Constraint hiding constrained PRF for NC1 from LWE

Ran Canetti, Yilei Chen, from Boston University
Once upon a time, a Swede, a Dane, and a Norwegian found themselves on a small island.
There's a cannibal tribe on the island. They imprison the three men. Each of the three men is allowed to make a final wish.
Norwegian: I want to meet my wife.
Norwegian: I want to meet my wife. The cannibals agree. Finally they eat the Norwegian and turn his skin into a canoe.
Swede: I want to have another cigarette.
Swede: I want to have another cigarette. The cannibals agree. Finally they eat the Swede and turn his skin into a canoe.
Dane: ...
Dane: I want a puncturable PRF!
Dane: I want a puncturable PRF!
Then you cannot turning my skin into a canoe!!!!!!!
Puncturable/constrained PRF
[Boneh, Waters 13, Kiayias, Papadopoulos, Triandopoulos, Zacharias 13, Boyle, Goldwasser, Ivan 14, Sahai, Waters 14]

K

original key
Puncturable/constrained PRF

[Boneh, Waters 13, Kiayias, Papadopoulos, Triandopoulos, Zacharias 13, Boyle, Goldwasser, Ivan 14, Sahai, Waters 14]

\[
Punctured\text{ key}\]

\[
original\text{ key}
\]

\[
K
\]

\[
F_{K\{x^*\}}(x) = \begin{cases} 
? & \text{, if } x=x^* \\
F_K(x), & \text{else}
\end{cases}
\]

Puncture(K, x*) => K{x*} s.t. F_k(x*) is pseudorandom, give the K{x*} that preserve the original outputs elsewhere.
Puncturable/constrained PRF

\[ \text{Puncturable/constrained PRF} \]

[Boneh, Waters 13, Kiayias, Papadopoulos, Triandopoulos, Zacharias 13, Boyle, Goldwasser, Ivan 14, Sahai, Waters 14]

\[ \text{Puncture}(K, x^*) \Rightarrow K\{x^*\} \text{ s.t. } F_k(x^*) \text{ is pseudorandom, give the } K\{x^*\} \text{ that preserve the original outputs elsewhere.} \]

\[ F_{K\{x^*\}}(x) = \begin{cases} F_K(x), & \text{else} \\ ?, & \text{if } x = x^* \end{cases} \]

In general: \( \text{Constrain}(K, C) \Rightarrow K\{C\} \)
Puncturable/constrained PRF
[Boneh, Waters 13, Kiayias, Papadopoulos, Triandopoulos, Zacharias 13, Boyle, Goldwasser, Ivan 14, Sahai, Waters 14]

\[ F_{K(x^*)}(x) = \begin{cases} \text{? , if } x=x^* \\ F_K(x), \text{ else} \end{cases} \]

Puncturing: \( K \rightarrow K\{x^*\} \) s.t. \( F_k(x^*) \) is pseudorandom, give the \( K\{x^*\} \) that preserve the original outputs elsewhere.

In general: \( \text{Constrain}(K, C) \rightarrow K\{C\} \)

They have many applications (delegate PRF, broadcast encryption, identity-based KE, ...) best known for being good friends of iO
Puncturable PRF from GGM

[Goldreich, Goldwasser, Micali 84]
Puncturable PRF from GGM
[Goldreich, Goldwasser, Micali 84]
Puncturable PRF from GGM
[Goldreich, Goldwasser, Micali 84]

The constrained key reveals the point $x$
What about hiding the constraint?
Dane: I want a puncturable PRF!
Then you cannot turning my skin into a canoe!!!!!!
Dane: I want a puncturable PRF!
Then you cannot turning my skin into a canoe!!!!!!
YOU DON'T EVEN KNOW HOW I PUNCTURED
Some motivation scenario:
“Tricky” encryption key

Some 3-letter agent
Some motivation scenario: “Tricky” encryption key

full key

Some 3-letter agent
Some motivation scenario: “Tricky” encryption key

Some 3-letter agent
Boneh, Lewi, Wu (PKC17, eprint 2015/1167)

What

Where

How
Boneh, Lewi, Wu (PKC17, eprint 2015/1167)

**What** are Constraint-Hiding CPRFs: an indistinguishability-based definition

**Where**

**How**
Boneh, Lewi, Wu (PKC17, eprint 2015/1167)

What are Constraint-Hiding CPRFs: an indistinguishability-based definition

Where to find them (secure for many keys):
- $iO(PPRF)$ is CHCPRF
- Can achieve bit-fixing, puncturing under multilinear DDH, subgroup-hiding
Boneh, Lewi, Wu (PKC17, eprint 2015/1167)

What are Constraint-Hiding CPRFs: an indistinguishability-based definition

Where to find them (secure for many keys):
- \text{iO}(PPRF) is CHCPRF
- Can achieve bit-fixing, puncturing under multilinear DDH, subgroup-hiding

How to use them:
Private-key deniable encryption,
Privately-detectable watermarking,
Searchable encryption
This talk:
Canetti, Chen (Eurocrypt17)

