Peculiar Properties of Lattice-Based Encryption

Chris Peikert
Georgia Institute of Technology

Public Key Cryptography
and the Geometry of Numbers

7 May 2010

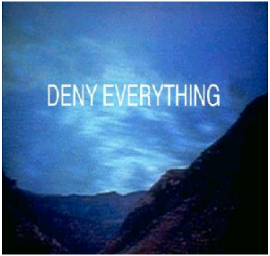
Talk Agenda

Encryption schemes with special features:

Talk Agenda

Encryption schemes with special features:

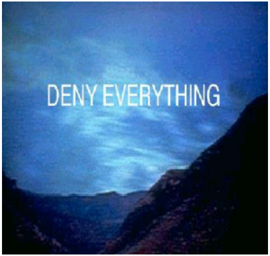
1. "(Bi-)Deniability"



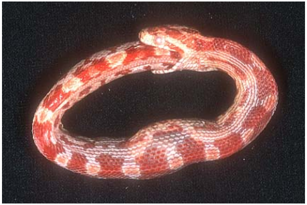
Talk Agenda

Encryption schemes with special features:

1. "(Bi-)Deniability"



2. "Circular" Security



Part 1:

Deniable Encryption

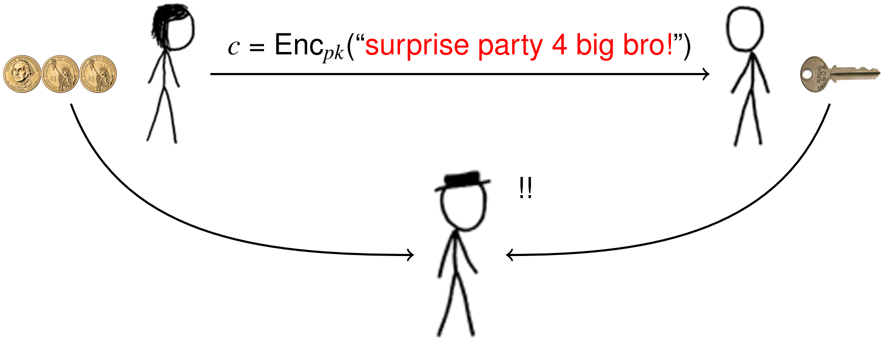
- A. O'Neill, C. Peikert (2010)
  "Bideniable Public-Key Encryption"

Deniable Encryption


c = Enc$_{pk}$("surprise party 4 big bro!")

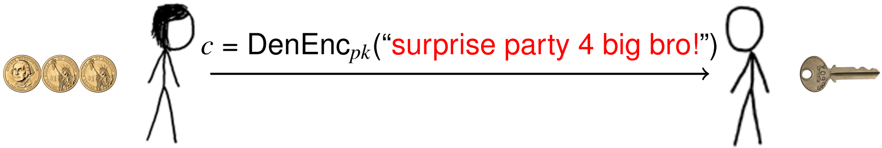

(Images courtesy xkcd.org)

Deniable Encryption



(Images courtesy xkcd.org)

Deniable Encryption



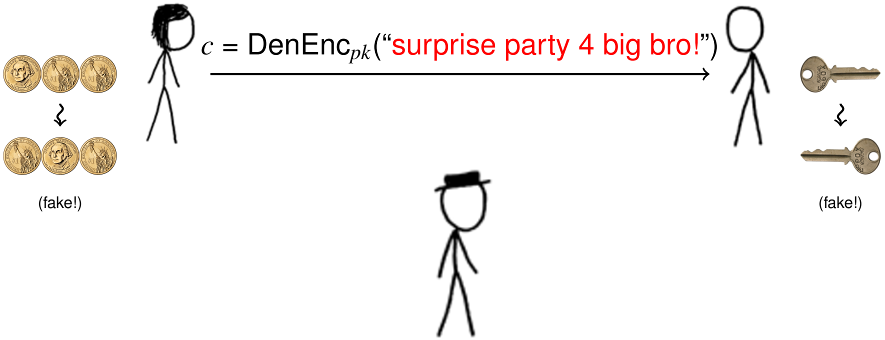
c = DenEnc_{pk}(“surprise party 4 big bro!”)

What We Want

1. Bob gets Alice’s intended message, but …

(Images courtesy xkcd.org)

Deniable Encryption

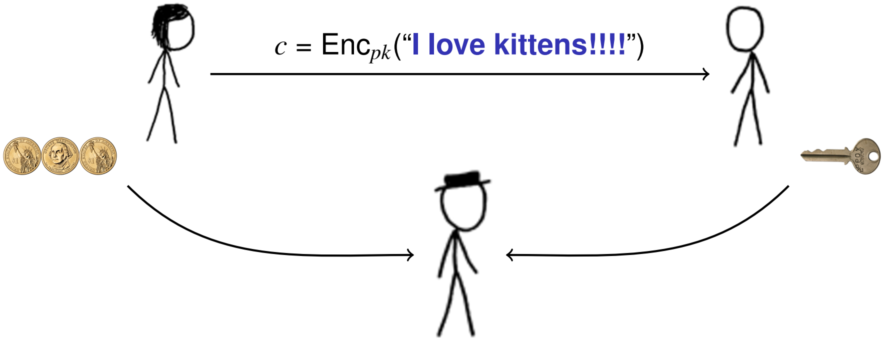


What We Want

1. Bob gets Alice's intended message, but . . .

(Images courtesy xkcd.org)

Deniable Encryption



What We Want

1. Bob gets Alice's intended message, but . . .
2. Fake coins & keys 'look as if' another message was encrypted!

(Images courtesy xkcd.org)

Applications of Deniability

1. Anti-coercion: ‘off the record’ communication (journalists, lawyers, whistle-blowers), 1984

Applications of Deniability

1. Anti-coercion: ‘off the record’ communication (journalists, lawyers, whistle-blowers), 1984

2. Voting: can reveal *any* candidate, so can’t ‘sell’ vote (?)

Applications of Deniability

1. Anti-coercion: ‘off the record’ communication (journalists, lawyers, whistle-blowers), 1984
2. Voting: can reveal *any* candidate, so can’t ‘sell’ vote (?)
3. Secure protocols tolerating *adaptive* break-ins [CFGN’96]

State of the Art

Theory [CanettiDworkNaorOstrovsky'97]

- Sender-deniable encryption scheme
- Receiver-deniability by adding interaction & switching roles
- Bi-deniability by interaction w/ 3rd parties (one must remain uncoerced)

State of the Art

Theory [CanettiDworkNaorOstrovsky'97]

- Sender-deniable encryption scheme
- Receiver-deniability by adding interaction & switching roles
- Bi-deniability by interaction w/ 3rd parties (one must remain uncoerced)

Practice: TrueCrypt, Rubberhose, ...

