Fine-Grained Cryptography

Akshay Degwekar   Vinod Vaikuntanathan   Prashant Nalini Vasudevan

MIT

June 9, 2016

Cryptography

" *Cryptographers seldom sleep well.* – Joe Kilian, several years ago"

– Alon Rosen, yesterday at lunch

Cryptography

" *Cryptographers seldom sleep well.* – Joe Kilian, several years ago"

– Alon Rosen, yesterday at lunch

Present-day cryptography employs several hardness assumptions.

- Factoring.
- Lattice problems.
- ...

Cryptography

“ *Cryptographers seldom sleep well.* – Joe Kilian, several years ago”

– Alon Rosen, yesterday at lunch

Present-day cryptography employs several hardness assumptions.

- Factoring.
- Lattice problems.
- ...

What are the weakest assumptions we can do with?

- We need BPP $\neq$ NP. Is this sufficient? ([AGGM06], [BB15], ...)
- How about BPP $\neq$ SZK? ([Ost91], [AR15], ...)

Cryptography

“ *Cryptographers seldom sleep well.* – Joe Kilian, several years ago”

– Alon Rosen, yesterday at lunch

Present-day cryptography employs several hardness assumptions.

- Factoring.
- Lattice problems.
- ...

What are the weakest assumptions we can do with?

- We need BPP $\neq$ NP. Is this sufficient? ([AGGM06], [BB15], ...)
- How about BPP $\neq$ SZK? ([Ost91], [AR15], ...)

In what settings can we do with minimal assumptions?

- With *no* assumptions?

Cryptography in other settings

- [Mer78]: Public-key encryption in the random oracle model where honest parties run in time $O(n)$, *unconditionally* secure against adversaries that run in time $o(n^2)$.

Cryptography in other settings

- [Mer78]: Public-key encryption in the random oracle model where honest parties run in time $O(n)$, *unconditionally* secure against adversaries that run in time $o(n^2)$.

- [Mau92, CM97]: Symmetric encryption and key-exchange protocols where honest parties use $O(s)$ space, *unconditionally* secure against adversaries that use $o(s^2)$ space.

Cryptography in other settings

- [Mer78]: Public-key encryption in the random oracle model where honest parties run in time $O(n)$, *unconditionally* secure against adversaries that run in time $o(n^2)$.
- [Mau92, CM97]: Symmetric encryption and key-exchange protocols where honest parties use $O(s)$ space, *unconditionally* secure against adversaries that use $o(s^2)$ space.
- [Has87]: One-way permutation computable in NC^0, *unconditionally* secure against adversaries computable in AC^0.

Cryptography in other settings

- [Mer78]: Public-key encryption in the random oracle model where honest parties run in time $O(n)$, *unconditionally* secure against adversaries that run in time $o(n^2)$.

- [Mau92, CM97]: Symmetric encryption and key-exchange protocols where honest parties use $O(s)$ space, *unconditionally* secure against adversaries that use $o(s^2)$ space.

- [Has87]: One-way permutation computable in NC^0, *unconditionally* secure against adversaries computable in AC^0.

$$f(x_1, \ldots, x_n) = (x_1 \oplus x_2, x_2 \oplus x_3, \ldots, x_{n-1} \oplus x_n, x_n)$$

Fine-Grained Cryptographic Primitives

Primitives that are:

1. Secure against adversaries with restricted computational power.
2. Computable with less computational power than these adversaries.

Fine-Grained Cryptographic Primitives

Primitives that are:

1. Secure against adversaries with restricted computational power.
2. Computable with less computational power than these adversaries.

Examples of restrictions:

- Bounded running time. (E.g., Merkle Puzzles [Mer78].)

Fine-Grained Cryptographic Primitives

Primitives that are:

1. Secure against adversaries with restricted computational power.
2. Computable with less computational power than these adversaries.

Examples of restrictions:

- Bounded running time. (E.g., Merkle Puzzles [Mer78].)
- Bounded space. (E.g., the bounded storage model from [Mau92] and [CM97].)

Fine-Grained Cryptographic Primitives

Primitives that are:

1. Secure against adversaries with restricted computational power.
2. Computable with less computational power than these adversaries.

Examples of restrictions:

- Bounded running time. (E.g., Merkle Puzzles [Mer78].)
- Bounded space. (E.g., the bounded storage model from [Mau92] and [CM97].)
- Bounded circuit-depth. (E.g., [Has87], this work.)

Fine-Grained Cryptographic Primitives

Primitives that are:

1. Secure against adversaries with restricted computational power.
2. Computable with less computational power than these adversaries.

Examples of restrictions:

- Bounded running time. (E.g., Merkle Puzzles [Mer78].)
- Bounded space. (E.g., the bounded storage model from [Mau92] and [CM97].)
- Bounded circuit-depth. (E.g., [Has87], this work.)
  - Constant depth, unbounded fan-in - AC^0.
  - Logarithmic depth, bounded fan-in - NC^1.

Results

Unconditional constructions against AC^0:

- OWF, PRG. (other constructions known from [Has87, AW85, Vio12, MST06])
- Weak PRF.
- Symmetric Encryption.
- Collision Resistant Hash Functions.

Constructions against NC^1 based on $\mathsf{L} \not\subseteq \mathsf{NC}^1$:

- OWF, PRG. (similar, independent, constructions in [AR15])
- Public-Key Encryption.
- Collision Resistant Hash Functions.

Results

Unconditional constructions against AC^0:

- OWF, PRG. (other constructions known from [Has87, AW85, Vio12, MST06])
- Weak PRF.
- Symmetric Encryption.
- Collision Resistant Hash Functions.

Constructions against NC^1 based on $L \not\subseteq NC^1$:

- OWF, PRG. (similar, independent, constructions in [AR15])
- Public-Key Encryption.
- Collision Resistant Hash Functions.

Ingredients

Ingredients

Theorem ([Bra10, Tal14])

Polynomial-sized circuits of depth d cannot distinguish between a $\log^{4d}(m)$-wise independent distribution over $\{0,1\}^m$ and U_m.

Ingredients

Theorem ([Bra10, Tal14])

Polynomial-sized circuits of depth d cannot distinguish between a $\log^{4d}(m)$-wise independent distribution over $\{0,1\}^m$ and U_m.

