# Low Noise LPN: KDM Secure Public Key Encryption and Sample Amplification

Nico Döttling[1]

[1]Aarhus University

March 31, 2015

# Learning Parity with Noise

- Solve noisy linear decoding problem for random linear codes over $\mathbb{F}_2$
- Given $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, find secret $\mathbf{s}$

## Learning Parity with Noise

- Solve noisy linear decoding problem for random linear codes over $\mathbb{F}_2$
- Given $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, find secret $\mathbf{s}$
- **Bounded Samples Version:** $\mathrm{LPN}(n, m, \rho)$, $\mathbf{A}$ has $n$ columns and $m$ rows, $\mathbf{e}$ chosen component-wise by $\mathrm{Ber}(\rho)$

# Learning Parity with Noise

- Solve noisy linear decoding problem for random linear codes over $\mathbb{F}_2$
- Given $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, find secret $\mathbf{s}$
- **Bounded Samples Version:** $\mathsf{LPN}(n, m, \rho)$, $\mathbf{A}$ has $n$ columns and $m$ rows, $\mathbf{e}$ chosen component-wise by $\mathsf{Ber}(\rho)$
- **Unbounded Samples Version:** $\mathsf{LPN}(n, \rho)$, Given an unbounded number of samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ for fresh random $\mathbf{a}$ and $e \leftarrow_{\$} \mathsf{Ber}(\rho)$, find $\mathbf{s}$

# Learning Parity with Noise

- Solve noisy linear decoding problem for random linear codes over $\mathbb{F}_2$
- Given $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, find secret $\mathbf{s}$
- **Bounded Samples Version:** $\mathsf{LPN}(n, m, \rho)$, $\mathbf{A}$ has $n$ columns and $m$ rows, $\mathbf{e}$ chosen component-wise by $\mathsf{Ber}(\rho)$
- **Unbounded Samples Version:** $\mathsf{LPN}(n, \rho)$, Given an unbounded number of samples of the form $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ for fresh random $\mathbf{a}$ and $e \leftarrow_{\$} \mathsf{Ber}(\rho)$, find $\mathbf{s}$

# Decisional Learning Parity with Noise

- **Decisional Problem** $\mathsf{DLPN}(n, m, \rho)$: distinguish $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ from $(\mathbf{A}, \mathbf{u})$ for uniformly random $\mathbf{u}$
- **Unbounded samples version** $\mathsf{DLPN}(n, \rho)$: Distinguish samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ from $(\mathbf{a}, u)$ (again, $u$ is uniformly random)

# Decisional Learning Parity with Noise

- **Decisional Problem** DLPN$(n, m, \rho)$: distinguish $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ from $(\mathbf{A}, \mathbf{u})$ for uniformly random $\mathbf{u}$
- **Unbounded samples version** DLPN$(n, \rho)$: Distinguish samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ from $(\mathbf{a}, u)$ (again, $u$ is uniformly random)
- Hardness of DLPN follows from LPN e.g. [KS(S)06, AIK07]

# Decisional Learning Parity with Noise

- **Decisional Problem** $\mathsf{DLPN}(n, m, \rho)$: distinguish $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ from $(\mathbf{A}, \mathbf{u})$ for uniformly random $\mathbf{u}$
- **Unbounded samples version** $\mathsf{DLPN}(n, \rho)$: Distinguish samples $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ from $(\mathbf{a}, u)$ (again, $u$ is uniformly random)
- Hardness of DLPN follows from LPN e.g. [KS(S)06, AIK07]

# LPN based Crypto

- High noise LPN ($\rho < 1/2$ constant): Private key crypto [BFKL93,HB01,...]
- Low noise LPN ($\rho = O(1/\sqrt{n})$): Public key crypto [Ale03]

# LPN based Crypto

- High noise LPN ($\rho < 1/2$ constant): Private key crypto [BFKL93,HB01,...]
- Low noise LPN ($\rho = O(1/\sqrt{n})$): Public key crypto [Ale03]
- Trick: Inner product of two random low weight vectors x and y is biased to 0

# LPN based Crypto

- High noise LPN ($\rho < 1/2$ constant): Private key crypto [BFKL93,HB01,...]
- Low noise LPN ($\rho = O(1/\sqrt{n})$): Public key crypto [Ale03]
- Trick: Inner product of two random low weight vectors $\mathbf{x}$ and $\mathbf{y}$ is biased to 0
- Trapdoors similar to lattice based crypto

# LPN based Crypto

- High noise LPN ($\rho < 1/2$ constant): Private key crypto [BFKL93,HB01,...]
- Low noise LPN ($\rho = O(1/\sqrt{n})$): Public key crypto [Ale03]
- Trick: Inner product of two random low weight vectors $\mathbf{x}$ and $\mathbf{y}$ is biased to 0
- Trapdoors similar to lattice based crypto
- Recently: IND-CCA secure public key encryption [DMN12,KMP14], composable oblivious transfer [DDN14]

# LPN based Crypto

- High noise LPN ($\rho < 1/2$ constant): Private key crypto [BFKL93,HB01,...]
- Low noise LPN ($\rho = O(1/\sqrt{n})$): Public key crypto [Ale03]
- Trick: Inner product of two random low weight vectors $\mathbf{x}$ and $\mathbf{y}$ is biased to 0
- Trapdoors similar to lattice based crypto
- Recently: IND-CCA secure public key encryption [DMN12,KMP14], composable oblivious transfer [DDN14]

# Results in this work

- LPN Sample amplification: Base hardness of unbounded samples LPN on bounded samples LPN (modest noise increase)
- KDM secure public key encryption from low noise LPN