- Limited deniability: "*move along, no message here...*"

  Plausible for *storage*, but not so much for *communication*.

This Work

1. Bi-deniable encryption: sender & receiver *simultaneously* coercible

This Work

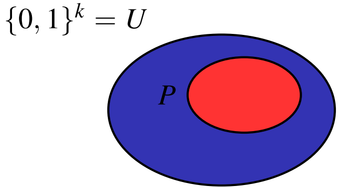
1. Bi-deniable encryption: sender & receiver *simultaneously* coercible
   - A true public-key scheme: non-interactive, no 3rd parties
   - Uses special properties of lattices [Ajtai'96,Regev'05,GPV'08,...]
   - Has large keys ... but this is inherent [Nielsen'02]

This Work

1. Bi-deniable encryption: sender & receiver *simultaneously* coercible
   - A true public-key scheme: non-interactive, no 3rd parties
   - Uses special properties of lattices [Ajtai'96,Regev'05,GPV'08,...]
   - Has large keys ...but this is inherent [Nielsen'02]

2. "Plan-ahead" bi-deniability with *short* keys
   - Bounded number of alternative messages, decided in advance

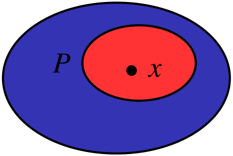
A Core Tool: Translucent Sets [CDNO'97]



Public description pk with secret 'trapdoor' sk.

A Core Tool: Translucent Sets [CDNO'97]

$\{0,1\}^k = U$



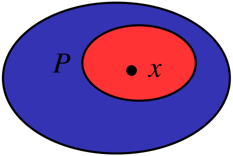
Public description pk with secret 'trapdoor' sk.

Properties

1. Given only pk,
   - ⋆ Can efficiently sample from P (and from U, trivially).
   - ⋆ P-sample is pseudorandom: 'looks like' a U-sample…
   - ⋆ …so it can be 'faked' as a U-sample.

A Core Tool: Translucent Sets [CDNO'97]

$\{0,1\}^k = U$



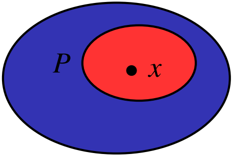
Public description pk with secret 'trapdoor' sk.

Properties

1. Given only pk,
   - ⋆ Can efficiently sample from P (and from U, trivially).
   - ⋆ P-sample is pseudorandom: 'looks like' a U-sample…
   - ⋆ …so it can be 'faked' as a U-sample.
2. Given sk, can easily distinguish P from U.

A Core Tool: Translucent Sets [CDNO'97]

$\{0,1\}^k = U$



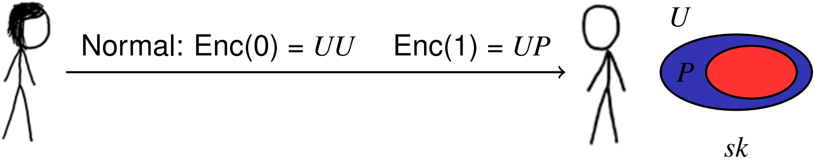
Public description pk with secret 'trapdoor' sk.

Properties

1. Given only pk,
   - ⋆ Can efficiently sample from P (and from U, trivially).
   - ⋆ P-sample is pseudorandom: 'looks like' a U-sample...
   - ⋆ ...so it can be 'faked' as a U-sample.
2. Given sk, can easily distinguish P from U.

- Many instantiations: trapdoor perms (RSA), DDH, lattices, ...

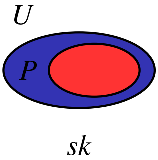
Translucence for Deniability [CDNO'97]



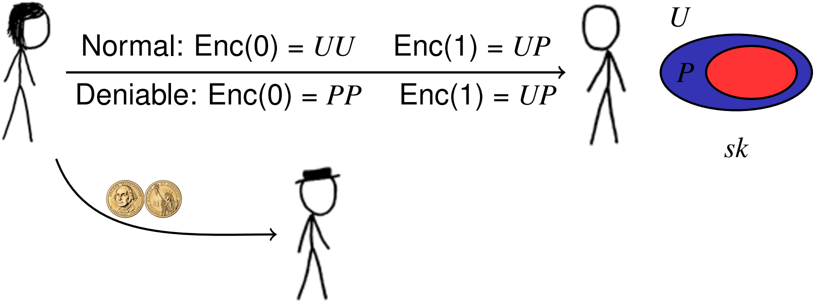
Translucence for Deniability [CDNO'97]

Normal: Enc(0) = UU Enc(1) = UP

Deniable: Enc(0) = PP Enc(1) = UP



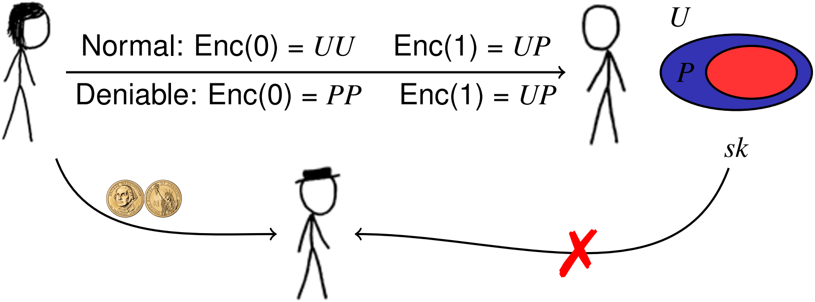
Translucence for Deniability [CDNO'97]



Deniability

✔ Alice can fake: $PP \to UP \to UU$

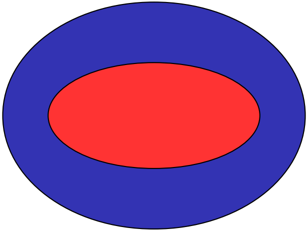
Translucence for Deniability [CDNO'97]



Deniability

- ✔ Alice can fake: $PP \to UP \to UU$
- ✗ What about Bob?? His sk reveals the true nature of the samples!