Corollary

Let D_m be a distribution over $\{0,1\}^m$ that is $\log^{\omega(1)}(m)$-wise independent. Then:

$$D_m \approx_{\mathrm{AC}^0} U_m$$

Ingredients

Observation

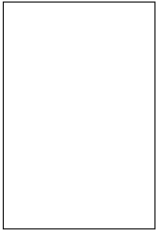
Let $\mathbf{M} \in \{0,1\}^{m \times n}$ *be a matrix such that any set of* k *rows are linearly independent. If* $\mathbf{x}$ *is distributed uniformly over* $\{0,1\}^n$*, then* $\mathbf{Mx}$ *is* k*-wise independent.*

Ingredients

Observation

Let $\mathbf{M} \in \{0,1\}^{m \times n}$ *be a matrix such that any set of* k *rows are linearly independent. If* $\mathbf{x}$ *is distributed uniformly over* $\{0,1\}^n$*, then* $\mathbf{Mx}$ *is* k*-wise independent.*

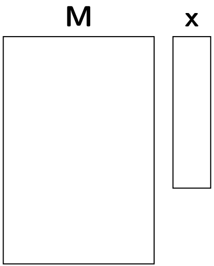
$\mathbf{M}$



Ingredients

Observation

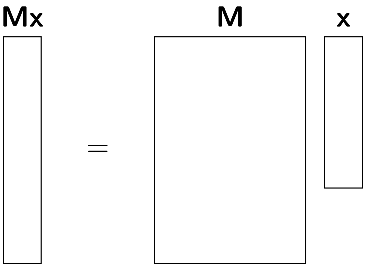
Let $\mathbf{M} \in \{0,1\}^{m \times n}$ *be a matrix such that any set of* k *rows are linearly independent. If* $\mathbf{x}$ *is distributed uniformly over* $\{0,1\}^n$*, then* $\mathbf{Mx}$ *is* k*-wise independent.*



Ingredients

Observation

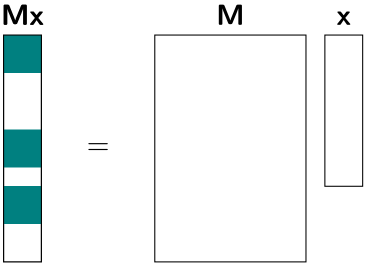
Let $\mathbf{M} \in \{0,1\}^{m \times n}$ *be a matrix such that any set of* k *rows are linearly independent. If* $\mathbf{x}$ *is distributed uniformly over* $\{0,1\}^n$*, then* $\mathbf{Mx}$ *is* k*-wise independent.*



Ingredients

Observation

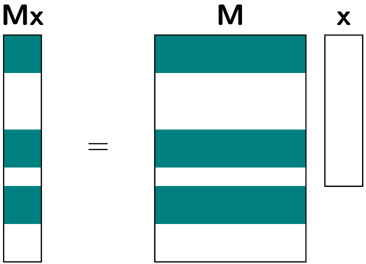
Let $\mathbf{M} \in \{0,1\}^{m \times n}$ *be a matrix such that any set of* k *rows are linearly independent. If* $\mathbf{x}$ *is distributed uniformly over* $\{0,1\}^n$*, then* $\mathbf{Mx}$ *is* k*-wise independent.*



Ingredients

Observation

Let $\mathbf{M} \in \{0,1\}^{m \times n}$ *be a matrix such that any set of* k *rows are linearly independent. If* $\mathbf{x}$ *is distributed uniformly over* $\{0,1\}^n$*, then* $\mathbf{Mx}$ *is* k*-wise independent.*



Ingredients

Lemma ([Gal62])

If rows of $\mathbf{M}_{m \times n}$ *are chosen to be random sparse vectors, and* $m = \mathrm{poly}(n)$*, then w.h.p. any set of* $\left(\frac{n}{\log^3(n)}\right)$ *rows of* $\mathbf{M}$ *are linearly independent.*

Ingredients

Lemma ([Gal62])

If rows of $\mathbf{M}_{m\times n}$ *are chosen to be random sparse vectors, and* $m = \mathrm{poly}(n)$*, then w.h.p. any set of* $\left(\frac{n}{\log^3(n)}\right)$ *rows of* $\mathbf{M}$ *are linearly independent.*

- $\implies$ *w.h.p.* $\mathbf{Mx}$ *is* $\left(\frac{n}{\log^3(n)}\right)$*-wise independent.*

Ingredients

Lemma ([Gal62])

If rows of $\mathbf{M}_{m\times n}$ *are chosen to be random sparse vectors, and* $m = \mathrm{poly}(n)$*, then w.h.p. any set of* $\left(\frac{n}{\log^3(n)}\right)$ *rows of* $\mathbf{M}$ *are linearly independent.*

- $\implies$ *w.h.p.* $\mathbf{Mx}$ *is* $\left(\frac{n}{\log^3(n)}\right)$*-wise independent.*
- $\implies (\mathbf{M}, \mathbf{Mx}) \approx_{\mathrm{AC}^0} (\mathbf{M}, U_m)$

Ingredients

Lemma (Sparse Matrix Lemma)

If rows of $\mathbf{M}_{m\times n}$ are chosen to be random sparse vectors, and $m = \mathrm{poly}(n)$, then w.h.p. any set of $\left(\frac{n}{\log^3(n)}\right)$ rows of $\mathbf{M}$ are linearly independent.

- $\implies$ *w.h.p.* $\mathbf{Mx}$ *is* $\left(\frac{n}{\log^3(n)}\right)$*-wise independent.*
- $\implies$ $(\mathbf{M}, \mathbf{Mx}) \approx_{\mathrm{AC}^0} (\mathbf{M}, U_m)$

Ingredients

Theorem (Implied by [AB84])

There is an AC^0 *circuit* C *such that for* $\mathbf{v_1}, \mathbf{v_2} \in \{0,1\}^n$*, if at least one of them is* $\log^2(n)$*-sparse, then* $C(\mathbf{v_1}, \mathbf{v_2}) = \langle \mathbf{v_1}, \mathbf{v_2} \rangle$*.*

Ingredients

Theorem (Implied by [AB84])

There is an AC^0 *circuit* C *such that for* $\mathbf{v_1}, \mathbf{v_2} \in \{0,1\}^n$*, if at least one of them is* $\log^2(n)$*-sparse, then* $C(\mathbf{v_1}, \mathbf{v_2}) = \langle \mathbf{v_1}, \mathbf{v_2} \rangle$*.*

Corollary

If rows of $\mathbf{M}$ *are sparse,* $\mathbf{Mx}$ *can be computed in* AC^0*.*

Symmetric Encryption against AC^0

Symmetric Encryption against AC^0

KeyGen(1^n) :

Random $\mathbf{k} \in \{0,1\}^n$.

