use same protocols for unmasking
- One particular example is a simple committed Naor-Reingold PRF
  - Prior to that, no simple committed protocol

Set-Intersection - The Committed Oblivious PRF Approach

- <u>Overhead:</u>

  - **O**($\mathbf{m_X}$+$\mathbf{m_Y}$) under strong-DDH assumption

    Same as [JL09] but for prime order groups

    No setup (CRS)

    Proof complexity is independent of PRF domain

  - **O**(($\mathbf{m_X}$+$\mathbf{m_Y}$))**log** ($\mathbf{m_X}$+$\mathbf{m_Y}$)) under DDH assumption

Future Research

- Use algebraic PRFs for more applications
- Construct new algebraic PRFs for large domains

- Recent related work:
  - Aggregate Pseudorandom Functions and Connections to Learning

    Aloni Cohen and Shafi Goldwasser and Vinod Vaikuntanathan