What

Where

How
This talk:
Canetti, Chen (Eurocrypt17)

What are Constraint-Hiding CPRFs:
A simulation-based definition of CHCPRF

Where

How
This talk:
Canetti, Chen (Eurocrypt17)

What are Constraint-Hiding CPRFs:
A simulation-based definition of CHCPRF

Where to find them:
Simulation-based 1-key CHCPRFs for NC1 from Learning With Errors

How
This talk:
Canetti, Chen (Eurocrypt17)

What are Constraint-Hiding CPRFs:
A simulation-based definition of CHCPRF

Where to find them:
Simulation-based 1-key CHCPRFs for NC1 from
Learning With Errors

How to use them:
- 1-key CHCPRF implies 1-key private-key
  functional encryption (reusable garbled circuits)
- 2-key CHCPRF implies obfuscation*
Plan for the talk:

Part 1: Definition, relation to obfuscation, functional encryption

Part 2: How to construct CHCPRFs, more on GGH15 mmaps
Defining constraint-hiding constraint PRF (CHCPRF)
Master_KeyGen -> MSK

definition of CHCPRF
Master_KeyGen -> MSK
Cons(MSK, C) -> K[C]
Master_KeyGen -> MSK
Cons(MSK, C) -> K[C]
Eval(K, x) -> F_K(x)
Simulation-based CHCPRF [CC 17]

for all p.p.t. adv, there’s a simulator, such that the outputs of the real and simulated distributions are indistinguishable.
Simulation-based definition of CHCPRF

\[ \text{Master KeyGen} \rightarrow \text{MSK} \]
\[ \text{Cons(MSK, C)} \rightarrow K[C] \]
\[ \text{Eval}(K, x) \rightarrow F_K(x) \]
Master _KeyGen_ $\rightarrow$ MSK
Cons(MSK, C) $\rightarrow$ K[C]
Eval(K, x) $\rightarrow$ $F_K(x)$

Simulation-based definition of CHCPRF
Master KeyGen $\rightarrow$ MSK
Cons(MSK, C) $\rightarrow$ K[C]
Eval(K, x) $\rightarrow$ $F_K(x)$

Simulation-based definition of CHCPRF

Real
adv
MSK
Master KeyGen
Constraint query C
MSK$^S$

Simulator
Master\_KeyGen \rightarrow MSK
Cons(MSK, C) \rightarrow K[C]
Eval(K, x) \rightarrow F_K(x)

Simulation-based definition of CHCPRF
Simulation-based definition of CHCPRF

Master KeyGen $\rightarrow$ MSK
Cons(MSK, C) $\rightarrow$ K[C]
Eval(K, x) $\rightarrow$ $F_K(x)$
Master KeyGen -> MSK
Cons(MSK, C) -> K[C]
Eval(K, x) -> F_K(x)

Simulation-based definition of CHCPRF

Real
adv
Simulator
Master_KeyGen $\rightarrow$ MSK
Cons(MSK, C) $\rightarrow$ K[C]
Eval(K, x) $\rightarrow$ $F_K(x)$

Simulation-based definition of CHCPRF

- MSK
  - Cons(MSK, C) $\rightarrow$ K[C]
  - Eval(MSK, x) $\rightarrow$ $F_K(x)$
- Master KeyGen
  - Constraint query C
  - Input query x
- MSK$^S$
  - $K^S \leftarrow \text{Sim}(MSK^S, 1^{\|C\|})$
  - $y^S \leftarrow \text{Sim}(MSK^S, x, C(x))$

Real
adv
Simulator
Correctness: for x s.t. C(x)=1, \( \Pr \left[ F_K(x) = F_{K[C]}(x) \right] > 1\text{-negl.} \)

Simulation-based definition of CHCPRF

**MSK**

\( \text{Cons}(\text{MSK}, C) \rightarrow K[C] \)

\( \text{ Eval}(\text{MSK}, x) \rightarrow F_K(x) \)

**Real**

**Master KeyGen**

\( \text{Constraint query } C \)

\( \text{ Input query } x \)

**adv**

**MSK}^S\)

\( K^S \leftarrow \text{Sim}(\text{MSK}^S, 1^{\left| C \right|}) \)

\( y^S \leftarrow \text{Sim}(\text{MSK}^S, x, C(x)) \)

**Simulator**
Correctness: for $x$ s.t. $C(x)=1$, $\Pr \left[ F_K(x) = F_{K[C]}(x) \right] > 1$-negl.

Pseudorandom & Constraint-hiding:

$$K[C], F_K(x) \approx_c K^S, y^S$$

(when $C(x)=0$, $y^S$ is from random)

---

**MSK**

$\text{Cons}(\text{MSK}, C) \rightarrow K[C]$

$\text{Eval}(\text{MSK}, x) \rightarrow F_K(x)$

**Master KeyGen**

Constraint query $C$

Input query $x$

**MSK^S**

$K^S \leftarrow \text{Sim}(\text{MSK}^S, 1^{|C|})$

$y^S \leftarrow \text{Sim}(\text{MSK}^S, x, C(x))$

---

Real

Simulation-based definition of CHCPRF

Simulator

adv
Theorem [CC17]
For 1-constrained key in the selective setting
sim-based $=$ ind-based
Sim-based definition for many constrained keys

Real

Simulator
Correctness: for x s.t. C(x)=1, Pr [ F_{K}(x) = F_{K[C]}(x) ] > 1-negl.