Symmetric Encryption against AC^0

KeyGen(1^n) :

Random $\mathbf{k} \in \{0,1\}^n$.

Enc$(1^n, \mathbf{k}, b)$:

Random sparse $\mathbf{c} \in \{0,1\}^n$ such that $\langle \mathbf{c}, \mathbf{k} \rangle = b$.

Symmetric Encryption against AC^0

$\mathsf{KeyGen}(1^n)$:

Random $\mathbf{k} \in \{0,1\}^n$.

$\mathsf{Enc}(1^n, \mathbf{k}, b)$:

Random sparse $\mathbf{c} \in \{0,1\}^n$ such that $\langle \mathbf{c}, \mathbf{k} \rangle = b$.

$\mathsf{Dec}(1^n, \mathbf{k}, \mathbf{c})$:

Output $\langle \mathbf{c}, \mathbf{k} \rangle$.

Symmetric Encryption against AC^0

$\mathsf{KeyGen}(1^n)$:
  Random $\mathbf{k} \in \{0,1\}^n$.

$\mathsf{Enc}(1^n, \mathbf{k}, b)$:
  Random sparse $\mathbf{c} \in \{0,1\}^n$ such that $\langle \mathbf{c}, \mathbf{k} \rangle = b$.

$\mathsf{Dec}(1^n, \mathbf{k}, \mathbf{c})$:
  Output $\langle \mathbf{c}, \mathbf{k} \rangle$.

- Some keys are more equal than others.

Symmetric Encryption against AC^0

$\mathsf{KeyGen}(1^n)$:
  Random $\mathbf{k} \in \{0,1\}^n$.

$\mathsf{Enc}(1^n, \mathbf{k}, b)$:
  Random sparse $\mathbf{c} \in \{0,1\}^n$ such that $\langle \mathbf{c}, \mathbf{k} \rangle = b$.

$\mathsf{Dec}(1^n, \mathbf{k}, \mathbf{c})$:
  Output $\langle \mathbf{c}, \mathbf{k} \rangle$.

- Some keys are more equal than others.
- Above is additively homomorphic.

Symmetric Encryption against AC^0

Game 1 - Semantic Security

Ch **Adv**

$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$ $\quad \xleftarrow[(\lvert m_0 \rvert = \lvert m_1 \rvert = m)]{m_0,\ m_1}$

$b \leftarrow \{0, 1\}$ $\quad \xrightarrow{\mathsf{Enc}(\mathbf{k}, m_b)}$

$\xleftarrow{b'}$

Symmetric Encryption against AC^0

Game 1 - Semantic Security

Ch | **Adv**

$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$ $\quad \xleftarrow[(|m_0| = |m_1| = m)]{m_0,\ m_1}$

$b \leftarrow \{0,1\}$ $\quad \xrightarrow{\mathsf{Enc}(\mathbf{k}, m_b)}$

$\xleftarrow{b'}$

Game 2

Ch | **Adv**

$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$ $\quad \xrightarrow[\{\mathbf{c}_i^1 \leftarrow \mathsf{Enc}(\mathbf{k}, 1)\}_i]{\{\mathbf{c}_i^0 \leftarrow \mathsf{Enc}(\mathbf{k}, 0)\}_i}$

$b \leftarrow \{0,1\}$ $\quad \xrightarrow{\mathsf{Enc}(\mathbf{k}, b)}$

$\xleftarrow{b'}$

Symmetric Encryption against AC^0

Game 1 - Semantic Security

$$
\begin{array}{lcr}
\textbf{Ch} & & \textbf{Adv} \\
\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n) & \xleftarrow[(|m_0| = |m_1| = m)]{m_0,\ m_1} & \\
b \leftarrow \{0,1\} & \xrightarrow{\mathsf{Enc}(\mathbf{k}, m_b)} & \\
 & \xleftarrow{b'} &
\end{array}
$$

Game 2

$$
\begin{array}{lcr}
\textbf{Ch} & & \textbf{Adv} \\
\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n) & \xrightarrow[\left\{\mathbf{c}_i^1 \leftarrow \mathsf{Enc}(\mathbf{k}, 1)\right\}_i]{\left\{\mathbf{c}_i^0 \leftarrow \mathsf{Enc}(\mathbf{k}, 0)\right\}_i} & \\
b \leftarrow \{0,1\} & \xrightarrow{\mathsf{Enc}(\mathbf{k}, b)} & \\
 & \xleftarrow{b'} &
\end{array}
$$

Equivalent by standard hybrid arguments.

Symmetric Encryption against AC^0

Game 2

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$	$\xrightarrow[\{\mathbf{c}_i^1 \leftarrow \mathsf{Enc}(\mathbf{k},1)\}_i]{\{\mathbf{c}_i^0 \leftarrow \mathsf{Enc}(\mathbf{k},0)\}_i}$	
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathsf{Enc}(\mathbf{k},b)}$	
	$\xleftarrow{b'}$	

Symmetric Encryption against AC^0

Game 3

$$\begin{array}{lcr}
\textbf{Ch} & & \textbf{Adv} \\
\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n) & & \\
\{b_i \leftarrow \{0,1\}\}_i & \xrightarrow{\{(\mathsf{Enc}(\mathbf{k}, b_i), b_i)\}_i} & \\
b \leftarrow \{0,1\} & \xrightarrow{\mathsf{Enc}(\mathbf{k}, b)} & \\
& \xleftarrow{b'} &
\end{array}$$

Game 2

$$\begin{array}{lcr}
\textbf{Ch} & & \textbf{Adv} \\
\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n) & \xrightarrow[\{\mathbf{c}_i^1 \leftarrow \mathsf{Enc}(\mathbf{k}, 1)\}_i]{\{\mathbf{c}_i^0 \leftarrow \mathsf{Enc}(\mathbf{k}, 0)\}_i} & \\
b \leftarrow \{0,1\} & \xrightarrow{\mathsf{Enc}(\mathbf{k}, b)} & \\
& \xleftarrow{b'} &
\end{array}$$

Symmetric Encryption against AC^0

Game 3

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
$\{b_i \leftarrow \{0,1\}\}_i$	$\xrightarrow{\{(\mathsf{Enc}(\mathbf{k}, b_i), b_i)\}_i}$	
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathsf{Enc}(\mathbf{k}, b)}$	
	$\xleftarrow{b'}$	

Game 2

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$	$\xrightarrow[\{\mathbf{c}_i^1 \leftarrow \mathsf{Enc}(\mathbf{k}, 1)\}_i]{\{\mathbf{c}_i^0 \leftarrow \mathsf{Enc}(\mathbf{k}, 0)\}_i}$	
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathsf{Enc}(\mathbf{k}, b)}$	
	$\xleftarrow{b'}$	

Game 2 Challenger can be simulated by Game 3 Adversary in AC^0.