# Results in this work

- LPN Sample amplification: Base hardness of unbounded samples LPN on bounded samples LPN (modest noise increase)
- KDM secure public key encryption from low noise LPN
- **Common Theme:** Computational Rerandomization of LPN instances

# Results in this work

- LPN Sample amplification: Base hardness of unbounded samples LPN on bounded samples LPN (modest noise increase)
- KDM secure public key encryption from low noise LPN
- **Common Theme:** Computational Rerandomization of LPN instances

# LPN Rerandomization

- Goal: Given a *few* LPN samples, generate more samples
- In [Lyu05]: Given instance $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$, set

$$\mathbf{a}' = \mathbf{r}^\top \mathbf{A}$$
$$y' = \mathbf{r}^\top \mathbf{y} = \mathbf{r}^\top \mathbf{A}\mathbf{s} + \mathbf{r}^\top \mathbf{e} = \langle \mathbf{a}', \mathbf{s} \rangle + \langle \mathbf{r}, \mathbf{e} \rangle$$

## LPN Rerandomization

- Goal: Given a *few* LPN samples, generate more samples
- In [Lyu05]: Given instance $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$, set

$$\mathbf{a}' = \mathbf{r}^\top \mathbf{A}$$
$$y' = \mathbf{r}^\top \mathbf{y} = \mathbf{r}^\top \mathbf{A}\mathbf{s} + \mathbf{r}^\top \mathbf{e} = \langle \mathbf{a}', \mathbf{s} \rangle + \langle \mathbf{r}, \mathbf{e} \rangle$$

- Given that $\mathbf{r}$ has $\approx n$ bits of entropy, LHL yields that $\mathbf{a}'$ is stat. close to uniform $\Rightarrow$ $(\mathbf{a}', y')$ correct LPN sample.

# LPN Rerandomization

- Goal: Given a *few* LPN samples, generate more samples
- In [Lyu05]: Given instance $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$, set

$$\mathbf{a}' = \mathbf{r}^\top \mathbf{A}$$
$$y' = \mathbf{r}^\top \mathbf{y} = \mathbf{r}^\top \mathbf{A}\mathbf{s} + \mathbf{r}^\top \mathbf{e} = \langle \mathbf{a}', \mathbf{s} \rangle + \langle \mathbf{r}, \mathbf{e} \rangle$$

- Given that $\mathbf{r}$ has $\approx n$ bits of entropy, LHL yields that $\mathbf{a}'$ is stat. close to uniform $\Rightarrow$ $(\mathbf{a}', y')$ correct LPN sample.
- Necessary condition: $\mathbf{r}$ must be heavy!
- Problem: Noise term $\langle \mathbf{r}, \mathbf{e} \rangle$ is stat. close to unbiased coin.

# LPN Rerandomization

- Goal: Given a *few* LPN samples, generate more samples
- In [Lyu05]: Given instance $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$, set

$$\mathbf{a}' = \mathbf{r}^\top \mathbf{A}$$
$$y' = \mathbf{r}^\top \mathbf{y} = \mathbf{r}^\top \mathbf{A}\mathbf{s} + \mathbf{r}^\top \mathbf{e} = \langle \mathbf{a}', \mathbf{s} \rangle + \langle \mathbf{r}, \mathbf{e} \rangle$$

- Given that $\mathbf{r}$ has $\approx n$ bits of entropy, LHL yields that $\mathbf{a}'$ is stat. close to uniform $\Rightarrow$ $(\mathbf{a}', y')$ correct LPN sample.
- Necessary condition: $\mathbf{r}$ must be heavy!
- Problem: Noise term $\langle \mathbf{r}, \mathbf{e} \rangle$ is stat. close to unbiased coin.

# Extended (low noise) LPN

- Need a computational substitute for the LHL
- Extended LPN problem [AP12,KMP14]: LPN with leakage

# Extended (low noise) LPN

- Need a computational substitute for the LHL
- Extended LPN problem [AP12,KMP14]: LPN with leakage
- If advice is of the form $(\mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$ for a (random) low weight $\mathbf{z}$, then DLPN remains hard

# Extended (low noise) LPN

- Need a computational substitute for the LHL
- Extended LPN problem [AP12,KMP14]: LPN with leakage
- If advice is of the form $(\mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$ for a (random) low weight $\mathbf{z}$, then DLPN remains hard
- Specifically:

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) \approx_c (\mathbf{A}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$$

# Extended (low noise) LPN

- Need a computational substitute for the LHL
- Extended LPN problem [AP12,KMP14]: LPN with leakage
- If advice is of the form $(\mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$ for a (random) low weight $\mathbf{z}$, then DLPN remains hard
- Specifically:

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) \approx_c (\mathbf{A}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$$

- Full paper (eprint): Arbitrary short leakage function $\gamma$.

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}, \gamma(\mathbf{e})) \approx_c (\mathbf{A}, \mathbf{u}, \gamma(\mathbf{e}))$$

# Extended (low noise) LPN

- Need a computational substitute for the LHL
- Extended LPN problem [AP12,KMP14]: LPN with leakage
- If advice is of the form $(\mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$ for a (random) low weight $\mathbf{z}$, then DLPN remains hard
- Specifically:

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) \approx_c (\mathbf{A}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$$

- Full paper (eprint): Arbitrary short leakage function $\gamma$.

$$(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e}, \gamma(\mathbf{e})) \approx_c (\mathbf{A}, \mathbf{u}, \gamma(\mathbf{e}))$$

- Can be based on standard LPN

# Extended (low noise) LPN

- Need a computational substitute for the LHL
- Extended LPN problem [AP12,KMP14]: LPN with leakage
- If advice is of the form $(\mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$ for a (random) low weight $\mathbf{z}$, then DLPN remains hard
- Specifically:

$$(\mathbf{A}, \mathbf{As} + \mathbf{e}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle) \approx_c (\mathbf{A}, \mathbf{u}, \mathbf{z}, \langle \mathbf{z}, \mathbf{e} \rangle)$$