Constraint-hiding: \[ K[C_1], K[C_2] \approx_c K_1^S, K_2^S \]

\[ \text{MSK} \]
\[ \text{Cons(MSK, C_1)} \rightarrow K[C_1] \]
\[ \text{Cons(MSK, C_2)} \rightarrow K[C_2] \]

\[ \text{Master KeyGen} \]
\[ \text{Constraint query } C_1 \]
\[ \text{Constraint query } C_2 \]

\[ \text{MSK}^S \]
\[ K_1^S \leftarrow \text{Sim}(\text{MSK}^S, 1^{|C|}) \]
\[ K_2^S \leftarrow \text{Sim}(\text{MSK}^S, 1^{|C|}) \]

\[ \text{Real} \]
\[ \text{Simulator} \]

Sim-based definition for many keys
Correctness: for $x$ s.t. $C(x)=1$, $\Pr [ F_{K}(x) = F_{K[C]}(x) ] > 1\text{-negl.}$

Constraint-hiding: $K[C_1], K[C_2] \approx_{c} K_1^S, K_2^S$

---

**MSK**

Cons(MSK, $C_1$) -> $K[C_1]$

Cons(MSK, $C_2$) -> $K[C_2]$

---

**Master KeyGen**

Constraint query $C_1$

Constraint query $C_2$

---

**MSK_S**

$K_1^S$ <- Sim(MSK_S, $1^{|C|}$)

$K_2^S$ <- Sim(MSK_S, $1^{|C|}$)

---

Real

Simulator

Relaxed Sim-based definition for many keys
Reminiscent of obfuscation ...

Hide the program in the constraint
Theorem [CC 17]: Two-key CHCPRF (for function class C) implies obfuscation (for C)
- Two-key relaxed sim-CHCPRF implies strong VBB obfuscation
- Two-key ind-CHCPRF implies iO
Theorem [ CC 17 ]: Two-key CHCPRF (for function class C) implies obfuscation (for C)
- Two-key relaxed sim-CHCPRF implies strong VBB obfuscation
- Two-key ind-CHCPRF implies iO

Construction:

\[ \text{Obf} = ( K[C], K[Z] ) \]

\[ \text{Eval}(x): \text{check consistency} \]
\[ \text{Eval}( K[C], x) =? \text{Eval}( K[Z], x) \]

Idea implicit from the [GGHRSW13] candidate obfuscation
In the rest of the talk, we will focus on:

1-key simulation-based definition for CHCPRF.
CHCPRF => Functional encryption

Decrypt and eval
Theorem [ CC 17 ] 1-key sim-based CHCPRF implies 1-key private-key functional encryption (reusable garbled circuits).
Theorem [ CC 17 ] 1-key sim-based CHCPRF implies 1-key private-key functional encryption (reusable garbled circuits).

Construction: from normal encryption Sym and CHCPRF F

Enc(m;r): \[ ct = \text{Enc}_{\text{Sym.K}}(m;r); \quad \text{tag} = F[K](ct) \]

FSK[Sym.K, F.K, C]: constrained key for the “decryption and eval” functionality C(Dec_{Sym.K}( . ))

Eval: compute \( F[C(\text{Dec}_{\text{Sym.K}}( . ))](ct) \), and compare with tag
Main construction:
1-key sim-based CHCPRFs for NC1 from *Learning With Errors*, based on the multilinear maps by Gentry, Gorbunov, Halevi (GGH15)

Combine:
- Lattices-based PRFs
- Barrington’s theorem to embed functionality
- GGH15 encoding to provide a public constrained mode

Demonstrate a proof methodology of GGH15-based applications.
Short intro to BPR12

[Banerjee, Peikert, Rosen 12]
-- the first LWE-based PRF
Learning with errors

\[ Y = S \cdot A + E \mod q \]

- **A** is \( n \)-by-\( m \) in \( \mathbb{Z}_q \) (\( n \) is the lattice dimension, \( m > n \log q \))
- **Search LWE**: Given \( A, y = sA + E \), find \( s \)
- **Decisional LWE**: distinguish \( y \) from random

As hard as worst-case approx-SIVP (Quantumly) [Regev 05]
(classically for subexponential \( q \)) [Peikert 09, BLPRS 13]
Learning with errors

\[ A \text{ is } n\text{-by-}m \text{ in } \mathbb{Z}_q \quad \text{(n is the lattice dimension, } m > n \log q) \]

Search LWE: Given A, y = sA + E, find s
Decisional LWE: distinguish y from random
As hard as worst-case approx-SIVP (Quantumly) [Regev 05]
(classically for subexponential q) [Peikert 09, BLPRS 13]
Entries of S are small (e.g. from the error distribution)
As hard as normal LWE [Applebaum, Cash, Peikert, Sahai 09]
Banerjee, Peikert, Rosen ‘12
Subset-product & rounding