Symmetric Encryption against AC^0

Game 3

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
$\{b_i \leftarrow \{0,1\}\}_i$	$\xrightarrow{\{(\mathsf{Enc}(\mathbf{k}, b_i), b_i)\}_i}$	
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathsf{Enc}(\mathbf{k}, b)}$	
	$\xleftarrow{b'}$	

Symmetric Encryption against AC^0

Game 3

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
$\{b_i \leftarrow \{0,1\}\}_i$	$\xrightarrow{\{(\mathsf{Enc}(\mathbf{k}, b_i), b_i)\}_i}$	
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathsf{Enc}(\mathbf{k}, b)}$	
	$\xleftarrow{b'}$	

Game 4

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
sparse $\{\mathbf{r}_i\}_i$	$\xrightarrow{\{(\mathbf{r}_i, \langle \mathbf{k}, \mathbf{r_i} \rangle)\}_i}$	
sparse $\mathbf{r}$	$\xrightarrow{\mathbf{r}}$	
$b = \langle \mathbf{k}, \mathbf{r} \rangle$	$\xleftarrow{b'}$	

Symmetric Encryption against AC^0

Game 3

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
$\{b_i \leftarrow \{0,1\}\}_i$	$\xrightarrow{\{(\mathsf{Enc}(\mathbf{k}, b_i), b_i)\}_i}$	
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathsf{Enc}(\mathbf{k}, b)}$	
	$\xleftarrow{b'}$	

Game 4

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
sparse $\{\mathbf{r}_i\}_i$	$\xrightarrow{\{(\mathbf{r}_i, \langle \mathbf{k}, \mathbf{r_i} \rangle)\}_i}$	
sparse $\mathbf{r}$	$\xrightarrow{\mathbf{r}}$	
$b = \langle \mathbf{k}, \mathbf{r} \rangle$	$\xleftarrow{b'}$	

Reverse Sampling Lemma

For 'balanced' keys $\mathbf{k}$*,* $(\mathsf{Enc}(\mathbf{k}, b_i), b_i)$ *and* $(\mathbf{r}_i, \langle \mathbf{k}, \mathbf{r_i} \rangle)$ *are statistically close.*

Symmetric Encryption against AC^0

Game 4

Ch **Adv**

$\mathbf{k} \leftarrow \mathrm{KeyGen}(1^n)$

sparse $\{\mathbf{r}_i\}_i$ $\xrightarrow{\{(\mathbf{r}_i, \langle \mathbf{k}, \mathbf{r_i} \rangle)\}_i}$

sparse $\mathbf{r}$ $\xrightarrow{\mathbf{r}}$

$b = \langle \mathbf{k}, \mathbf{r} \rangle$ $\xleftarrow{b'}$

Symmetric Encryption against AC^0

Game 4

Ch		**Adv**
$\mathbf{k} \leftarrow \text{KeyGen}(1^n)$		
sparse $\{\mathbf{r}_i\}_i$	$\xrightarrow{\mathbf{R}, \mathbf{Rk}}$	
sparse $\mathbf{r}$	$\xrightarrow{\mathbf{r}}$	
$b = \langle \mathbf{k}, \mathbf{r} \rangle$	$\xleftarrow{b'}$	

Symmetric Encryption against AC^0

Game 5

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
sparse $\{\mathbf{r}_i\}_i$		
random $\mathbf{r}'$	$\xrightarrow{\mathbf{R}, \mathbf{r}'}$	
sparse $\mathbf{r}$	$\xrightarrow{\mathbf{r}}$	
$b = \langle \mathbf{k}, \mathbf{r} \rangle$	$\xleftarrow{b'}$	

Game 4

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
sparse $\{\mathbf{r}_i\}_i$	$\xrightarrow{\mathbf{R}, \mathbf{Rk}}$	
sparse $\mathbf{r}$	$\xrightarrow{\mathbf{r}}$	
$b = \langle \mathbf{k}, \mathbf{r} \rangle$	$\xleftarrow{b'}$	

Symmetric Encryption against AC^0

Game 5

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
sparse $\{\mathbf{r}_i\}_i$		
random $\mathbf{r}'$	$\xrightarrow{\mathbf{R}, \mathbf{r}'}$	
sparse $\mathbf{r}$	$\xrightarrow{\mathbf{r}}$	
$b = \langle \mathbf{k}, \mathbf{r} \rangle$	$\xleftarrow{b'}$	

Game 4

Ch		**Adv**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^n)$		
sparse $\{\mathbf{r}_i\}_i$	$\xrightarrow{\mathbf{R}, \mathbf{Rk}}$	
sparse $\mathbf{r}$	$\xrightarrow{\mathbf{r}}$	
$b = \langle \mathbf{k}, \mathbf{r} \rangle$	$\xleftarrow{b'}$	

Indistinguishable by Sparse Matrix Lemma.

CRHF against AC^0

$\mathsf{KeyGen}(1^n)$:

- $\mathbf{k} \leftarrow \mathsf{KeyGen}^{Enc}(1^{\ell(n)})$
- Choose random bits $b_1, \ldots, b_n$. Let $\mathbf{c_i} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, b_i)$.