- Full paper (eprint): Arbitrary short leakage function $\gamma$.

$$(\mathbf{A}, \mathbf{As} + \mathbf{e}, \gamma(\mathbf{e})) \approx_c (\mathbf{A}, \mathbf{u}, \gamma(\mathbf{e}))$$

- Can be based on standard LPN

# Extended (low noise) LPN

- Dual Formulation:

$$(\mathbf{A}, \mathbf{r}^\top \mathbf{A}, \mathbf{z}, \langle \mathbf{r}, \mathbf{z} \rangle) \approx_c (\mathbf{A}, \mathbf{U}, \mathbf{z}, \langle \mathbf{r}, \mathbf{z} \rangle)$$

- Matrix Form

$$(\mathbf{A}, \mathbf{R}\mathbf{A}, \mathbf{z}, \mathbf{R}\mathbf{z}) \approx_c (\mathbf{A}, \mathbf{U}, \mathbf{z}, \mathbf{R}\mathbf{z})$$

# Extended (low noise) LPN

- Dual Formulation:

$$(\mathbf{A}, \mathbf{r}^\top \mathbf{A}, \mathbf{z}, \langle \mathbf{r}, \mathbf{z} \rangle) \approx_c (\mathbf{A}, \mathbf{U}, \mathbf{z}, \langle \mathbf{r}, \mathbf{z} \rangle)$$

- Matrix Form

$$(\mathbf{A}, \mathbf{R}\mathbf{A}, \mathbf{z}, \mathbf{R}\mathbf{z}) \approx_c (\mathbf{A}, \mathbf{U}, \mathbf{z}, \mathbf{R}\mathbf{z})$$

# LPN with bounded samples vs. LPN with unbounded samples

**Theorem**

DLPN$(n, \rho')$ *is as hard as* DLPN$(n, 2n, \rho)$ *whenever* $\rho' \geq \rho^2 2n$

# Proof Idea

| | static | volatile | samples |
|---|---|---|---|
| 1. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ | $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$<br>$e \leftarrow_\$ \mathrm{Ber}(\rho')$ | $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ |
| | | | |
| | | | |
| | | | |

# Proof Idea

eDLPN: $(\mathbf{A}, \mathbf{r}^\top\mathbf{A}, \mathbf{z}, \mathbf{r}^\top\mathbf{z}) \approx_c (\mathbf{A}, \mathbf{a}, \mathbf{z}, \mathbf{r}^\top\mathbf{z}) \equiv (\mathbf{A}, \mathbf{a}, \mathbf{z}, e)$

| | static | volatile | samples |
|---|---|---|---|
| 1. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ | $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$<br>$e \leftarrow_\$ \mathrm{Ber}(\rho')$ | $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s}\rangle + e)$ |
| 2. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$<br>$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n\times n}$<br>$\mathbf{z} \leftarrow_\$ S(2n, \rho)$ | $\mathbf{r} \leftarrow_\$ \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top\mathbf{A}, \mathbf{r}^\top\mathbf{A}\mathbf{s} + \mathbf{r}^\top\mathbf{z})$ |
| | | | |
| | | | |

# Proof Idea

| | static | volatile | samples |
|---|---|---|---|
| 1. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ | $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$<br>$e \leftarrow_\$ \mathrm{Ber}(\rho')$ | $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ |
| 2. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$<br>$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$<br>$\mathbf{z} \leftarrow_\$ S(2n, \rho)$ | $\mathbf{r} \leftarrow_\$ \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top \mathbf{A}\mathbf{s} + \mathbf{r}^\top \mathbf{z})$<br>$= (\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top (\mathbf{A}\mathbf{s} + \mathbf{z}))$ |
| | | | |
| | | | |

# Proof Idea

| | static | volatile | samples |
|---|---|---|---|
| 1. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ | $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$ <br> $e \leftarrow_\$ \mathrm{Ber}(\rho')$ | $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ |
| 2. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ <br> $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$ <br> $\mathbf{z} \leftarrow_\$ S(2n, \rho)$ | $\mathbf{r} \leftarrow_\$ \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top \mathbf{A}\mathbf{s} + \mathbf{r}^\top \mathbf{z})$ <br> $= (\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top (\mathbf{A}\mathbf{s} + \mathbf{z}))$ |
| 3. | $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$ <br> $\mathbf{y} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{z}$ | $\mathbf{r} \leftarrow_\$ \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top \mathbf{y})$ |

# Proof Idea

$$\text{DLPN: } (\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{z}) \approx_c (\mathbf{A}, \mathbf{u})$$

| | static | volatile | samples |
|---|---|---|---|
| 1. | $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$ | $\mathbf{a} \leftarrow_{\$} \mathbb{F}_2^n$<br>$e \leftarrow_{\$} \mathrm{Ber}(\rho')$ | $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ |
| 2. | $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$<br>$\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{2n \times n}$<br>$\mathbf{z} \leftarrow_{\$} S(2n, \rho)$ | $\mathbf{r} \leftarrow_{\$} \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top \mathbf{A}\mathbf{s} + \mathbf{r}^\top \mathbf{z})$<br>$= (\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top (\mathbf{A}\mathbf{s} + \mathbf{z}))$ |
| 3. | $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{2n \times n}$<br>$\mathbf{y} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{z}$ | $\mathbf{r} \leftarrow_{\$} \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top \mathbf{y})$ |
| 4. | $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{2n \times n}$<br>$\mathbf{u} \leftarrow_{\$} \mathbb{F}_2^{2n}$ | $\mathbf{r} \leftarrow_{\$} \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top \mathbf{A}, \mathbf{r}^\top \mathbf{u})$ |