Key:

\[
\begin{align*}
S_{1,1} & \quad S_{2,1} & \quad \ldots & \quad S_{n,1} \\
S_{1,0} & \quad S_{2,0} & \quad \ldots & \quad S_{n,0}
\end{align*}
\]

Eval:

\[
F(x) = \left\{ \prod_{i=1}^{n} s_{i,x_i} \right\}_2 A
\]

\[
S_{i,b}
\]

are LWE secrets from low-norm distributions

mod q
Rounding: $\{t\}_p : \mathbb{Z}_q \rightarrow \mathbb{Z}_p$

Compute $t*p/q$, then round to the nearest integer

In this talk, $p=2$, $q/p > \exp(L)$, $q/p \sim$ super-polynomial

Amount of noise
Main observation: After rounding, can inject noises without changing functionality whp.

\[ F(x) = \{ \prod_{i} s_{i,x_i} A \}_2 \]
F(0110) = \{ s_{1,0} s_{2,1} s_{3,1} s_{4,0} A \}_2

F(x) = \{ \prod s_{i,xi} A \}_2

\text{Uniform Small Unspecified}

A \text{ is public, } S_{i,xi} \text{ are secret}

mod q
F(x) = \{ \prod_{i, x_i} A \}_2

F(0110) = \{ s_{1,0} s_{2,1} s_{3,1} s_{4,0} A \}_2

\approx \{ s_{1,0} s_{2,1} s_{3,1} (s_{4,0} A + E_{4,0}) \}_2

A is public, S_{i, x_i} are secret

Uniform Small Unspecified

A mod q

Banerjee, Peikert, Rosen 12

Proof of pseudorandomness
\[ F(x) = \{ \prod_{i} s_{i,xi} A \}_2 \]

\[ F(0110) = \{ s_{1,0}s_{2,1}s_{3,1}s_{4,0} A \}_2 \approx \{ s_{1,0}s_{2,1}s_{3,1}(s_{4,0} A+E_{4,0}) \}_2 \approx \{ s_{1,0}s_{2,1}s_{3,1}Y^{**0} \}_2 \]

- \( A \) is public, \( s_{i,xi} \) are secret
- \( S_i \) is the secret
- \( \text{mod } q \)
Banerjee, Peikert, Rosen 12
Proof of pseudorandomness

\[ F(x) = \prod_{i} s_{i,x_i} A \]

\[ F(0110) = \{ s_{1,0} s_{2,1} s_{3,1} s_{4,0} A \}_2 \]

\[ \approx \{ s_{1,0} s_{2,1} s_{3,1} (s_{4,0} A + E_{4,0}) \}_2 \]

\[ \approx \{ s_{1,0} s_{2,1} s_{3,1} Y^{**0} \}_2 \]

\[ \approx \{ s_{1,0} s_{2,1} (s_{3,1} Y^{**0} + E_{3,1}) \}_2 \]

A is public, \( S_{i,x_i} \) are secret.

**Uniform** Small **Unspecified**

**A**
F(x) = \{ \prod_{i,xi} A \}_2

F(0110) = \{ s_{1,0} s_{2,1} s_{3,1} s_{4,0} A \}_2
\approx \{ s_{1,0} s_{2,1} s_{3,1} s_{4,0} (s_{4,0} A + E_{4,0}) \}_2
\approx \{ s_{1,0} s_{2,1} s_{3,1} Y^{**0} \}_2
\approx \{ s_{1,0} s_{2,1} (s_{3,1} Y^{***0} + E_{3,1}) \}_2
\approx \{ s_{1,0} s_{2,1} Y^{*10} \}_2
\approx \{ Y_{0110} \}_2

\text{Uniform Small Unspecified}

A \text{ is public, } S_{i,xi} \text{ are secret}

\text{mod } q
Banerjee, Peikert, Rosen '12
Subset-product & rounding

Key:

\[
\begin{array}{cccc}
S_{1,1} & S_{2,1} & \cdots & S_{n,1} \\
S_{1,0} & S_{2,0} & \cdots & S_{n,0} \\
\end{array}
\]

Mod q

Eval:

\[ F(x) = \{ \prod_{i} S_{i,x_i} A \}_2 \]

What we need in addition to build a CHCPRF:

+ Embed \textcolor{red}{\textbf{structures}} in the secret terms to perform functionality (Barrington’s theorem)
+ A proper \textcolor{red}{\textbf{public mode}} of the function (GGH15 encoding)
Barrington’s theorem
(used to embed a circuit into the key)
Barrington 1986: log-depth boolean circuits can be recognized by subset products of permutation matrices of width 5.

Example: how to represent an AND gate

Our construction only work for certain representation of Barrington (e.g. $S_5$)
Barrington 1986: log-depth boolean circuits can be recognized by subset products of permutation matrices of width 5.