CRHF against AC^0

KeyGen(1^n):

- $\mathbf{k} \leftarrow \text{KeyGen}^{Enc}(1^{\ell(n)})$
- Choose random bits $b_1, \ldots, b_n$. Let $\mathbf{c_i} = \text{Enc}(1^{\ell(n)}, \mathbf{k}, b_i)$.
- Output the following matrix $\mathbf{C}$:

$$\mathbf{C} = \left[\begin{array}{c|c|c|c} \mathbf{c_1} & \mathbf{c_2} & \cdots & \mathbf{c_n} \end{array}\right]$$

CRHF against AC^0

KeyGen(1^n):

- $\mathbf{k} \leftarrow \mathrm{KeyGen}^{Enc}(1^{\ell(n)})$
- Choose random bits $b_1, \ldots, b_n$. Let $\mathbf{c_i} = \mathrm{Enc}(1^{\ell(n)}, \mathbf{k}, b_i)$.
- Output the following matrix $\mathbf{C}$:

$$\mathbf{C} = \left[\begin{array}{c|c|c|c} \mathbf{c_1} & \mathbf{c_2} & \cdots & \mathbf{c_n} \end{array}\right]$$

Eval$(1^n, \mathbf{C}, \mathbf{x})$:

- Output $\mathbf{Cx}$.

CRHF against AC^0

KeyGen(1^n):

- $\mathbf{k} \leftarrow \text{KeyGen}^{Enc}(1^{\ell(n)})$
- Choose random bits $b_1, \ldots, b_n$. Let $\mathbf{c_i} = \text{Enc}(1^{\ell(n)}, \mathbf{k}, b_i)$.
- Output the following matrix $\mathbf{C}$:

$$\mathbf{C} = \begin{bmatrix} \mathbf{c_1} & \mathbf{c_2} & \cdots & \mathbf{c_n} \end{bmatrix} =$$

Eval($1^n, \mathbf{C}, \mathbf{x}$):

- Output $\mathbf{Cx}$.

CRHF against AC^0

Enc Challenger	**Enc Adversary**	**CRHF Adversary**
$\mathbf{k} \leftarrow \mathrm{KeyGen}(1^{\ell(n)})$	*sparse* $m_0, m_1 \in \{0,1\}^n$	

CRHF against AC^0

Enc Challenger		Enc Adversary	CRHF Adversary
$\mathbf{k} \leftarrow \mathrm{KeyGen}(1^{\ell(n)})$		*sparse* $m_0, m_1 \in \{0,1\}^n$	
	$\xleftarrow{\quad m_0,\ m_1 \quad}$		

CRHF against AC^0

Enc Challenger		**Enc Adversary**	**CRHF Adversary**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^{\ell(n)})$		*sparse* $m_0, m_1 \in \{0,1\}^n$	
	$\xleftarrow{m_0,\ m_1}$		
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{C} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, m_b)}$		

CRHF against AC^0

Enc Challenger		**Enc Adversary**		**CRHF Adversary**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^{\ell(n)})$		*sparse* $m_0, m_1 \in \{0,1\}^n$		
	$\xleftarrow{\quad m_0,\ m_1 \quad}$			
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{C} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, m_b)}$			
			$\xrightarrow{\quad \mathbf{C} \quad}$	

CRHF against AC^0

Enc Challenger		**Enc Adversary**		**CRHF Adversary**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^{\ell(n)})$		*sparse* $m_0, m_1 \in \{0,1\}^n$		
	$\xleftarrow{\ m_0,\ m_1\ }$			
$b \leftarrow \{0,1\}$	$\xrightarrow{\ \mathbf{C} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, m_b)\ }$			
			$\xrightarrow{\ \mathbf{C}\ }$	
			$\xleftarrow{\ \mathbf{x},\ \mathbf{y}\ }$	

CRHF against AC^0

$$
\begin{array}{lccc}
\textbf{Enc Challenger} & & \textbf{Enc Adversary} & & \textbf{CRHF Adversary} \\
\mathbf{k} \leftarrow \mathsf{KeyGen}(1^{\ell(n)}) & & \textit{sparse } m_0, m_1 \in \{0,1\}^n & & \\
 & \xleftarrow{\quad m_0,\ m_1 \quad} & & & \\
b \leftarrow \{0,1\} & \xrightarrow{\mathbf{C} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, m_b)} & & & \\
 & & & \xrightarrow{\quad \mathbf{C} \quad} & \\
 & & & \xleftarrow{\quad \mathbf{x},\ \mathbf{y} \quad} & \\
 & & \text{If } \mathbf{C}(\mathbf{x}-\mathbf{y}) \neq 0 : b' \leftarrow \{0,1\} & &
\end{array}
$$

CRHF against AC^0

Enc Challenger		**Enc Adversary**		**CRHF Adversary**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^{\ell(n)})$		*sparse* $m_0, m_1 \in \{0,1\}^n$		
	$\xleftarrow{\quad m_0,\ m_1 \quad}$			
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{C} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, m_b)}$			
			$\xrightarrow{\quad \mathbf{C} \quad}$	
			$\xleftarrow{\quad \mathbf{x},\ \mathbf{y} \quad}$	

If $\mathbf{C}(\mathbf{x}-\mathbf{y}) \neq 0 : b' \leftarrow \{0,1\}$

Else, if $\langle \mathbf{m_0}, \mathbf{x}-\mathbf{y} \rangle$

$= \langle \mathbf{m_1}, \mathbf{x}-\mathbf{y} \rangle = 0 : b' \leftarrow \{0,1\}$

CRHF against AC^0

Enc Challenger

$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^{\ell(n)})$

Enc Adversary

sparse $m_0, m_1 \in \{0,1\}^n$

CRHF Adversary

$\xleftarrow{\quad m_0,\ m_1 \quad}$

$b \leftarrow \{0,1\}$ $\quad \xrightarrow{\mathbf{C} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, m_b)}$

$\xrightarrow{\quad \mathbf{C} \quad}$

$\xleftarrow{\quad \mathbf{x},\ \mathbf{y} \quad}$

If $\mathbf{C}(\mathbf{x}-\mathbf{y}) \neq 0 : b' \leftarrow \{0,1\}$

Else, if $\langle \mathbf{m_0}, \mathbf{x}-\mathbf{y} \rangle$

$\quad = \langle \mathbf{m_1}, \mathbf{x}-\mathbf{y} \rangle = 0 : b' \leftarrow \{0,1\}$