# Proof Idea

eDLPN: $(\mathbf{A}, \mathbf{r}^\top\mathbf{A}, \mathbf{u}, \mathbf{r}^\top\mathbf{u}) \approx_c (\mathbf{A}, \mathbf{a}, \mathbf{u}, \mathbf{r}^\top\mathbf{u}) \approx_s (\mathbf{A}, \mathbf{a}, \mathbf{u}, u)$

| | static | volatile | samples |
|---|---|---|---|
| 1. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$ | $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$<br>$e \leftarrow_\$ \mathrm{Ber}(\rho')$ | $(\mathbf{a}, \langle \mathbf{a}, \mathbf{s} \rangle + e)$ |
| 2. | $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$<br>$\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$<br>$\mathbf{z} \leftarrow_\$ S(2n, \rho)$ | $\mathbf{r} \leftarrow_\$ \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top\mathbf{A}, \mathbf{r}^\top\mathbf{A}\mathbf{s} + \mathbf{r}^\top\mathbf{z})$<br>$= (\mathbf{r}^\top\mathbf{A}, \mathbf{r}^\top(\mathbf{A}\mathbf{s} + \mathbf{z}))$ |
| 3. | $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$<br>$\mathbf{y} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{z}$ | $\mathbf{r} \leftarrow_\$ \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top\mathbf{A}, \mathbf{r}^\top\mathbf{y})$ |
| 4. | $\mathbf{A} \leftarrow_\$ \mathbb{F}_2^{2n \times n}$<br>$\mathbf{u} \leftarrow_\$ \mathbb{F}_2^{2n}$ | $\mathbf{r} \leftarrow_\$ \mathrm{Ber}(2n, \rho)$ | $(\mathbf{r}^\top\mathbf{A}, \mathbf{r}^\top\mathbf{u})$ |
| 5. | | $\mathbf{a} \leftarrow_\$ \mathbb{F}_2^n$<br>$u \leftarrow_\$ \mathbb{F}_2$ | $(\mathbf{a}, u)$ |

# Key Dependent Message Secure Encryption

- Schemes that stay secure even if adversary is given encryptions of secret keys
- Simplest Case: Circular Security.

# Key Dependent Message Secure Encryption

- Schemes that stay secure even if adversary is given encryptions of secret keys
- Simplest Case: Circular Security.
- Why care?

# Key Dependent Message Secure Encryption

- Schemes that stay secure even if adversary is given encryptions of secret keys
- Simplest Case: Circular Security.
- Why care?
- Harddisk encryption, symbolic soundness, FHE Bootstrapping...

# Key Dependent Message Secure Encryption

- Schemes that stay secure even if adversary is given encryptions of secret keys
- Simplest Case: Circular Security.
- Why care?
- Harddisk encryption, symbolic soundness, FHE Bootstrapping...
- Does not follow naturally from IND-CPA encryption

# Key Dependent Message Secure Encryption

- Schemes that stay secure even if adversary is given encryptions of secret keys
- Simplest Case: Circular Security.
- Why care?
- Harddisk encryption, symbolic soundness, FHE Bootstrapping...
- Does not follow naturally from IND-CPA encryption

# Key Dependent Message Secure Encryption

- Schemes that stay secure even if adversary is given encryptions of secret keys
- Simplest Case: Circular Security.
- Why care?
- Harddisk encryption, symbolic soundness, FHE Bootstrapping...
- Does not follow naturally from IND-CPA encryption

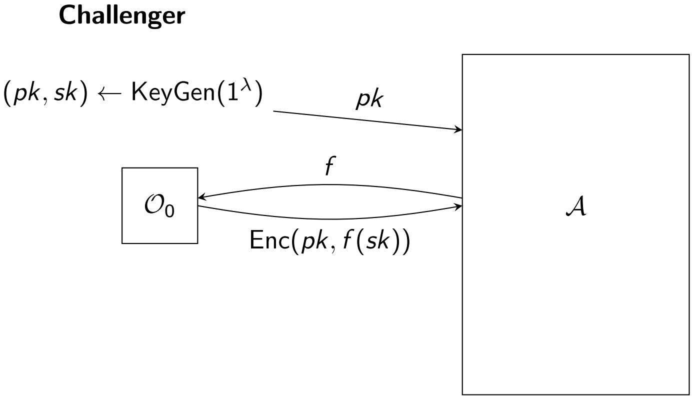
# KDM-CPA Security

**Challenger**

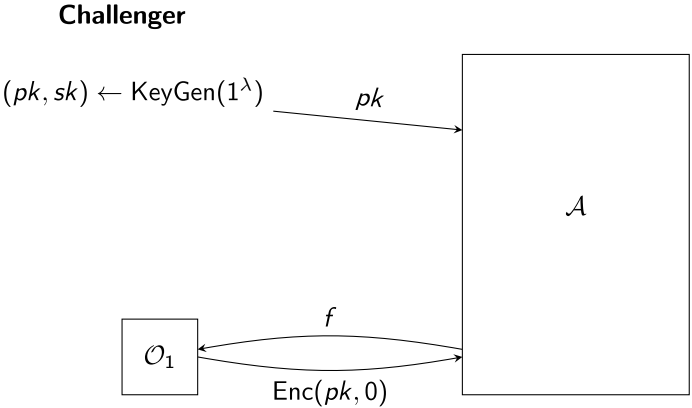
$(pk, sk) \leftarrow \mathsf{KeyGen}(1^\lambda)$ $\xrightarrow{pk}$