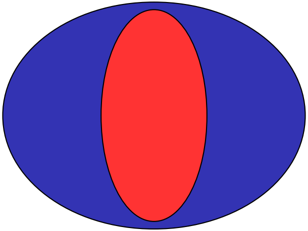
Our Contribution: Bi-Translucent Sets



Properties

1. Each pk has many sk, each inducing a *slightly different* P-test.

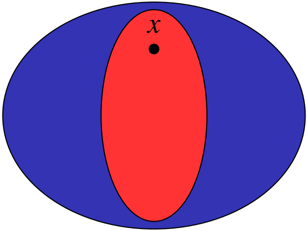
Our Contribution: Bi-Translucent Sets



Properties

1. Each pk has many sk, each inducing a *slightly different* P-test.

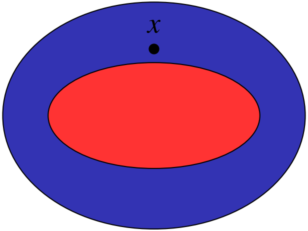
Our Contribution: Bi-Translucent Sets



Properties

1. Each pk has many sk, each inducing a *slightly different* P-test.
2. Most sk classify a given P-sample correctly.

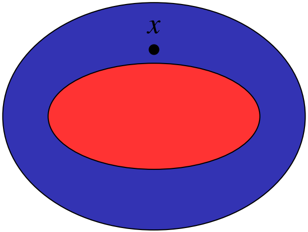
Our Contribution: Bi-Translucent Sets



Properties

1. Each pk has many sk, each inducing a *slightly different* P-test.
2. Most sk classify a given P-sample correctly.
3. Can generate pk with a faking key: given fk and a P-sample x, can find a 'proper-looking' sk that classifies x as a U-sample.

Our Contribution: Bi-Translucent Sets

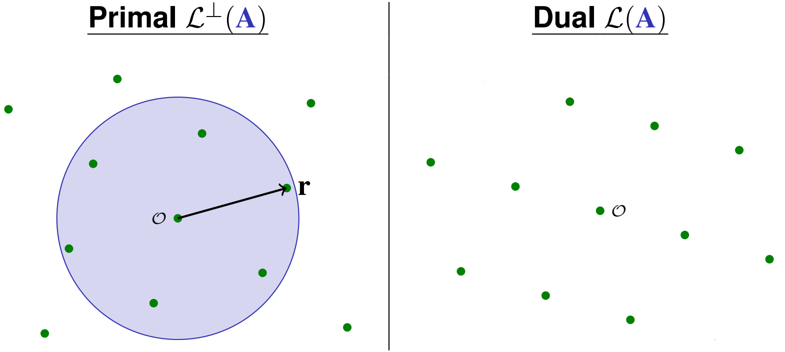


Properties

1. Each pk has many sk, each inducing a *slightly different* P-test.
2. Most sk classify a given P-sample correctly.
3. Can generate pk with a faking key: given fk and a P-sample x, can find a 'proper-looking' sk that classifies x as a U-sample.

$\Rightarrow$ Bob can also fake $P \to U$!

Lattice-Based Bi-Translucent Set

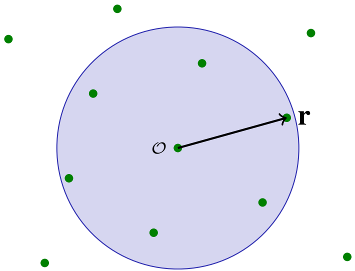


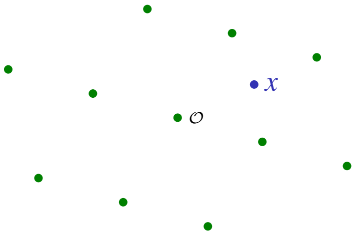
Basic Translucency

- pk = parity check $\mathbf{A}$ of lattice $\mathcal{L}^{\perp}(\mathbf{A})$.
- sk = Gaussian (short) vector $\mathbf{r} \in \mathcal{L}^{\perp}$. (I.e., $\mathbf{A}\mathbf{r} = \mathbf{0} \in \mathbb{Z}_q^n$.)

Lattice-Based Bi-Translucent Set

Primal $\mathcal{L}^{\perp}(\mathbf{A})$ | **Dual $\mathcal{L}(\mathbf{A})$**

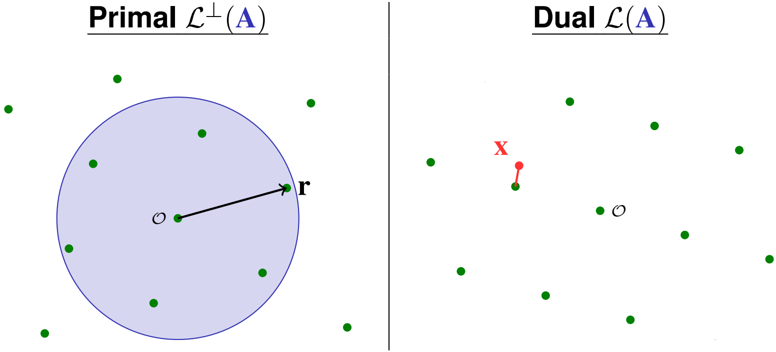




Basic Translucency

- pk = parity check $\mathbf{A}$ of lattice $\mathcal{L}^{\perp}(\mathbf{A})$.
- sk = Gaussian (short) vector $\mathbf{r} \in \mathcal{L}^{\perp}$. (I.e., $\mathbf{A}\mathbf{r} = \mathbf{0} \in \mathbb{Z}_q^n$.)
- U-sample = uniform $\mathbf{x}$ in $\mathbb{Z}_q^m$. Then $\langle \mathbf{r}, \mathbf{x} \rangle$ is uniform mod q.