Else, if $\langle \mathbf{m_0}, \mathbf{x}-\mathbf{y} \rangle =: b' = 0$

CRHF against AC^0

Enc Challenger		**Enc Adversary**		**CRHF Adversary**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^{\ell(n)})$		*sparse* $m_0, m_1 \in \{0,1\}^n$		
	$\xleftarrow{\quad m_0,\ m_1 \quad}$			
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{C} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, m_b)}$			
			$\xrightarrow{\quad \mathbf{C} \quad}$	
			$\xleftarrow{\quad \mathbf{x},\ \mathbf{y} \quad}$	

If $\mathbf{C}(\mathbf{x}-\mathbf{y}) \neq 0 : b' \leftarrow \{0,1\}$

Else, if $\langle \mathbf{m_0}, \mathbf{x}-\mathbf{y}\rangle$

$\quad = \langle \mathbf{m_1}, \mathbf{x}-\mathbf{y}\rangle = 0 : b' \leftarrow \{0,1\}$

Else, if $\langle \mathbf{m_0}, \mathbf{x}-\mathbf{y}\rangle =: b' = 0$

Else: $b' = 1$

CRHF against AC^0

Enc Challenger		**Enc Adversary**		**CRHF Adversary**
$\mathbf{k} \leftarrow \mathsf{KeyGen}(1^{\ell(n)})$		*sparse* $m_0, m_1 \in \{0,1\}^n$		
	$\xleftarrow{\quad m_0,\ m_1 \quad}$			
$b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{C} = \mathsf{Enc}(1^{\ell(n)}, \mathbf{k}, m_b)}$			
			$\xrightarrow{\quad \mathbf{C} \quad}$	
			$\xleftarrow{\quad \mathbf{x},\ \mathbf{y} \quad}$	
		If $\mathbf{C}(\mathbf{x}-\mathbf{y}) \neq 0 : b' \leftarrow \{0,1\}$ Else, if $\langle \mathbf{m_0}, \mathbf{x}-\mathbf{y} \rangle$ $= \langle \mathbf{m_1}, \mathbf{x}-\mathbf{y} \rangle = 0 : b' \leftarrow \{0,1\}$ Else, if $\langle \mathbf{m_0}, \mathbf{x}-\mathbf{y} \rangle =: b' = 0$ Else: $b' = 1$		
	$\xleftarrow{\quad b' \quad}$			

Candidate PKE against AC^0

Candidate PKE against AC^0

$\mathsf{KeyGen}(1^n)$:

Random $\mathbf{A} \in \{0,1\}^{n \times n}$, sparse $\mathbf{k} \in \{0,1\}^n$.

$\mathbf{pk} = (\mathbf{A}, \mathbf{Ak}), \mathbf{sk} = \mathbf{k}$.

Candidate PKE against AC^0

$\mathsf{KeyGen}(1^n)$:

Random $\mathbf{A} \in \{0,1\}^{n \times n}$, sparse $\mathbf{k} \in \{0,1\}^n$.

$\mathbf{pk} = (\mathbf{A}, \mathbf{Ak}), \mathbf{sk} = \mathbf{k}$.

$\mathsf{Enc}(1^n, \mathbf{pk} = (\mathbf{A}, \mathbf{Ak}), b)$:

Sparse $\mathbf{s} \in \{0,1\}^n$.

$b = 0$: Output $\mathbf{c} = (\mathbf{s}^T\mathbf{A}, \mathbf{s}^T\mathbf{Ak})$.

$b = 1$: Output $\mathbf{c} = (\mathbf{s}^T\mathbf{A}, b')$, where $b' \leftarrow \{0,1\}$.

Candidate PKE against AC^0

KeyGen(1^n) :

Random $\mathbf{A} \in \{0,1\}^{n \times n}$, sparse $\mathbf{k} \in \{0,1\}^n$.

$\mathbf{pk} = (\mathbf{A}, \mathbf{Ak}), \mathbf{sk} = \mathbf{k}$.

Enc$(1^n, \mathbf{pk} = (\mathbf{A}, \mathbf{Ak}), b)$:

Sparse $\mathbf{s} \in \{0,1\}^n$.

$b = 0$: Output $\mathbf{c} = (\mathbf{s}^T \mathbf{A}, \mathbf{s}^T \mathbf{Ak})$.

$b = 1$: Output $\mathbf{c} = (\mathbf{s}^T \mathbf{A}, b')$, where $b' \leftarrow \{0,1\}$.

Dec$(1^n, \mathbf{k}, \mathbf{c} = (\mathbf{c_1}^T, c_2))$:

If $\langle \mathbf{c_1}, \mathbf{k} \rangle = c_2$, output 0, else 1.

Candidate PKE against AC^0

KeyGen(1^n) :
  Random $\mathbf{A} \in \{0,1\}^{n \times n}$, sparse $\mathbf{k} \in \{0,1\}^n$.
  $\mathbf{pk} = (\mathbf{A}, \mathbf{Ak}), \mathbf{sk} = \mathbf{k}$.

Enc$(1^n, \mathbf{pk} = (\mathbf{A}, \mathbf{Ak}), b)$:
  Sparse $\mathbf{s} \in \{0,1\}^n$.
  $b = 0$: Output $\mathbf{c} = (\mathbf{s}^T\mathbf{A}, \mathbf{s}^T\mathbf{Ak})$.
  $b = 1$: Output $\mathbf{c} = (\mathbf{s}^T\mathbf{A}, b')$, where $b' \leftarrow \{0,1\}$.

Dec$(1^n, \mathbf{k}, \mathbf{c} = (\mathbf{c_1}^T, c_2))$:
  If $\langle \mathbf{c_1}, \mathbf{k} \rangle = c_2$, output 0, else 1.

- Secure if $(\mathbf{A}, \mathbf{Ak}) \approx_{AC^0} (\mathbf{A}, \mathbf{r})$ for random $\mathbf{A}$, sparse $\mathbf{k}$.

Results

Unconditional constructions against AC^0:

- OWF, PRG. (other constructions known from [Has87], [Vio12], [MST06])
- Weak PRF.
- Symmetric Encryption.
- Collision Resistant Hash Functions.

Constructions against NC^1 based on $L \not\subseteq NC^1$:

- OWF, PRG. (similar, independent, constructions in [AR15])
- Public-Key Encryption.
- Collision Resistant Hash Functions.

Public-Key Encryption against NC^1

- Based on the worst-case assumption that $L \not\subseteq NC^1$.
  - L - class of languages with polynomial-sized branching programs.
  - NC^1 - class of languages with polynomial-sized *constant-width* branching programs.