$\mathcal{A}$

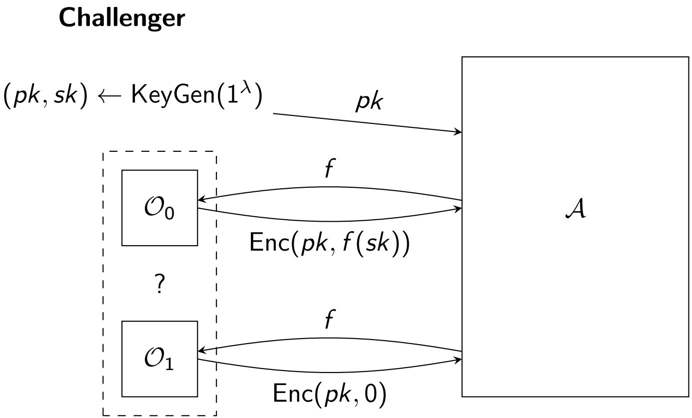
# KDM-CPA Security



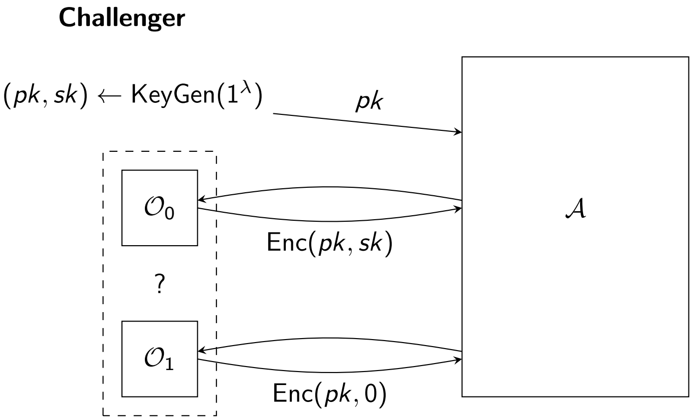
# KDM-CPA Security



# KDM-CPA Security

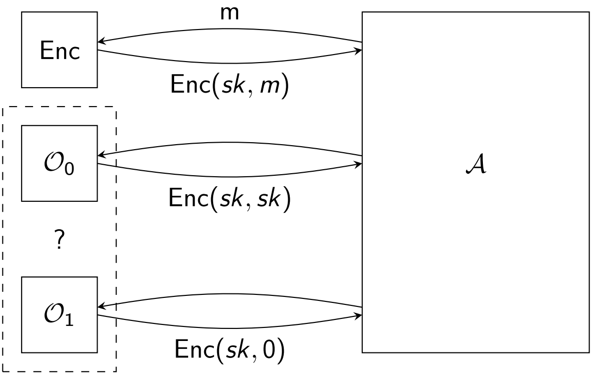


# CIRC-CPA Security



# CIRC-CPA Security (Private Key)

**Challenger**



# Private Key Scheme of [ACPS12]

Let $\mathbf{G}$ be the generator of an asymptotically good $[k, n]$ code that can efficiently decode from a constant fraction of errors.

- KeyGen: $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$

# Private Key Scheme of [ACPS12]

Let $\mathbf{G}$ be the generator of an asymptotically good $[k, n]$ code that can efficiently decode from a constant fraction of errors.

- KeyGen: $\mathbf{s} \leftarrow_\$ \mathbb{F}_2^n$
- Enc$(\mathbf{s}, m)$: $\mathbf{C}_1 \leftarrow_\$ \mathbb{F}_2^{k \times n}$, $\mathbf{e} \leftarrow_\$ \text{Ber}(k, \rho)$, $\mathbf{c}_1 = \mathbf{C}_1\mathbf{s} + \mathbf{e} + \mathbf{Gm}$
  $c = (\mathbf{C}_1, \mathbf{c}_2)$

# Private Key Scheme of [ACPS12]

Let $\mathbf{G}$ be the generator of an asymptotically good $[k, n]$ code that can efficiently decode from a constant fraction of errors.

- KeyGen: $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$
- $\mathsf{Enc}(\mathbf{s}, m)$: $\mathbf{C}_1 \leftarrow_{\$} \mathbb{F}_2^{k \times n}$, $\mathbf{e} \leftarrow_{\$} \mathsf{Ber}(k, \rho)$, $\mathbf{c}_1 = \mathbf{C}_1\mathbf{s} + \mathbf{e} + \mathbf{Gm}$
  $c = (\mathbf{C}_1, \mathbf{c}_2)$
- $\mathsf{Dec}(\mathbf{s}, c)$: $(\mathbf{C}_1, \mathbf{c}_2) = c$, $\mathbf{z} = \mathbf{c}_2 - \mathbf{C}_1 \cdot \mathbf{s}$
  $m = \mathsf{Decode}(\mathbf{z})$

## Correctness

Let $(\mathbf{C}_1, \mathbf{c}_2)$ be a valid ciphertext. Consider the value $\mathbf{z}$ computed during decryption.

$$\mathbf{z} = \mathbf{c}_2 - \mathbf{C}_1 \cdot \mathbf{s}$$

# Correctness

Let $(\mathbf{C}_1, \mathbf{c}_2)$ be a valid ciphertext. Consider the value $\mathbf{z}$ computed during decryption.

$$\begin{aligned} \mathbf{z} &= \mathbf{c}_2 - \mathbf{C}_1 \cdot \mathbf{s} \\ &= \mathbf{Gm} + \mathbf{C}_1\mathbf{s} + \mathbf{e} - \mathbf{C}_1\mathbf{s} \end{aligned}$$

# Correctness

Let $(\mathbf{C}_1, \mathbf{c}_2)$ be a valid ciphertext. Consider the value $\mathbf{z}$ computed during decryption.

$$\begin{aligned}\mathbf{z} &= \mathbf{c}_2 - \mathbf{C}_1 \cdot \mathbf{s} \\ &= \mathbf{G}\mathbf{m} + \mathbf{C}_1\mathbf{s} + \mathbf{e} - \mathbf{C}_1\mathbf{s}\end{aligned}$$

## Correctness

Let $(\mathbf{C}_1, \mathbf{c}_2)$ be a valid ciphertext. Consider the value $\mathbf{z}$ computed during decryption.

$$\begin{aligned} \mathbf{z} &= \mathbf{c}_2 - \mathbf{C}_1 \cdot \mathbf{s} \\ &= \mathbf{Gm} + \mathbf{C}_1\mathbf{s} + \mathbf{e} - \mathbf{C}_1\mathbf{s} \\ &= \mathbf{Gm} + \underbrace{\mathbf{e}}_{\text{weight} \approx \rho m} \end{aligned}$$

Scheme is correct if decoding corrects $\rho n$ errors.