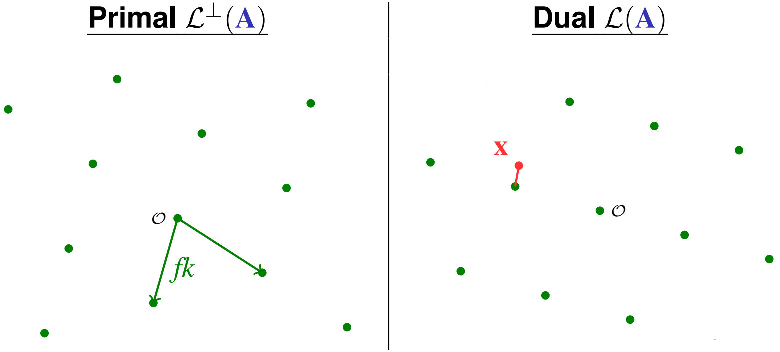
Lattice-Based Bi-Translucent Set



Basic Translucency

- pk = parity check $\mathbf{A}$ of lattice $\mathcal{L}^{\perp}(\mathbf{A})$.
- sk = Gaussian (short) vector $\mathbf{r} \in \mathcal{L}^{\perp}$. (I.e., $\mathbf{Ar} = \mathbf{0} \in \mathbb{Z}_q^n$.)
- U-sample = uniform $\mathbf{x}$ in $\mathbb{Z}_q^m$. Then $\langle \mathbf{r}, \mathbf{x} \rangle$ is uniform mod q.
- P-sample = $\mathbf{x} = \mathbf{A}^t\mathbf{s} + \mathbf{e}$ (LWE). Then $\langle \mathbf{r}, \mathbf{x} \rangle \approx 0 \bmod q$.

Lattice-Based Bi-Translucent Set

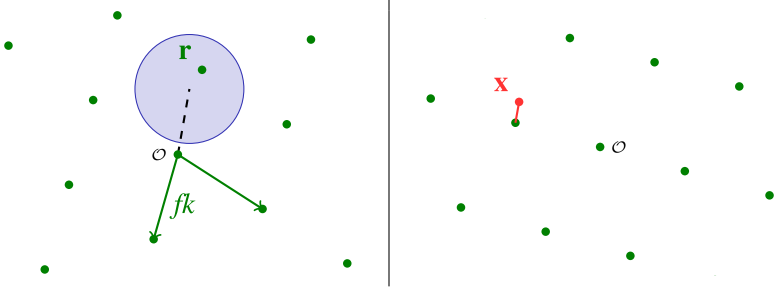


Receiver Faking

- Faking key = short *basis* of $\mathcal{L}^{\perp}$ (a la [GPV'08,. . .])

Lattice-Based Bi-Translucent Set

Primal $\mathcal{L}^{\perp}(\mathbf{A})$ | **Dual** $\mathcal{L}(\mathbf{A})$

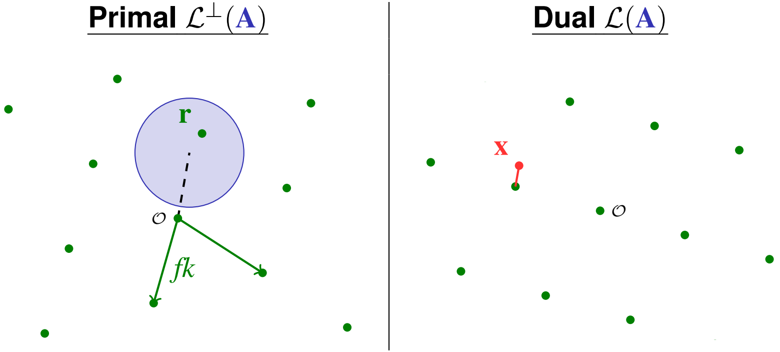


Receiver Faking

- Faking key = short *basis* of $\mathcal{L}^{\perp}$ (a la [GPV'08,...])
- Given P-sample $\mathbf{x}$, choose fake $\mathbf{r} \in \mathcal{L}^{\perp}$ *correlated* with $\mathbf{x}$'s error.

  Then $\langle \mathbf{r}, \mathbf{x} \rangle$ is uniform mod $q \Rightarrow \mathbf{x}$ is classified as a U-sample.

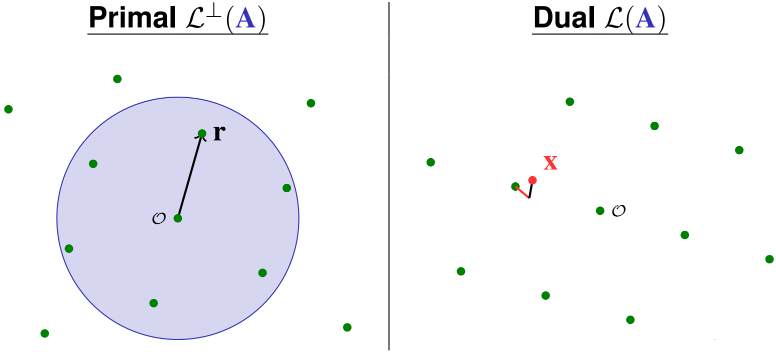
Lattice-Based Bi-Translucent Set



Security (in a nutshell)

- Fake $\mathbf{r}$ depends heavily on $\mathbf{x}$. Why would it 'look like' a 'normal' $\mathbf{r}$?

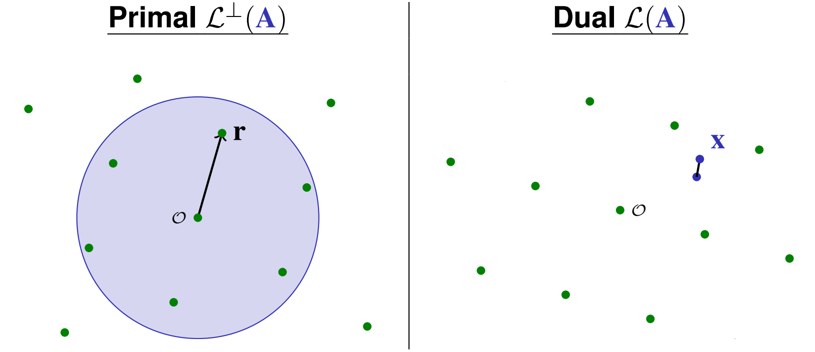
Lattice-Based Bi-Translucent Set



Security (in a nutshell)

- Fake $\mathbf{r}$ depends heavily on $\mathbf{x}$. Why would it 'look like' a 'normal' $\mathbf{r}$?
- Alternative experiment: choose Gaussian $\mathbf{r}$ (as normal), then let $\mathbf{x} = \text{LWE} + \text{Gauss} \cdot \mathbf{r}$. This $(\mathbf{r}, \mathbf{x})$ has *the same** joint distrib!

Lattice-Based Bi-Translucent Set



Security (in a nutshell)

- Fake $\mathbf{r}$ depends heavily on $\mathbf{x}$. Why would it 'look like' a 'normal' $\mathbf{r}$?
- Alternative experiment: choose Gaussian $\mathbf{r}$ (as normal), then let $\mathbf{x} =$ LWE $+$ Gauss $\cdot\, \mathbf{r}$. This $(\mathbf{r}, \mathbf{x})$ has *the same** joint distrib!
- Finally, replace LWE with uniform $\Rightarrow$ normal $\mathbf{r}$ and U-sample $\mathbf{x}$.