Public-Key Encryption against NC^1

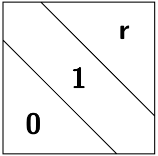
- Based on the worst-case assumption that $\mathsf{L} \not\subseteq \mathsf{NC}^1$.
  - L - class of languages with polynomial-sized branching programs.
  - NC^1 - class of languages with polynomial-sized *constant-width* branching programs.
- Makes use of algebraic structure in the Randomised Encodings for L by Ishai-Kushilevitz [IK00].

Public-Key Encryption against NC^1

$$\mathbf{M_0^n} = \left[\begin{array}{c|c} \mathbf{0}^T & 0 \\ \hline \mathbf{I}_{n-1} & \mathbf{0} \end{array}\right] \qquad \mathbf{M_1^n} = \left[\begin{array}{c|c} \mathbf{0}^T & 1 \\ \hline \mathbf{I}_{n-1} & \mathbf{0} \end{array}\right]$$

Public-Key Encryption against NC^1

$$\mathbf{M_0^n} = \left[\begin{array}{c|c} \mathbf{0}^T & 0 \\ \hline \mathbf{I}_{n-1} & \mathbf{0} \end{array}\right] \qquad \mathbf{M_1^n} = \left[\begin{array}{c|c} \mathbf{0}^T & 1 \\ \hline \mathbf{I}_{n-1} & \mathbf{0} \end{array}\right]$$

$R_L^n =$ 

$$R_R^n = \left[\begin{array}{c|c} \mathbf{I}_{n-1} & \mathbf{r} \\ \hline \mathbf{0}^T & 1 \end{array}\right]$$

Public-Key Encryption against NC^1

$$\mathbf{M_0^n} = \left[\begin{array}{c|c} \mathbf{0}^T & 0 \\ \hline \mathbf{I}_{n-1} & \mathbf{0} \end{array}\right] \qquad \mathbf{M_1^n} = \left[\begin{array}{c|c} \mathbf{0}^T & 1 \\ \hline \mathbf{I}_{n-1} & \mathbf{0} \end{array}\right]$$

$$R_L^n = \left[\begin{array}{ccc} \mathbf{1} & & \mathbf{r} \\ & \ddots & \\ \mathbf{0} & & \mathbf{1} \end{array}\right] \qquad R_R^n = \left[\begin{array}{c|c} \mathbf{I}_{n-1} & \mathbf{r} \\ \hline \mathbf{0}^T & 1 \end{array}\right]$$

Theorem (IK00)

If $\mathsf{L} \not\subseteq \mathsf{NC}^1$*, then, for infinitely many values of* n*:*

$$R_L^n \mathbf{M_0^n} R_R^n \approx_{\mathsf{NC}^1} R_L^n \mathbf{M_1^n} R_R^n$$

Public-Key Encryption against NC^1

$\mathsf{KeyGen}(1^n)$:

$$\mathbf{pk} = \mathbf{R}_{\mathbf{L}}^{\mathbf{n}}\mathbf{M}_{\mathbf{0}}^{\mathbf{n}}\mathbf{R}_{\mathbf{R}}^{\mathbf{n}} = \begin{pmatrix} & & \mathbf{r}' \\ & \mathbf{1} & \\ \mathbf{0} & & \end{pmatrix} \begin{pmatrix} \mathbf{0}^T & 0 \\ \mathbf{I}_{n-1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{I}_{n-1} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{pmatrix} \qquad \mathbf{sk} = \begin{pmatrix} \mathbf{r} \\ 1 \end{pmatrix}$$

Public-Key Encryption against NC^1

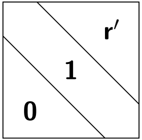
KeyGen(1^n):

$$\mathbf{pk} = \mathbf{R}_\mathbf{L}^\mathbf{n}\mathbf{M}_\mathbf{0}^\mathbf{n}\mathbf{R}_\mathbf{R}^\mathbf{n} = \begin{bmatrix} \mathbf{1} & \mathbf{r}' \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{0}^T & 0 \\ \mathbf{I}_{n-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{n-1} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix} \qquad \mathbf{sk} = \begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix}$$

(Notice that $\mathbf{pk} \cdot \mathbf{sk} = \mathbf{0}$.)

Public-Key Encryption against NC^1

KeyGen(1^n):

$\mathbf{pk} = \mathbf{R_L^n M_0^n R_R^n} =$  $\begin{pmatrix} \mathbf{0}^T & 0 \\ \mathbf{I}_{n-1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{I}_{n-1} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{pmatrix} \qquad \mathbf{sk} = \begin{pmatrix} \mathbf{r} \\ 1 \end{pmatrix}$

(Notice that $\mathbf{pk} \cdot \mathbf{sk} = \mathbf{0}$.)

Enc$(1^n, \mathbf{pk}, b)$:

- Pick $\mathbf{s} \leftarrow \{0,1\}^n$. Let $\mathbf{t} = (0\ 0 \cdots\ 0\ 1)^T$.
- Output $\mathbf{c}^T = \mathbf{s}^T\mathbf{pk} + b\mathbf{t}^T$.

Public-Key Encryption against NC^1

KeyGen(1^n):

$$\mathbf{pk} = \mathbf{R_L^n M_0^n R_R^n} = \begin{bmatrix} \mathbf{1} & \mathbf{r}' \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{0}^T & 0 \\ \mathbf{I}_{n-1} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{n-1} & \mathbf{r} \\ \mathbf{0}^T & 1 \end{bmatrix} \qquad \mathbf{sk} = \begin{bmatrix} \mathbf{r} \\ 1 \end{bmatrix}$$

(Notice that $\mathbf{pk} \cdot \mathbf{sk} = \mathbf{0}$.)

Enc($1^n, \mathbf{pk}, b$):

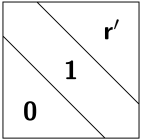
- Pick $\mathbf{s} \leftarrow \{0,1\}^n$. Let $\mathbf{t} = (0\ 0 \cdots\ 0\ 1)^T$.
- Output $\mathbf{c}^T = \mathbf{s}^T\mathbf{pk} + b\mathbf{t}^T$.

Dec($1^n, \mathbf{sk}, \mathbf{c}$): Output $\langle \mathbf{c}, \mathbf{sk} \rangle$.