# Circular Security

Circular secure under standard (high noise) LPN

| Game | challenge ciphertext | remark |
|---|---|---|
| Real | $\mathbf{C}_1, \mathbf{C}_1\mathbf{s} + \mathbf{e} + \mathbf{G}\mathbf{s}$ | |

# Circular Security

Circular secure under standard (high noise) LPN

| Game | challenge ciphertext | remark |
|---|---|---|
| Real | $\mathbf{C}_1, \mathbf{C}_1\mathbf{s} + \mathbf{e} + \mathbf{G}\mathbf{s}$ | |
| Real | $\mathbf{C}_1, (\mathbf{C}_1 + \mathbf{G})\mathbf{s} + \mathbf{e}$ | |

## Circular Security

Circular secure under standard (high noise) LPN

| Game | challenge ciphertext | remark |
|---|---|---|
| Real | $\mathbf{C}_1, \mathbf{C}_1\mathbf{s} + \mathbf{e} + \mathbf{G}\mathbf{s}$ | |
| Real | $\mathbf{C}_1, (\mathbf{C}_1 + \mathbf{G})\mathbf{s} + \mathbf{e}$ | |
| Real | $\mathbf{C}_1 - \mathbf{G}, \mathbf{C}_1\mathbf{s} + \mathbf{e}$ | $\mathbf{C}_1$ uniform |

# Circular Security

Circular secure under standard (high noise) LPN

| Game | challenge ciphertext | remark |
|---|---|---|
| Real | $\mathbf{C}_1, \mathbf{C}_1\mathbf{s} + \mathbf{e} + \mathbf{G}\mathbf{s}$ | |
| Real | $\mathbf{C}_1, (\mathbf{C}_1 + \mathbf{G})\mathbf{s} + \mathbf{e}$ | |
| Real | $\mathbf{C}_1 - \mathbf{G}, \mathbf{C}_1\mathbf{s} + \mathbf{e}$ | $\mathbf{C}_1$ uniform |
| Ideal | $\mathbf{C}_1 - \mathbf{G}, \mathbf{u}$ | DLPN |

# Circular Security

Circular secure under standard (high noise) LPN

| Game | challenge ciphertext | remark |
|---|---|---|
| Real | $\mathbf{C}_1, \mathbf{C}_1\mathbf{s} + \mathbf{e} + \mathbf{G}\mathbf{s}$ | |
| Real | $\mathbf{C}_1, (\mathbf{C}_1 + \mathbf{G})\mathbf{s} + \mathbf{e}$ | |
| Real | $\mathbf{C}_1 - \mathbf{G}, \mathbf{C}_1\mathbf{s} + \mathbf{e}$ | $\mathbf{C}_1$ uniform |
| Ideal | $\mathbf{C}_1 - \mathbf{G}, \mathbf{u}$ | DLPN |
| Ideal | $\mathbf{U}, \mathbf{u}$ | |

# Our Scheme

- Turn this into a public key scheme using rerandomization.
- Make public key a rerandomizable LPN instance.

# Our Scheme

- Turn this into a public key scheme using rerandomization.
- Make public key a rerandomizable LPN instance.

# Our Scheme

- KeyGen: $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$, $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{m \times n}$,$\mathbf{e} \leftarrow_{\$} \mathsf{Ber}(\rho)^m$
  $\mathbf{y} = \mathbf{As} + \mathbf{e}$, $pk = (\mathbf{A}, \mathbf{y})$, $sk = \mathbf{s}$

# Our Scheme

- KeyGen: $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$, $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{m\times n}$,$\mathbf{e} \leftarrow_{\$} \mathsf{Ber}(\rho)^m$
  $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e}$, $pk = (\mathbf{A}, \mathbf{y})$, $sk = \mathbf{s}$
- $\mathsf{Enc}(pk, m)$: $\mathbf{R} \leftarrow_{\$} \mathsf{Ber}(\rho)^{k\times m}$, $\mathbf{C}_1 = \mathbf{R}\mathbf{A}$, $\mathbf{c}_2 = \mathbf{R}\mathbf{y} + \mathbf{G}\mathbf{m}$
  $c = (\mathbf{C}_1, \mathbf{c}_2)$

# Our Scheme

- KeyGen: $\mathbf{s} \leftarrow_{\$} \mathbb{F}_2^n$, $\mathbf{A} \leftarrow_{\$} \mathbb{F}_2^{m\times n}$, $\mathbf{e} \leftarrow_{\$} \mathrm{Ber}(\rho)^m$
  $\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e}$, $pk = (\mathbf{A}, \mathbf{y})$, $sk = \mathbf{s}$
- $\mathrm{Enc}(pk, m)$: $\mathbf{R} \leftarrow_{\$} \mathrm{Ber}(\rho)^{k\times m}$, $\mathbf{C}_1 = \mathbf{R}\mathbf{A}$, $\mathbf{c}_2 = \mathbf{R}\mathbf{y} + \mathbf{G}\mathbf{m}$
  $c = (\mathbf{C}_1, \mathbf{c}_2)$
- $\mathrm{Dec}(sk, c)$: $(\mathbf{C}_1, \mathbf{c}_2) = c$, $\mathbf{z} = \mathbf{c}_2 - \mathbf{C}_1 \cdot \mathbf{s}$
  $m = \mathrm{Decode}(\mathbf{z})$