Closing Thoughts on Deniability

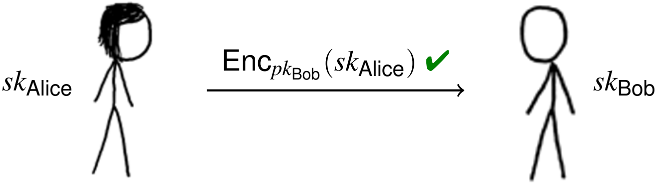
- Faking sk requires ‘oblivious’ misclassification (of P as U)
- Bi-deniability from other cryptographic assumptions?
- Full deniability, without alternative algorithms?

Part 2:

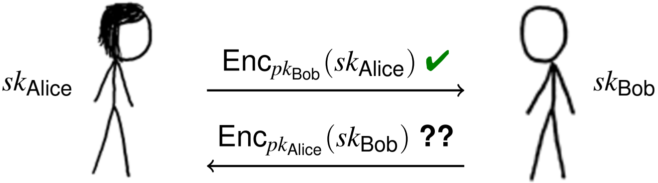
Circular-Secure Encryption

- B. Applebaum, D. Cash, C. Peikert, A. Sahai (CRYPTO 2009)
  "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems"

Circular / “Clique” / Key-Dependent Security

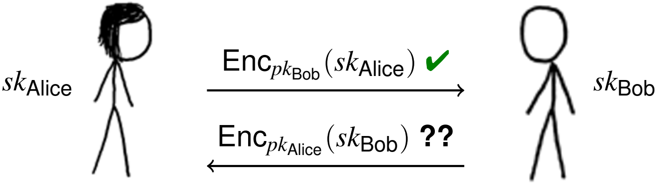


Circular / "Clique" / Key-Dependent Security



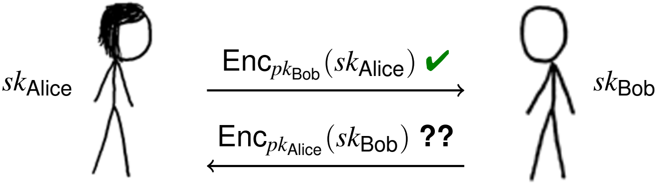
- "Semantic security" [GM'02] only guarantees security for messages that the *adversary can itself generate*.

Circular / “Clique” / Key-Dependent Security



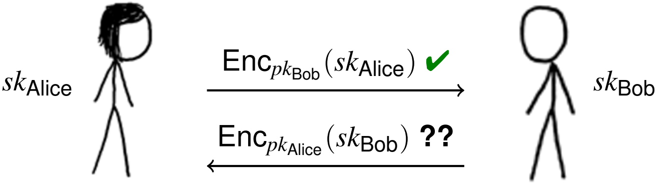
- “Semantic security” [GM’02] only guarantees security for messages that the *adversary can itself generate*.
  - ⋆ $\mathcal{F}$-KDM security: adversary also gets $\text{Enc}_{pk}(f(sk))$ for any $f \in \mathcal{F}$
  - ⋆ Clique security: adversary gets $\text{Enc}_{pk_i}(f(sk_j))$ for any i, j

Circular / “Clique” / Key-Dependent Security



- “Semantic security” [GM’02] only guarantees security for messages that the *adversary can itself generate*.
  - ⋆ $\mathcal{F}$-KDM security: adversary also gets $\text{Enc}_{pk}(f(sk))$ for any $f \in \mathcal{F}$
  - ⋆ Clique security: adversary gets $\text{Enc}_{pk_i}(f(sk_j))$ for any i, j
- Applications: formal analysis [ABHS’05], disk encryption, anonymity systems [CL’01], fully homomorphic encryption [G’09]

Circular / “Clique” / Key-Dependent Security



- “Semantic security” [GM’02] only guarantees security for messages that the *adversary can itself generate*.
  - ⋆ $\mathcal{F}$-KDM security: adversary also gets $\text{Enc}_{pk}(f(sk))$ for any $f \in \mathcal{F}$
  - ⋆ Clique security: adversary gets $\text{Enc}_{pk_i}(f(sk_j))$ for any i, j
- Applications: formal analysis [ABHS’05], disk encryption, anonymity systems [CL’01], fully homomorphic encryption [G’09]
- Some (semantically secure) schemes are actually circular-*insecure* [ABBC’10,GH’10]

Solutions

[Boneh-Halevi-Hamburg-Ostrovsky'08]

- Based on decisional Diffie-Hellman (DDH) assumption

Solutions

[Boneh-Halevi-Hamburg-Ostrovsky'08]

- Based on decisional Diffie-Hellman (DDH) assumption

Our Scheme [Applebaum-Cash-P-Sahai'09]

- Based on Learning With Errors (LWE) assumption [Regev'05]

Solutions

[Boneh-Halevi-Hamburg-Ostrovsky'08]

- Based on decisional Diffie-Hellman (DDH) assumption
- Security: Clique & KDM for affine functions

Our Scheme [Applebaum-Cash-P-Sahai'09]

- Based on Learning With Errors (LWE) assumption [Regev'05]
- Security: same. Follows general [BHHO'08] approach.

Solutions

[Boneh-Halevi-Hamburg-Ostrovsky'08]

- Based on decisional Diffie-Hellman (DDH) assumption
- Security: Clique & KDM for affine functions
- Large computation & communication. For k-bit message:

Public key	Enc Time	Ciphertext
k^2 group elts $\Downarrow$ k^3 bits	k expon $\Downarrow$ k^4 bit ops	$\geq k$ group elts $\Downarrow$ $\geq k^2$ bits

Our Scheme [Applebaum-Cash-P-Sahai'09]

- Based on Learning With Errors (LWE) assumption [Regev'05]
- Security: same. Follows general [BHHO'08] approach.