Example: how to represent an AND gate 0 and 0

Our construction only work for certain representation of Barrington (e.g. $S_5$)
Barrington 1986: log-depth boolean circuits can be recognized by subset products of permutation matrices of width 5.

Example: how to represent an AND gate

Our construction only work for certain representation of Barrington (e.g. $S_5$)
Barrington 1986: log-depth boolean circuits can be recognized by subset products of permutation matrices of width 5.

Example: how to represent an AND gate \(1 \text{ and } 0\)

Our construction only works for certain representations of Barrington (e.g. \(S_5\))
Barrington 1986: log-depth boolean circuits can be recognized by subset products of permutation matrices of width 5.

Example: how to represent an AND gate

\[ P \quad Q \quad P^{-1} \quad Q^{-1} = C \neq I \]

Our construction only work for certain representation of Barrington (e.g. \( S_5 \))
Representation of the constraint predicate: branching program

1 \quad B_{1,1} \quad B_{2,1} \quad B_{3,1} \ldots \quad B_{L,1}

0 \quad B_{1,0} \quad B_{2,0} \quad B_{3,0} \ldots \quad B_{L,0}

Steps 1 2 3 ... L

Eval: \quad \prod B_{z(i),x_{z(i)}} = I \text{ or } C
We set the secrets like:

\[ \mathbf{B}_{i,b} \otimes \mathbf{s}_{i,b} \]

Representation of secrets (to be encoded by GGH15): \[ \mathbf{B}_{i,b} \otimes \mathbf{s}_{i,b} \]

e.g. \[ \mathbf{I} \otimes \mathbf{s} = \]

\[ \mathbf{P} \otimes \mathbf{s} = \]
GGH15 encoding
[Gentry, Gorbunov, Halevi 15]
GGH15 “graph-induced multilinear maps”

Multilinear maps perspective:

\[ g, g^{S_1}, \ldots, g^{S_k}, g^{\prod S} \]

Normal hard group for DLOG

\[ A, S_1 A + E_1, \ldots, S_k A + E_k, \prod S A + E \]

GGH15: (Ring)LWE analogy

The “plaintexts” are encoded in the secret terms of LWE
Trapdoor [Ajtai 99, Alwen, Peikert 09, Micciancio, Peikert 12] Can sample A with a trapdoor T.

Can sample small preimage from Gaussian [Klein ‘00, GPV’08]
GGH15 encoding for the $i^{th}$ hop:

$A_i$

$S_{i,0}$

$S_{i,1}$

$A_{i+1}$
GGH15 encoding for the $i^{\text{th}}$ hop:

Encode($s_{i,b}$): 2 steps

1. $Y_{i,b} = s_{i,b} A_{i+1} + E_{i,b}$
GGH15 encoding for the $i^{th}$ hop:

Encode($s_{i,b}$): 2 steps
1. $Y_{i,b} = s_{i,b} A_{i+1} + E_{i,b}$
2. Sample (by the trapdoor of $A_i$) small $D_{i,b}$ s.t. $A_i D_{i,b} = Y_{i,b}$

$D_{i,b} = \text{Encoding}(s_{i,b})$
GGH15 for L hops:
GGH15 for L hops:

Encode($s_{i,b}$): 2 steps
GGH15 for L hops:

Encode($s_{i,b}$): 2 steps

1. $Y_{i,b} = s_{i,b} A_{i+1} + E_{i,b}$

$Y_{1,1} = s_{1,1} A_2 + E_{1,1}$

$Y_{1,0} = s_{1,0} A_2 + E_{1,0}$

$Y_{L,1} = s_{L,1} A_{L+1} + E_{L,1}$

$Y_{L,0} = s_{L,0} A_{L+1} + E_{L,0}$
GGH15 for L hops:

Encode($s_{i,b}$): 2 steps
1. $Y_{i,b} = s_{i,b} A_{i+1} + E_{i,b}$
2. Sample (by the trapdoor of $A_i$) small $D_{i,b}$ s.t. $A_i D_{i,b} = Y_{i,b}$

Let $D_{i,b}$ be Encoding($s_{i,b}$)
GGH15 for L hops:

Review: What are public
Understanding the functionality of GGH15
Evaluation of GGH15 (prove by example):

\[
\text{Eval}(0110) = A_1 D_{1,0} D_{2,1} D_{3,1} D_{4,0}
\]
Evaluation of GGH15 (prove by example):

\[
\begin{pmatrix}
S_{1,1} & \text{A}_2 \\
S_{1,0} & \\
\end{pmatrix}
+ \begin{pmatrix}
E_{1,1} \\
E_{1,0} \\
\end{pmatrix}
\]

\[
\text{Eval}(0110) = A_1 D_{1,0} D_{2,1} D_{3,1} D_{4,0}
\]

\[
= (s_{1,0} A_2 + E_{1,0}) D_{2,1} D_{3,1} D_{4,0}
\]
Evaluation of GGH15 (prove by example):

Eval(0110)