# Circular Security

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}\mathbf{y} + \mathbf{G}\mathbf{s})$ | |

# Circular Security

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}\mathbf{y} + \mathbf{G}\mathbf{s})$ | |
| 2. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}(\mathbf{A}\mathbf{s} + \mathbf{e}) + \mathbf{G}\mathbf{s})$ | |

# Circular Security

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}\mathbf{y} + \mathbf{G}\mathbf{s})$ | |
| 2. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}(\mathbf{A}\mathbf{s} + \mathbf{e}) + \mathbf{G}\mathbf{s})$ | |
| 3. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, (\mathbf{R}\mathbf{A} + \mathbf{G})\mathbf{s} + \mathbf{R}\mathbf{e})$ | |

# Circular Security

$$\text{eDLPN: } (\mathbf{A}, \mathbf{RA}, \mathbf{e}, \mathbf{Re}) \approx_c (\mathbf{A}, \mathbf{U}, \mathbf{e}, \mathbf{Re})$$

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, \mathbf{Ry} + \mathbf{Gs})$ | |
| 2. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, \mathbf{R}(\mathbf{As} + \mathbf{e}) + \mathbf{Gs})$ | |
| 3. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, (\mathbf{RA} + \mathbf{G})\mathbf{s} + \mathbf{Re})$ | |
| 4. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{U}, (\mathbf{U} + \mathbf{G})\mathbf{s} + \mathbf{Re})$ | eDLPN |

# Circular Security

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}\mathbf{y} + \mathbf{G}\mathbf{s})$ | |
| 2. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}(\mathbf{A}\mathbf{s} + \mathbf{e}) + \mathbf{G}\mathbf{s})$ | |
| 3. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, (\mathbf{R}\mathbf{A} + \mathbf{G})\mathbf{s} + \mathbf{R}\mathbf{e})$ | |
| 4. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{U}, (\mathbf{U} + \mathbf{G})\mathbf{s} + \mathbf{R}\mathbf{e})$ | eDLPN |
| 5. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{U}' - \mathbf{G}, \mathbf{U}'\mathbf{s} + \mathbf{R}\mathbf{e})$ | |

# Circular Security

$$\text{eDLPN: } (\mathbf{A}, \mathbf{RA}, \mathbf{e}, \mathbf{Re}) \approx_c (\mathbf{A}, \mathbf{U}, \mathbf{e}, \mathbf{Re})$$

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, \mathbf{Ry} + \mathbf{Gs})$ | |
| 2. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, \mathbf{R}(\mathbf{As} + \mathbf{e}) + \mathbf{Gs})$ | |
| 3. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, (\mathbf{RA} + \mathbf{G})\mathbf{s} + \mathbf{Re})$ | |
| 4. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{U}, (\mathbf{U} + \mathbf{G})\mathbf{s} + \mathbf{Re})$ | eDLPN |
| 5. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{U}' - \mathbf{G}, \mathbf{U}'\mathbf{s} + \mathbf{Re})$ | |
| 6. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA} - \mathbf{G}, \mathbf{RAs} + \mathbf{Re})$ | eDLPN |

# Circular Security

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}\mathbf{y} + \mathbf{G}\mathbf{s})$ | |
| 2. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, \mathbf{R}(\mathbf{A}\mathbf{s} + \mathbf{e}) + \mathbf{G}\mathbf{s})$ | |
| 3. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A}, (\mathbf{R}\mathbf{A} + \mathbf{G})\mathbf{s} + \mathbf{R}\mathbf{e})$ | |
| 4. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{U}, (\mathbf{U} + \mathbf{G})\mathbf{s} + \mathbf{R}\mathbf{e})$ | eDLPN |
| 5. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{U}' - \mathbf{G}, \mathbf{U}'\mathbf{s} + \mathbf{R}\mathbf{e})$ | |
| 6. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A} - \mathbf{G}, \mathbf{R}\mathbf{A}\mathbf{s} + \mathbf{R}\mathbf{e})$ | eDLPN |
| 7. | $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$ | $(\mathbf{R}\mathbf{A} - \mathbf{G}, \mathbf{R}(\mathbf{A}\mathbf{s} + \mathbf{e}))$ | |

## Circular Security

$$\text{DLPN: } (\mathbf{A}, \mathbf{As}+\mathbf{e}) \approx_c (\mathbf{A}, \mathbf{u})$$

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y}=\mathbf{As}+\mathbf{e})$ | $(\mathbf{RA}, \mathbf{Ry}+\mathbf{Gs})$ | |
| 2. | $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ | $(\mathbf{RA}, \mathbf{R}(\mathbf{As}+\mathbf{e})+\mathbf{Gs})$ | |
| 3. | $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ | $(\mathbf{RA}, (\mathbf{RA}+\mathbf{G})\mathbf{s}+\mathbf{Re})$ | |
| 4. | $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ | $(\mathbf{U}, (\mathbf{U}+\mathbf{G})\mathbf{s}+\mathbf{Re})$ | eDLPN |
| 5. | $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ | $(\mathbf{U}'-\mathbf{G}, \mathbf{U}'\mathbf{s}+\mathbf{Re})$ | |
| 6. | $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ | $(\mathbf{RA}-\mathbf{G}, \mathbf{RAs}+\mathbf{Re})$ | eDLPN |
| 7. | $(\mathbf{A}, \mathbf{As}+\mathbf{e})$ | $(\mathbf{RA}-\mathbf{G}, \mathbf{R}(\mathbf{As}+\mathbf{e}))$ | |
| 8. | $(\mathbf{A}, \mathbf{u})$ | $(\mathbf{RA}-\mathbf{G}, \mathbf{Ru})$ | DLPN |