Solutions

[Boneh-Halevi-Hamburg-Ostrovsky'08]

- Based on decisional Diffie-Hellman (DDH) assumption
- Security: Clique & KDM for affine functions
- Large computation & communication. For k-bit message:

Public key	Enc Time	Ciphertext
k^2 group elts	k expon	$\geq k$ group elts
$\Downarrow$	$\Downarrow$	$\Downarrow$
k^3 bits	k^4 bit ops	$\geq k^2$ bits

Our Scheme [Applebaum-Cash-P-Sahai'09]

- Based on Learning With Errors (LWE) assumption [Regev'05]
- Security: same. Follows general [BHHO'08] approach.
- Efficiency: comes 'for free*' with existing schemes! [R'05,PVW'08]

Public key	Enc Time	Ciphertext
$\sim k^2$ bits	$\sim k^2$ ops	$\sim k$ bits

Regev's Cryptosystem

- Decision LWE problem: distinguish samples

$$(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \quad \text{from} \quad \text{uniform } (\mathbf{a}_i\ ,\ b_i)$$

Regev's Cryptosystem

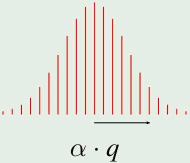
- Decision LWE problem: distinguish samples

$$(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \quad \text{from} \quad \text{uniform } (\mathbf{a}_i\ ,\ b_i)$$

The Scheme

- Keys: $sk = \mathbf{s} \leftarrow \mathbb{Z}_q^n$,

$$pk = \left(\begin{matrix} \vdots \\ \mathbf{A}^t \\ \vdots \end{matrix} \right) \ , \ \left(\begin{matrix} \vdots \\ \mathbf{b} \\ \vdots \end{matrix} \right) = \mathbf{A}^t \mathbf{s} + \mathbf{e}$$



Regev's Cryptosystem

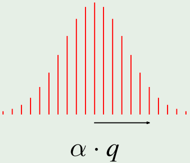
- Decision LWE problem: distinguish samples

$$(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \quad \text{from} \quad \text{uniform } (\mathbf{a}_i\ ,\ b_i)$$

The Scheme

- Keys: $sk = \mathbf{s} \leftarrow \mathbb{Z}_q^n$,

$$pk = \begin{pmatrix} \vdots \\ \mathbf{A}^t \\ \vdots \end{pmatrix}\ ,\ \begin{pmatrix} \vdots \\ \mathbf{b} \\ \vdots \end{pmatrix} = \mathbf{A}^t\mathbf{s} + \mathbf{e}$$



- Encrypt: Let $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$ for $\mathbf{r} \leftarrow \{0,1\}^m$.
  For message $\mu \in \mathbb{Z}_p$ (where $p \ll q$), ciphertext = $(\mathbf{u}\ ,\ v + \mu \cdot \lfloor \frac{q}{p} \rceil)$.

Regev's Cryptosystem

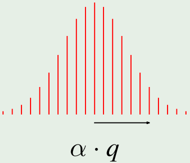
- Decision LWE problem: distinguish samples

$$(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \quad \text{from} \quad \text{uniform } (\mathbf{a}_i\ ,\ b_i)$$

The Scheme

- Keys: $sk = \mathbf{s} \leftarrow \mathbb{Z}_q^n$,

$$pk = \left(\begin{array}{c} \vdots \\ \mathbf{A}^t \\ \vdots \end{array} \right) \ , \ \left(\begin{array}{c} \vdots \\ \mathbf{b} \\ \vdots \end{array} \right) = \mathbf{A}^t\mathbf{s} + \mathbf{e}$$



- Encrypt: Let $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$ for $\mathbf{r} \leftarrow \{0,1\}^m$.
  For message $\mu \in \mathbb{Z}_p$ (where $p \ll q$), ciphertext = $(\mathbf{u}\ ,\ v + \mu \cdot \lfloor \frac{q}{p} \rfloor)$.
- Decrypt $(\mathbf{u}, v')$: find the $\mu \in \mathbb{Z}_p$ such that $v' - \langle \mathbf{u}, \mathbf{s} \rangle \approx \mu \cdot \lfloor \frac{q}{p} \rfloor$.

Regev's Cryptosystem

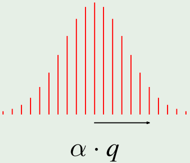
- Decision LWE problem: distinguish samples

$$(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s}\rangle + e_i) \in \mathbb{Z}_q^n \times \mathbb{Z}_q \quad \text{from} \quad \text{uniform } (\mathbf{a}_i\ ,\ b_i)$$

The Scheme

- Keys: $sk = \mathbf{s} \leftarrow \mathbb{Z}_q^n$,

$$pk = \begin{pmatrix} \vdots \\ \mathbf{A}^t \\ \vdots \end{pmatrix}, \quad \begin{pmatrix} \vdots \\ \mathbf{b} \\ \vdots \end{pmatrix} = \mathbf{A}^t\mathbf{s} + \mathbf{e}$$



- Encrypt: Let $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r}\rangle)$ for $\mathbf{r} \leftarrow \{0,1\}^m$.
  For message $\mu \in \mathbb{Z}_p$ (where $p \ll q$), ciphertext = $(\mathbf{u}\ ,\ v + \mu \cdot \lfloor \frac{q}{p} \rceil)$.
- Decrypt $(\mathbf{u}, v')$: find the $\mu \in \mathbb{Z}_p$ such that $v' - \langle \mathbf{u}, \mathbf{s}\rangle \approx \mu \cdot \lfloor \frac{q}{p} \rceil$.
- Security proof: uniform $pk = (\mathbf{A}, \mathbf{b}) \implies$ uniform ciphertext $(\mathbf{u}, v)$.

Self-Reference ?

An Observation

- With $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$, the ciphertext $(\mathbf{u}' = \mathbf{u} - \lfloor \frac{q}{p} \rfloor \cdot \mathbf{e}_1\ ,\ v)$ decrypts as $v - \langle \mathbf{u}', \mathbf{s} \rangle \approx (s_1 \bmod p) \cdot \lfloor \frac{q}{p} \rfloor$. (Or any affine fct of $\mathbf{s}$.)

Self-Reference ?

An Observation

- With $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$, the ciphertext $(\mathbf{u}' = \mathbf{u} - \lfloor \frac{q}{p} \rfloor \cdot \mathbf{e}_1\ ,\ v)$ decrypts as $v - \langle \mathbf{u}', \mathbf{s} \rangle \ \approx\ (s_1 \bmod p) \cdot \lfloor \frac{q}{p} \rfloor$.
- But: is $(\mathbf{u}', v)$ distributed the same as $(\mathbf{u}, v') \leftarrow \text{Enc}(s_1 \bmod p)$? And does $s_1 \in \mathbb{Z}_q$ 'fit' into the message space $\mathbb{Z}_p$?

