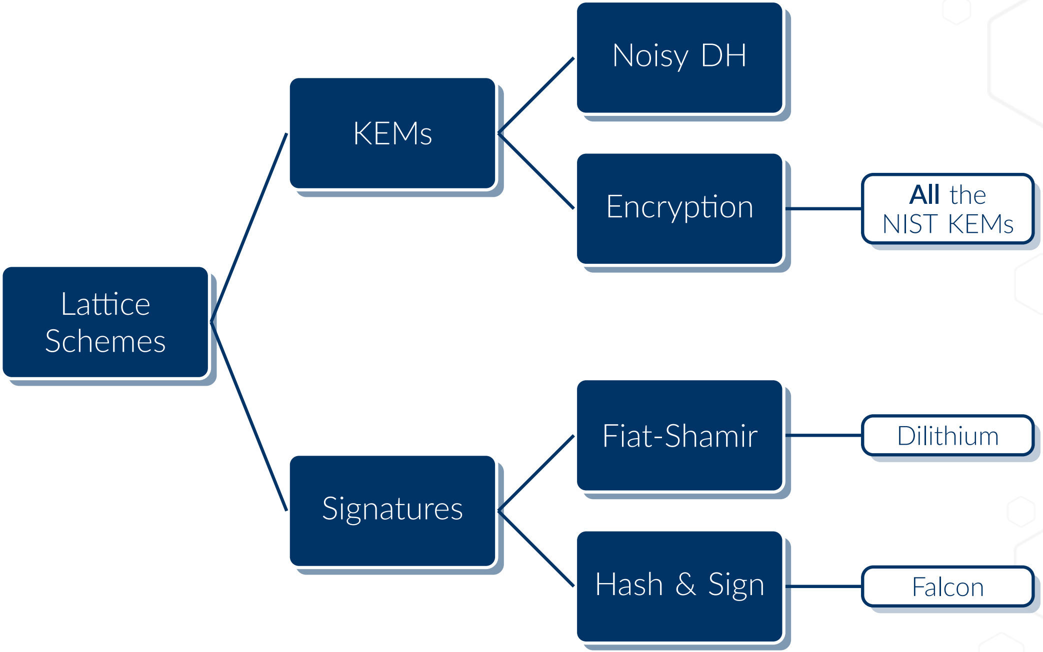
Basic Constructions over Lattices I: Key Establishment

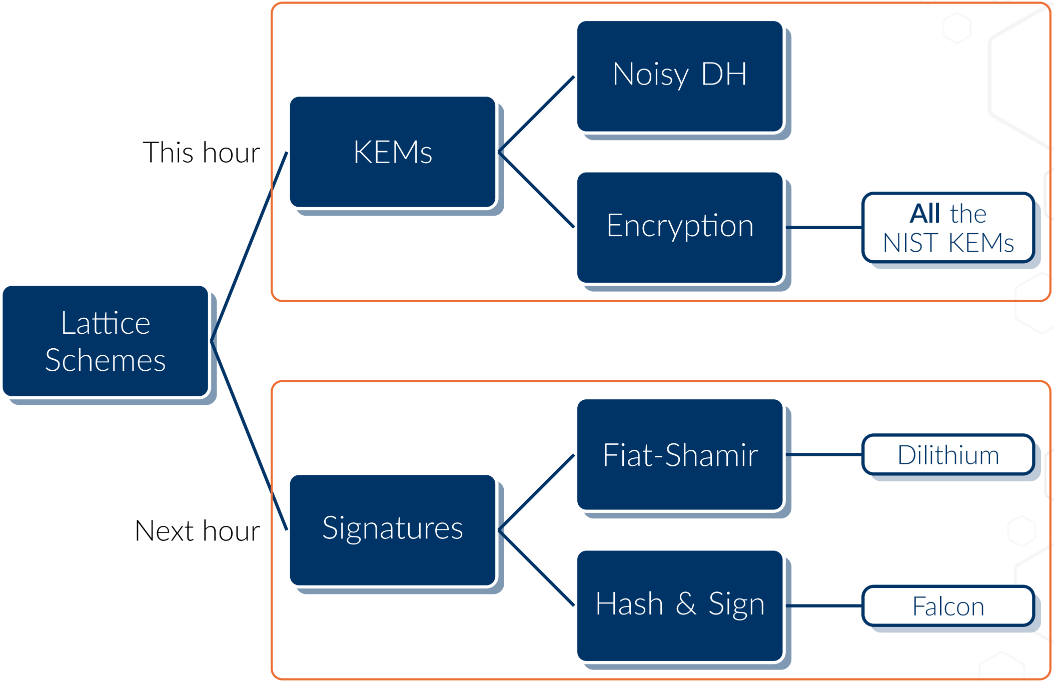
Thomas Prest

PQShield

ASCRYPTO 2021

Introduction


Lattice Schemes
KEMs
Noisy DH
Encryption
All the NIST KEMs
Signatures
Fiat-Shamir
Dilithium
Hash & Sign
Falcon