= $A_1D_{1,0} D_{2,1} D_{3,1} D_{4,0}$
= $(s_{1,0} A_2 + E_{1,0})D_{2,1} D_{3,1} D_{4,0}$
= $s_{1,0} A_2 D_{2,1} D_{3,1} D_{4,0} + "small"$
Evaluation of GGH15 (prove by example):

Eval(0110)

= \ A_1 D_{1,0} D_{2,1} D_{3,1} D_{4,0} \\
= (s_{1,0} A_2 + E_{1,0}) D_{2,1} D_{3,1} D_{4,0} \\
= s_{1,0} A_2 D_{2,1} D_{3,1} D_{4,0} + "small" \\
= s_{1,0} (s_{2,1} A_3 + E_{2,1}) D_{3,1} D_{4,0} + "small"
Evaluation of GGH15 (prove by example):

Eval(0110) = \( A_1 D_{1,0} D_{2,1} D_{3,1} D_{4,0} \)
= \((s_{1,0} A_2 + E_{1,0})\)\( D_{2,1} D_{3,1} D_{4,0} \)
= \( s_{1,0} A_2 D_{2,1} D_{3,1} D_{4,0} + \text{“small”} \)
= \( s_{1,0} (s_{2,1} A_3 + E_{2,1}) D_{3,1} D_{4,0} + \text{“small”} \)
= \( s_{1,0} s_{2,1} A_3 D_{3,1} D_{4,0} + \text{“still small”} \)
Evaluation of GGH15 (prove by example):

\[
\text{Eval}(0110) = A_1 D_{1,0} D_{2,1} D_{3,1} D_{4,0} \\
= (s_{1,0} A_2 + E_{1,0}) D_{2,1} D_{3,1} D_{4,0} \\
= s_{1,0} A_2 D_{2,1} D_{3,1} D_{4,0} + \text{“small”} \\
= s_{1,0} (s_{2,1} A_3 + E_{2,1}) D_{3,1} D_{4,0} + \text{“small”} \\
= s_{1,0} s_{2,1} A_3 D_{3,1} D_{4,0} + \text{“still small”} \\
= s_{1,0} s_{2,1} s_{3,1} A_4 D_{4,0} + \text{“still smallish”}
\]
Evaluation of GGH15 (prove by example):

\[
\text{Eval}(0110) = A_1 D_{1,0} D_{2,1} D_{3,1} D_{4,0} = (s_{1,0} A_{2} + E_{1,0}) D_{2,1} D_{3,1} D_{4,0} = s_{1,0} A_{2} D_{2,1} D_{3,1} D_{4,0} + \text{"small"} = s_{1,0} (s_{2,1} A_{3} + E_{2,1}) D_{3,1} D_{4,0} + \text{"small"} = s_{1,0} s_{2,1} A_{3} D_{3,1} D_{4,0} + \text{"still small"} = s_{1,0} s_{2,1} s_{3,1} A_{4} D_{4,0} + \text{"still smallish"} = s_{1,0} s_{2,1} s_{3,1} s_{4,0} A_{5} + \text{"small"}
\]
Evaluation of GGH15 (prove by example):

Evaluate

\[ A_{5} \]

\[ + \text{"small"} \]

A_1

\[ S_{1,1} \quad S_{2,1} \quad S_{3,1} \quad S_{4,1} \]

\[ S_{1,0} \quad S_{2,0} \quad S_{3,0} \quad S_{4,0} \]

\[ \text{Evaluate} \]

\[ D_{1,1} \quad D_{2,1} \quad D_{3,1} \quad D_{4,1} \]

\[ D_{1,0} \quad D_{2,0} \quad D_{3,0} \quad D_{4,0} \]
CHCPRF for NC1 constraint
NC1-CHCPRF from GGH15

Master public key: $A_1 \ldots A_{L+1}$ ($L = \#\text{steps in BP}$)
Master secret key: trapdoors of $A_1 \ldots A_L$, $s_{1,0}, s_{1,1}, \ldots, s_{L,0}, s_{L,1}$, & $J$
NC1-CHCPRF from GGH15

Master public key: $A_1 \ldots A_{L+1}$ ($L = \#\text{steps in BP}$)

Master secret key: trapdoors of $A_1 \ldots A_L$, $s_{1,0}, s_{1,1}, \ldots, s_{L,0}, s_{L,1}, \& J$

Constrained key gen: let $S_{i,b} := B_{i,b} \otimes s_{i,b}$, sample GGH15 encodings for $S_{i,b}$

Eval: $F(x) = \{ J A_1 \prod D_{i,x_{z(i)}} \}_2$ ($z: [L] \rightarrow [n]$ is the step-to-input mapping)

Constrained key:
NC1-CHCPRF from GGH15

Master public key: $A_1 \ldots A_{L+1}$ (L = #steps in BP)

Master secret key: trapdoors of $A_1 \ldots A_L$, $s_{1,0}$, $s_{1,1}$, ..., $s_{L,0}$, $s_{L,1}$, & J

Constrained key gen: let $S_{i,b} := B_{i,b} \otimes s_{i,b}$, sample GGH15 encodings for $S_{i,b}$