Self-Reference ?

An Observation

- With $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$, the ciphertext $(\mathbf{u}' = \mathbf{u} - \lfloor \frac{q}{p} \rceil \cdot \mathbf{e}_1\ ,\ v)$ decrypts as $v - \langle \mathbf{u}', \mathbf{s} \rangle \approx (s_1 \bmod p) \cdot \lfloor \frac{q}{p} \rceil$.
- But: is $(\mathbf{u}', v)$ distributed the same as $(\mathbf{u}, v') \leftarrow \mathsf{Enc}(s_1 \bmod p)$? No!
  And does $s_1 \in \mathbb{Z}_q$ 'fit' into the message space $\mathbb{Z}_p$? Also no!

Self-Reference !

An Observation

- With $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$, the ciphertext $(\mathbf{u}' = \mathbf{u} - \lfloor \frac{q}{p} \rceil \cdot \mathbf{e}_1\ ,\ v)$ decrypts as $v - \langle \mathbf{u}', \mathbf{s} \rangle \ \approx\ (s_1 \bmod p) \cdot \lfloor \frac{q}{p} \rceil$.
- But: is $(\mathbf{u}', v)$ distributed the same as $(\mathbf{u}, v') \leftarrow \mathsf{Enc}(s_1 \bmod p)$? No!
  And does $s_1 \in \mathbb{Z}_q$ 'fit' into the message space $\mathbb{Z}_p$? Also no!

Modifying the Scheme

1. Use $q = p^2$ for divisibility. (Need new search/decision reduction for LWE.)

Self-Reference !

An Observation

- With $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$, the ciphertext $(\mathbf{u}' = \mathbf{u} - \lfloor \frac{q}{p} \rceil \cdot \mathbf{e}_1\ ,\ v)$ decrypts as $v - \langle \mathbf{u}', \mathbf{s} \rangle \ \approx\ (s_1 \bmod p) \cdot \lfloor \frac{q}{p} \rceil$.
- But: is $(\mathbf{u}', v)$ distributed the same as $(\mathbf{u}, v') \leftarrow \mathsf{Enc}(s_1 \bmod p)$? No!
  And does $s_1 \in \mathbb{Z}_q$ 'fit' into the message space $\mathbb{Z}_p$? Also no!

Modifying the Scheme

1. Use $q = p^2$ for divisibility.
2. Give $(\mathbf{u}, v)$ a 'nice' distrib: use $\mathbf{r} \leftarrow \mathsf{Gaussian}(\mathbb{Z}^m)$.
   Then $(\mathbf{u}, v)$ is *itself* an $\mathsf{LWE}_{\mathbf{s}}$ sample*. [R'05,GPV'08]

Self-Reference !

An Observation

- With ($\mathbf{u} = \mathbf{A}\mathbf{r}$, $v = \langle \mathbf{b}, \mathbf{r} \rangle$), the ciphertext ($\mathbf{u}' = \mathbf{u} - \lfloor \frac{q}{p} \rceil \cdot \mathbf{e}_1$, v) decrypts as $v - \langle \mathbf{u}', \mathbf{s} \rangle \approx (s_1 \bmod p) \cdot \lfloor \frac{q}{p} \rceil$.
- But: is $(\mathbf{u}', v)$ distributed the same as $(\mathbf{u}, v') \leftarrow$ Enc$(s_1 \bmod p)$? No!
  And does $s_1 \in \mathbb{Z}_q$ 'fit' into the message space $\mathbb{Z}_p$? Also no!

Modifying the Scheme

1. Use $q = p^2$ for divisibility.
2. Give $(\mathbf{u}, v)$ a 'nice' distrib: use $\mathbf{r} \leftarrow$ Gaussian$(\mathbb{Z}^m)$.
   Then $(\mathbf{u}, v)$ is *itself* an LWE$_\mathbf{s}$ sample*. [R'05,GPV'08]
   (And for security, $(\mathbf{u}, v)$ is still uniform* when $(\mathbf{A}, \mathbf{b})$ is uniform.)

Self-Reference !

An Observation

- With $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$, the ciphertext $(\mathbf{u}' = \mathbf{u} - \lfloor \frac{q}{p} \rceil \cdot \mathbf{e}_1\ ,\ v)$ decrypts as $v - \langle \mathbf{u}', \mathbf{s} \rangle \approx (s_1 \bmod p) \cdot \lfloor \frac{q}{p} \rceil$.
- But: is $(\mathbf{u}', v)$ distributed the same as $(\mathbf{u}, v') \leftarrow \text{Enc}(s_1 \bmod p)$? No!
  And does $s_1 \in \mathbb{Z}_q$ 'fit' into the message space $\mathbb{Z}_p$? Also no!

Modifying the Scheme

1. Use $q = p^2$ for divisibility.
2. Give $(\mathbf{u}, v)$ a 'nice' distrib: use $\mathbf{r} \leftarrow \text{Gaussian}(\mathbb{Z}^m)$.
   Then $(\mathbf{u}, v)$ is *itself* an $\text{LWE}_{\mathbf{s}}$ sample*. [R'05,GPV'08]
   (And for security, $(\mathbf{u}, v)$ is still uniform* when $(\mathbf{A}, \mathbf{b})$ is uniform.)
3. Use a Gaussian secret $\mathbf{s}$, so each $s_i \in (-\frac{p}{2}, \frac{p}{2})$: self-reference!

Self-Reference !

An Observation

- With $(\mathbf{u} = \mathbf{A}\mathbf{r}\ ,\ v = \langle \mathbf{b}, \mathbf{r} \rangle)$, the ciphertext $(\mathbf{u}' = \mathbf{u} - \lfloor \frac{q}{p} \rfloor \cdot \mathbf{e}_1\ ,\ v)$ decrypts as $v - \langle \mathbf{u}', \mathbf{s} \rangle \ \approx\ (s_1 \bmod p) \cdot \lfloor \frac{q}{p} \rfloor$.
- But: is $(\mathbf{u}', v)$ distributed the same as $(\mathbf{u}, v') \leftarrow \mathsf{Enc}(s_1 \bmod p)$? No!
  And does $s_1 \in \mathbb{Z}_q$ 'fit' into the message space $\mathbb{Z}_p$? Also no!