Public-Key Encryption against NC^1

KeyGen(1^n):

$$\mathbf{pk} = \mathbf{R_L^n M_0^n R_R^n} =$$



$$\left[\begin{array}{c|c} \mathbf{0}^T & 0 \\ \hline \mathbf{I}_{n-1} & \mathbf{0} \end{array}\right] \left[\begin{array}{c|c} \mathbf{I}_{n-1} & \mathbf{r} \\ \hline \mathbf{0}^T & 1 \end{array}\right] \qquad \mathbf{sk} = \left[\begin{array}{c} \mathbf{r} \\ \hline 1 \end{array}\right]$$

(Notice that $\mathbf{pk} \cdot \mathbf{sk} = \mathbf{0}$.)

Enc($1^n, \mathbf{pk}, b$):

- Pick $\mathbf{s} \leftarrow \{0,1\}^n$. Let $\mathbf{t} = (0\ 0 \cdots\ 0\ 1)^T$.
- Output $\mathbf{c}^T = \mathbf{s}^T\mathbf{pk} + b\mathbf{t}^T$.

Dec($1^n, \mathbf{sk}, \mathbf{c}$): Output $\langle \mathbf{c}, \mathbf{sk} \rangle$. $(= (\mathbf{s}^T\mathbf{pk} + b\mathbf{t}^T)\mathbf{sk} = 0 + b\langle \mathbf{t}, \mathbf{sk} \rangle = b)$

Public-Key Encryption against NC^1

Game 1 - Semantic Security

Ch		**Adv**
$(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathsf{KeyGen}(1^n)$		
$b \leftarrow \{0, 1\}$	$\xrightarrow{\mathbf{pk},\ \mathsf{Enc}(1^n, \mathbf{pk}, b)}$	
	$\xleftarrow{b'}$	

Public-Key Encryption against NC^1

Game 1 - Semantic Security

Ch **Adv**

$\mathbf{pk} \leftarrow R_L^n \mathbf{M_n^0} R_R^n$

$s \leftarrow \{0,1\}^n, b \leftarrow \{0,1\}$ $\xrightarrow{\mathbf{pk},\ \mathbf{s}^T\mathbf{pk} + b\mathbf{t}^T}$

$\xleftarrow{b'}$

Public-Key Encryption against NC^1

Game 1 - Semantic Security

Ch		**Adv**
$\mathbf{pk} \leftarrow R_L^n \mathbf{M_n^0} R_R^n$		
$s \leftarrow \{0,1\}^n, b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{pk},\ \mathbf{s}^T\mathbf{pk} + b\mathbf{t}^T}$	
	$\xleftarrow{b'}$	

Game 2

Ch		**Adv**
$\mathbf{pk} \leftarrow R_L^n \mathbf{M_n^1} R_R^n$		
$s \leftarrow \{0,1\}^n, b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{pk},\ \mathbf{s}^T\mathbf{pk} + b\mathbf{t}^T}$	
	$\xleftarrow{b'}$	

Public-Key Encryption against NC^1

Game 1 - Semantic Security

Ch		**Adv**
$\mathbf{pk} \leftarrow R_L^n \mathbf{M_n^0} R_R^n$		
$s \leftarrow \{0,1\}^n, b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{pk},\ \mathbf{s}^T\mathbf{pk} + b\mathbf{t}^T}$	
	$\xleftarrow{b'}$	

Game 2

Ch		**Adv**
$\mathbf{pk} \leftarrow R_L^n \mathbf{M_n^1} R_R^n$		
$s \leftarrow \{0,1\}^n, b \leftarrow \{0,1\}$	$\xrightarrow{\mathbf{pk},\ \mathbf{r}^T}$	
	$\xleftarrow{b'}$	

Open Problems

- Public-Key Encryption against AC^0.
- Better PRGs and PRFs against NC^1.
- Improve upon Merkle puzzles without too many assumptions.
  - Perhaps using recent Fine-Grained Complexity results.
- Constructions against $\mathrm{AC}^0[p]$.

Miklós Ajtai and Michael Ben-Or.
A theorem on probabilistic constant depth computations.
In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 471–474, 1984.

Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz.
On basing one-way functions on np-hardness.
In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 701–710. ACM, 2006.

Benny Applebaum and Pavel Raykov.
On the relationship between statistical zero-knowledge and statistical randomized encodings.
Electronic Colloquium on Computational Complexity (ECCC), 22:186, 2015.

Miklós Ajtai and Avi Wigderson.
Deterministic simulation of probabilistic constant depth circuits (preliminary version).

In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 11–19, 1985.

Andrej Bogdanov and Christina Brzuska.
On basing size-verifiable one-way functions on np-hardness.
In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography - 12th Theory of Cryptography Conference, TCC 2015, Warsaw, Poland, March 23-25, 2015, Proceedings, Part I*, volume 9014 of *Lecture Notes in Computer Science*, pages 1–6. Springer, 2015.

Mark Braverman.
Polylogarithmic independence fools AC^0 circuits.
J. ACM, 57(5), 2010.

Christian Cachin and Ueli Maurer.
Unconditional security against memory-bounded adversaries.
In *Advances in CryptologyCRYPTO'97*, pages 292–306. Springer, 1997.

Robert G. Gallager.
Low-density parity-check codes.

IRE Trans. Information Theory, 8(1):21–28, 1962.

Johan Hastad.
One-way permutations in nc 0.
Information Processing Letters, 26(3):153–155, 1987.

Ueli M Maurer.
Conditionally-perfect secrecy and a provably-secure randomized cipher.
Journal of Cryptology, 5(1):53–66, 1992.

Ralph C. Merkle.
Secure communications over insecure channels.
Commun. ACM, 21(4):294–299, 1978.

Elchanan Mossel, Amir Shpilka, and Luca Trevisan.
On epsilon-biased generators in nc^0.
Random Struct. Algorithms, 29(1):56–81, 2006.

Rafail Ostrovsky.
One-way functions, hard on average problems, and statistical zero-knowledge proofs.

In *Proceedings of the Sixth Annual Structure in Complexity Theory Conference, Chicago, Illinois, USA, June 30 - July 3, 1991*, pages 133–138, 1991.

Avishay Tal.
Tight bounds on the fourier spectrum of ac^0.
Electronic Colloquium on Computational Complexity (ECCC), 21:174, 2014.

Emanuele Viola.
The complexity of distributions.
SIAM Journal on Computing, 41(1):191–218, 2012.