# Circular Security

eDLPN: $(\mathbf{A}, \mathbf{RA}, \mathbf{u}, \mathbf{Ru}) \approx (\mathbf{A}, \mathbf{U}, \mathbf{u}, \mathbf{Ru}) \approx_s (\mathbf{A}, \mathbf{U}, \mathbf{u}, \mathbf{u}')$

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, \mathbf{Ry} + \mathbf{Gs})$ | |
| 2. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, \mathbf{R}(\mathbf{As} + \mathbf{e}) + \mathbf{Gs})$ | |
| 3. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, (\mathbf{RA} + \mathbf{G})\mathbf{s} + \mathbf{Re})$ | |
| 4. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{U}, (\mathbf{U} + \mathbf{G})\mathbf{s} + \mathbf{Re})$ | eDLPN |
| 5. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{U}' - \mathbf{G}, \mathbf{U}'\mathbf{s} + \mathbf{Re})$ | |
| 6. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA} - \mathbf{G}, \mathbf{RAs} + \mathbf{Re})$ | eDLPN |
| 7. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA} - \mathbf{G}, \mathbf{R}(\mathbf{As} + \mathbf{e}))$ | |
| 8. | $(\mathbf{A}, \mathbf{u})$ | $(\mathbf{RA} - \mathbf{G}, \mathbf{Ru})$ | DLPN |
| 9. | $(\mathbf{A}, \mathbf{u})$ | $(\mathbf{U} - \mathbf{G}, \mathbf{u}')$ | eDLPN |

# Circular Security

| | public key | challenge ciphertext | remark |
|---|---|---|---|
| 1. | $(\mathbf{A}, \mathbf{y} = \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, \mathbf{Ry} + \mathbf{Gs})$ | |
| 2. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, \mathbf{R}(\mathbf{As} + \mathbf{e}) + \mathbf{Gs})$ | |
| 3. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA}, (\mathbf{RA} + \mathbf{G})\mathbf{s} + \mathbf{Re})$ | |
| 4. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{U}, (\mathbf{U} + \mathbf{G})\mathbf{s} + \mathbf{Re})$ | eDLPN |
| 5. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{U}' - \mathbf{G}, \mathbf{U}'\mathbf{s} + \mathbf{Re})$ | |
| 6. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA} - \mathbf{G}, \mathbf{RAs} + \mathbf{Re})$ | eDLPN |
| 7. | $(\mathbf{A}, \mathbf{As} + \mathbf{e})$ | $(\mathbf{RA} - \mathbf{G}, \mathbf{R}(\mathbf{As} + \mathbf{e}))$ | |
| 8. | $(\mathbf{A}, \mathbf{u})$ | $(\mathbf{RA} - \mathbf{G}, \mathbf{Ru})$ | DLPN |
| 9. | $(\mathbf{A}, \mathbf{u})$ | $(\mathbf{U} - \mathbf{G}, \mathbf{u}')$ | eDLPN |
| 10. | $(\mathbf{A}, \mathbf{u})$ | $(\mathbf{U}, \mathbf{u}')$ | |

# Remarks

- This scheme can be shown to be KDM-CPA secure (multi key setting) for affine functions under slightly stronger assumptions
- Specifically:

# Remarks

- This scheme can be shown to be KDM-CPA secure (multi key setting) for affine functions under slightly stronger assumptions
- Specifically:
  1. $\mathrm{DLPN}(n, \rho)$ for $\rho \approx 1/\sqrt{n}$

# Remarks

- This scheme can be shown to be KDM-CPA secure (multi key setting) for affine functions under slightly stronger assumptions
- Specifically:
  1. DLPN$(n, \rho)$ for $\rho \approx 1/\sqrt{n}$
  2. Or: DLPN$(n, 2n, \rho)$ for $\rho \approx 1/n^{3/4}$ (implies 1 by first result)

## Remarks

- This scheme can be shown to be KDM-CPA secure (multi key setting) for affine functions under slightly stronger assumptions
- Specifically:
    1. $\mathrm{DLPN}(n, \rho)$ for $\rho \approx 1/\sqrt{n}$
    2. Or: $\mathrm{DLPN}(n, 2n, \rho)$ for $\rho \approx 1/n^{3/4}$ (implies 1 by first result)

# Conclusion

- Computational LPN rerandomization via extended/leaky LPN is a powerful tool
- LPN with unbounded samples implied by LPN with few samples (smaller noise)

## Conclusion

- Computational LPN rerandomization via extended/leaky LPN is a powerful tool
- LPN with unbounded samples implied by LPN with few samples (smaller noise)
- Same technique yields KDM secure public key encryption

## Conclusion

- Computational LPN rerandomization via extended/leaky LPN is a powerful tool
- LPN with unbounded samples implied by LPN with few samples (smaller noise)
- Same technique yields KDM secure public key encryption
- Further applications for this technique?

# Conclusion

- Computational LPN rerandomization via extended/leaky LPN is a powerful tool
- LPN with unbounded samples implied by LPN with few samples (smaller noise)
- Same technique yields KDM secure public key encryption
- Further applications for this technique?

# Conclusion

- Computational LPN rerandomization via extended/leaky LPN is a powerful tool
- LPN with unbounded samples implied by LPN with few samples (smaller noise)
- Same technique yields KDM secure public key encryption
- Further applications for this technique?

# Thank You!