Eval: $F(x) = \{ JA_1 \prod D_{i,x_z(i)} \}_2$ (z: [L]->[n] is the step-to-input mapping)

Functionality check:
when $C(x)=1$, $F(x) = \{ JA_1 \prod D_{i,x_z(i)} \}_2$
  $= \{ J( I \otimes \prod s_{i,x_z(i)} ) A_{L+1} + \text{small noise} \}_2$
  $\approx s \{ J( I \otimes \prod s_{i,x_z(i)} ) A_{L+1} \}_2$

when $C(x)=0$, $F(x) = \{ JA_1 \prod D_{i,x_z(i)} \}_2$
  $= \{ J( C \otimes \prod s_{i,x_z(i)} ) A_{L+1} + \text{small noise} \}_2$
  $\approx s \{ J( C \otimes \prod s_{i,x_z(i)} ) A_{L+1} \}_2$
NC1-CHCPRF from GGH15 *

Compare to GGM
NC1-CHCPRF from GGH15

Compare to GGM

Our CHCPRF

original

fresh random
NC1-CHCPRF from GGH15 *

Example: $C(x) = 0$ iff $x_1 = x_2 = 1$ query $x = 11$

What are we trying to simulate?

Uniform Small Unspecified

$s_{i,xi}$ are secret, $A_i$, $D_{i,xi}$ are public

$$\{ I^\otimes(s_{1,1}s_{2,1}s_{3,1}s_{4,1}) A_5 \}_{2}$$

Real

Simulator
NC1-CHCPRF from GGH15 *

Example: $C(x) = 0$ iff $x_1 = x_2 = 1$ query $x = 11$

Proof by example with 1 input query
NC1-CHCPRF from GGH15 *

Example: $C(x) = 0$ iff $x_1 = x_2 = 1$ query $x = 11$

$s_{i,x_i}$ are secret, $A_i$, $D_{i,x_i}$ are public

$$\text{Eval}(11) = \{ I \otimes (s_{1,1}s_{2,1}s_{3,1}s_{4,1})A_5 \}_2$$
NC1-CHCPRF from GGH15 *

Example: $C(x)=0$ iff $x_1=x_2=1$ query $x=11$

$s_{i,xi}$ are secret, $A_i$, $D_{i,xi}$ are public

$$\text{Eval}(11) = \left\{ I \otimes (s_{1,1} s_{2,1} s_{3,1} s_{4,1}) A_5 \right\}_2$$

$$\approx \left\{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1}))((Q^{-1} \otimes s_{4,1}) A_5 + E_{4,1}) \right\}_2$$
Uniform Small Unspecified

$s_{i,x_i}$ are secret, $A_i$, $D_{i,x_i}$ are public

Example: $C(x) = 0$ iff $x_1 = x_2 = 1$ query $x = 11$

Perm-LWE
NC1-CHCPRF from GGH15 *

Example: $C(x) = 0$ iff $x_1 = x_2 = 1$ query $x = 11$

\[
\text{Eval}(11) = \{ I \otimes (s_{1,1} s_{2,1} s_{3,1} s_{4,1}) A_5 \}_2 \\
\approx_s \{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1}))((Q^{-1} \otimes s_{4,1}) A_5 + E_{4,1}) \}_2 \\
\approx_c \{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1})) A_4 D_{4,1} \}_2
\]
NC1-CHCPRF from GGH15 *

Example: $C(x)=0$ iff $x_1=x_2=1$  query $x=11$

$s, x_i$ are secret, $A, D, i, x_i$ are public

$$Y_{3,1} = (P^{-1} \otimes s_{3,1})A_4 + E_{3,1}$$

$$Y_{3,0} = (I \otimes s_{3,0})A_4 + E_{3,0}$$

$$\text{Eval}(11) = \{ I \otimes (s_{1,1} s_{2,1} s_{3,1} s_{4,1})A_5 \}_2$$

$$\approx_s \{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1}))((Q^{-1} \otimes s_{4,1})A_5 + E_{4,1}) \}_2$$

$$\approx_c \{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1}))A_4 D_{4,1} \}_2$$

$$\approx_s \{ (QP \otimes (s_{1,1} s_{2,1}))((P^{-1} \otimes s_{3,1})A_4 + E_{3,1})D_{4,1} \}_2$$
Perm-LWE + GPV
NC1-CHCPRF from GGH15 *

Example: $C(x) = 0$ iff $x_1 = x_2 = 1$  query $x = 11$

$\text{Eval}(11) = \{ I \otimes (s_{1,1} s_{2,1} s_{3,1} s_{4,1}) A_5 \}_2$

$\approx_s \{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1}))((Q^{-1} \otimes s_{4,1}) A_5 + E_{4,1}) \}_2$

$\approx_c \{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1})) A_4 D_{4,1} \}_2$

$\approx_s \{ (Q P \otimes (s_{1,1} s_{2,1}))((P^{-1} \otimes s_{3,1}) A_4 + E_{3,1}) D_{4,1} \}_2$