Modifying the Scheme

1. Use $q = p^2$ for divisibility.
2. Give $(\mathbf{u}, v)$ a 'nice' distrib: use $\mathbf{r} \leftarrow \mathsf{Gaussian}(\mathbb{Z}^m)$.
   Then $(\mathbf{u}, v)$ is *itself* an $\text{LWE}_{\mathbf{s}}$ sample*. [R'05,GPV'08]
   (And for security, $(\mathbf{u}, v)$ is still uniform* when $(\mathbf{A}, \mathbf{b})$ is uniform.)
3. Use a Gaussian secret $\mathbf{s}$, so each $s_i \in (-\frac{p}{2}, \frac{p}{2})$: self-reference!
   ?? But is it secure to use such an $\mathbf{s}$??

LWE with Gaussian Secret

- Transform $\text{LWE}_{\mathbf{s}}$ (for arbitrary $\mathbf{s}$) into $\text{LWE}_{\mathbf{e}}$ for Gaussian secret $\mathbf{e}$:

  Given the source $\text{LWE}_{\mathbf{s}}$ of samples $(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$,

LWE with Gaussian Secret

- Transform LWE$_{\mathbf{s}}$ (for arbitrary $\mathbf{s}$) into LWE$_{\mathbf{e}}$ for Gaussian secret $\mathbf{e}$:

  Given the source LWE$_{\mathbf{s}}$ of samples $(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$,

  1. Draw n samples $(\mathbf{A}\ ,\ \mathbf{b} = \mathbf{A}^t\mathbf{s} + \mathbf{e})$ so that $\mathbf{A}$ is invertible mod q.

LWE with Gaussian Secret

- Transform $\text{LWE}_{\mathbf{s}}$ (for arbitrary $\mathbf{s}$) into $\text{LWE}_{\mathbf{e}}$ for Gaussian secret $\mathbf{e}$:

  Given the source $\text{LWE}_{\mathbf{s}}$ of samples $(\mathbf{a}_i \ , \ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$,

  1. Draw n samples $(\mathbf{A} \ , \ \mathbf{b} = \mathbf{A}^t \mathbf{s} + \mathbf{e})$ so that $\mathbf{A}$ is invertible mod q.
  2. Draw and transform fresh samples:

$$
\begin{aligned}
(\mathbf{a}, b) \quad &\mapsto \quad (\mathbf{a}' = -\mathbf{A}^{-1}\mathbf{a} \ , \ b + \langle \mathbf{a}', \mathbf{b} \rangle) \\
&= \quad (\mathbf{a}' \ , \ \langle \mathbf{a}, \mathbf{s} \rangle + e - \langle \mathbf{A}^{-1}\mathbf{a}, \mathbf{A}^t \mathbf{s} \rangle + \langle \mathbf{a}', \mathbf{e} \rangle) \\
&= \quad (\mathbf{a}' \ , \ \langle \mathbf{a}', \mathbf{e} \rangle + e).
\end{aligned}
$$

LWE with Gaussian Secret

- Transform $\text{LWE}_{\mathbf{s}}$ (for arbitrary $\mathbf{s}$) into $\text{LWE}_{\mathbf{e}}$ for Gaussian secret $\mathbf{e}$:

  Given the source $\text{LWE}_{\mathbf{s}}$ of samples $(\mathbf{a}_i \ , \ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$,

  1. Draw n samples $(\mathbf{A} \ , \ \mathbf{b} = \mathbf{A}^t \mathbf{s} + \mathbf{e})$ so that $\mathbf{A}$ is invertible mod q.
  2. Draw and transform fresh samples:

$$\begin{aligned}(\mathbf{a}, b) \quad &\mapsto \quad (\mathbf{a}' = -\mathbf{A}^{-1}\mathbf{a} \ , \ b + \langle \mathbf{a}', \mathbf{b} \rangle) \\ &= \quad (\mathbf{a}' \ , \ \langle \mathbf{a}, \mathbf{s} \rangle + e - \langle \mathbf{A}^{-1}\mathbf{a}, \mathbf{A}^t \mathbf{s} \rangle + \langle \mathbf{a}', \mathbf{e} \rangle) \\ &= \quad (\mathbf{a}' \ , \ \langle \mathbf{a}', \mathbf{e} \rangle + e).\end{aligned}$$

  (Also maps uniform samples $(\mathbf{a}, b)$ to uniform $(\mathbf{a}', b')$).

LWE with Gaussian Secret

- Transform $\text{LWE}_{\mathbf{s}}$ (for arbitrary $\mathbf{s}$) into $\text{LWE}_{\mathbf{e}}$ for Gaussian secret $\mathbf{e}$:

  Given the source $\text{LWE}_{\mathbf{s}}$ of samples $(\mathbf{a}_i\ ,\ b_i = \langle \mathbf{a}_i, \mathbf{s} \rangle + e_i)$,

  1. Draw n samples $(\mathbf{A}\ ,\ \mathbf{b} = \mathbf{A}^t\mathbf{s} + \mathbf{e})$ so that $\mathbf{A}$ is invertible mod q.
  2. Draw and transform fresh samples:

$$
\begin{aligned}
(\mathbf{a}, b) \quad &\mapsto \quad (\mathbf{a}' = -\mathbf{A}^{-1}\mathbf{a}\ ,\ b + \langle \mathbf{a}', \mathbf{b} \rangle) \\
&= \quad (\mathbf{a}'\ ,\ \langle \mathbf{a}, \mathbf{s} \rangle + e - \langle \mathbf{A}^{-1}\mathbf{a}, \mathbf{A}^t\mathbf{s} \rangle + \langle \mathbf{a}', \mathbf{e} \rangle) \\
&= \quad (\mathbf{a}'\ ,\ \langle \mathbf{a}', \mathbf{e} \rangle + e).
\end{aligned}
$$

  (Also maps uniform samples $(\mathbf{a}, b)$ to uniform $(\mathbf{a}', b')$).

Clique & Affine Security (Again, For Free)

- Repeating transform produces ind. sources $\text{LWE}_{\mathbf{e}_1}$, $\text{LWE}_{\mathbf{e}_2}$, ...
- Side effect: a *known affine relation* between *unknowns* $\mathbf{s}$ and $\mathbf{e}_i$.

  This lets us create $\text{Enc}_{pk_i}(\text{affine}(\mathbf{e}_j))$ for any i, j.

Final Words

- The *simple, linear* structure of lattice-based encryption allows for many enhancements.
- There is much more to be done!

Final Words

- The *simple, linear* structure of lattice-based encryption allows for many enhancements.
- There is much more to be done!

Thanks!