$\approx_c \{ (Q P \otimes (s_{1,1} s_{2,1})) A_3 D_{3,1} D_{4,1} \}_2$

$\approx_c \ldots \approx_c \{ C^{-1} A_1 \prod D_{z(x), x_z(x)} \}_2$

CK done, pseudorandomness of the output still not
NC1-CHCPRF from GGH15 *

Example: $C(x) = 0$ iff $x_1 = x_2 = 1$ query $x = 11$

$$
\text{Eval}(11) = \left\{ I \otimes (s_{1,1} s_{2,1} s_{3,1} s_{4,1}) A_5 \right\}_2 \\
\approx_s \left\{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1}))((Q^{-1} \otimes s_{4,1}) A_5 + E_{4,1}) \right\}_2 \\
\approx_c \left\{ (Q \otimes (s_{1,1} s_{2,1} s_{3,1})) A_4 D_{4,1} \right\}_2 \\
\approx_s \left\{ (QP \otimes (s_{1,1} s_{2,1}))(P^{-1} \otimes s_{3,1}) A_4 + E_{3,1} \right\} D_{4,1} \right\}_2 \\
\approx_c \left\{ (QP \otimes (s_{1,1} s_{2,1})) A_3 D_{3,1} D_{4,1} \right\}_2 \\
\approx_c \ldots \approx_c \left\{ C^{-1} A_1 \prod D_{z(x), x_{z(x)}} \right\}_2 \\
\text{Current status:}
- CK ✓
- randomness of the outputs ✗
NC1-CHCPRF from GGH15 *

Example: $C(x) = 0$ iff $x_1 = x_2 = 1$ query $x = 11$

$$\text{Eval}(11) = \{ I \otimes (s_{1,1}s_{2,1}s_{3,1}s_{4,1})A_5 \}_2$$
$$\approx_s \{ (Q \otimes (s_{1,1}s_{2,1}s_{3,1}))((Q^{-1} \otimes s_{4,1})A_5 + E_{4,1}) \}_2$$
$$\approx_c \{ (Q \otimes (s_{1,1}s_{2,1}s_{3,1}))A_4D_{4,1} \}_2$$
$$\approx_s \{ (QP \otimes (s_{1,1}s_{2,1}))((P^{-1} \otimes s_{3,1})A_4 + E_{3,1})D_{4,1} \}_2$$
$$\approx_c \{ (QP \otimes (s_{1,1}s_{2,1}))A_3D_{3,1}D_{4,1} \}_2$$
$$\approx_c \cdots \approx_c \{ JC^{-1}A_1 \prod D_{z(x),x,z(x)} \}_2$$

Current status:
- CK ✓
- randomness of the outputs ✗

Solution:
Multiply a random vector $J$ on the left

$s_{i,\xi_i}$ are secret, $A_i$, $D_{i,\xi_i}$ are public
\[
\text{Real} \{ \text{e}^{(x)\sum_{d}x(x)^{z}}} \text{D} \} \approx \text{Eval} = \cdots
\]

\[
\{ (x)z^{x} \}^c \text{D} \text{A}_{\text{I}+1} \text{E} \}
\]
\[
\text{Real} \quad \{ J(I \otimes (\prod s_{z(x),x_{z(x)}}))A_{L+1} \}_2
\]

\[
\text{Simulator} \quad \{ \text{Uniform} \}_2
\]

\[
\text{Eval} = \ldots
\]

\[
\approx_c \{ (JC^{-1}A_{1} + E) \prod D_{z(x),x_{z(x)}} \}_2
\]

\[
\approx_c \{ \text{Uniform} \}_2
\]
Summary: NC1 CHCPRF from GGH15
- Constraint hiding: Perm-LWE + GPV
- Outputs: need additional protection J, justified by JLWE
Concurrent work:
Boneh, Kim, Montgomery (Eurocrypt 17)

1-key puncturable CHCPRFs from LWE.

Both root from previous lattices-based PRFs, but different method to constrain and hide.
Genealogy of Lattices-based PRFs

[BPR12] -- the settler
[BLMR13] -- key homomorphic
*[BP14] -- better key homomorphic, embed a tree
*[BFPPS15] -- [BP14] is puncturable
*[BV15] -- embed a circuit, constrained for P
*[BKM17] -- puncture privately, built from [BV15]
[CC17] -- constrained privately for NC1, influenced by GGH15 mmaps

* uses gadget matrix G, adapted from the lattices-based FHE, ABE, PE

Q: Is there a transformation between Dual-Regev-based homomorphic schemes and GGH15-based ones?

p.s. Hoeteck asked me if there’s an interpretation of [GVW13] ABE from [GGH15]. I thought for a little bit, not obvious.
More questions of GGH15

Q: What safe modes do we have confidence for GGH15?  
A: With limited number/restricted form of zeros, very likely.

Q: What is weird about GGH15 (as a useful mmaps)?  
A: Must prove from 1 direction (namely make sure that the trapdoor sampling is safe, from sink to source), not a desirable property of mmaps.

Q: Anything to say when the A matrices are hidden?  
A: There must be something to say … a question worth to understand
The end