This hour
Next hour
Lattice Schemes
KEMs
Noisy DH
Encryption
All the NIST KEMs
Signatures
Fiat-Shamir
Dilithium
Hash & Sign
Falcon

Roadmap

Approach of this course:

- Proceed by analogy between number-theoretic vs lattice-based schemes (e.g. El Gamal vs "noisy El Gamal" [LPR10, LP11])
- This way, we can separate paradigm (e.g. Fiat-Shamir) and assumption (e.g. LWE)
  - Allows to understand which notions are intrinsic to the paradigm or are assumption-dependent
  - Straightforward adaptations can fail, and it is important to understand why
  - In general, it is useful to understand where two assumptions (e.g. LWE vs DLOG) may be similar, and where they differ

Plan:

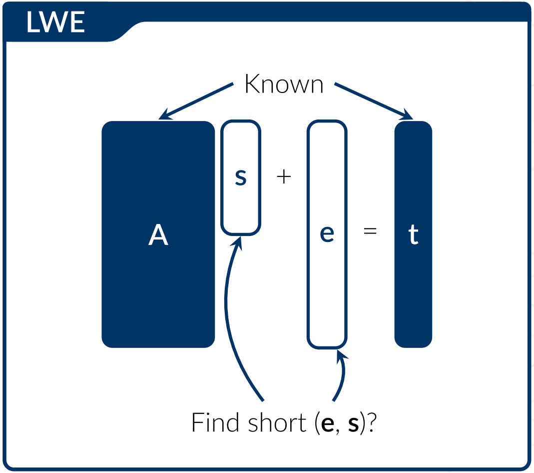
- *This hour:* key-establishment schemes based on LWE
  - There exist encryption based on NTRU as well.
    We ignore them here, but most of our remarks apply.
- *Next hour:* signature schemes based on LWE/SIS/NTRU

Learning with Errors (LWE)

The LWE assumption:

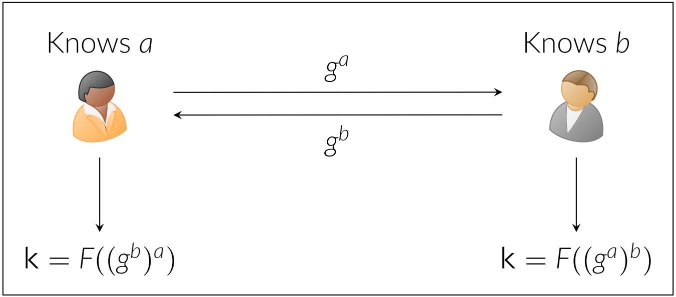
- Two fundamental variants:
  - *Search*: hard to find $(\mathbf{s}, \mathbf{e})$
  - *Decision*: hard to distinguish $(\mathbf{A}, \mathbf{t})$ from uniformly random $(\mathbf{A}, \mathbf{u})$
- Very versatile:
  - Base ring $\mathcal{R}$ $(\mathbb{Z}, \mathbb{Z}[x]/(x^d + 1), \dots)$
  - Dimensions of the vectors/matrices
  - Distributions (binomial, uniform, etc.)
- Resilient to some extent to bit dropping
  - If we drop the least significant bits of $\mathbf{t}$, $(\mathbf{A}, \mathbf{t})$ is still useful.
  - See also LWR, where $\mathbf{t} = \mathrm{MSB}(\mathbf{A} \cdot \mathbf{s})$.
- We will liken a lot LWE to DLOG:
  - *LWE*: $(\mathbf{A}, \mathbf{t} = \mathbf{A}\mathbf{s} + \mathbf{e}) \xrightarrow{\mathcal{A}} (\mathbf{s}, \mathbf{e})$?
  - *DLOG*: $(g, h = g^x) \xrightarrow{\mathcal{A}} x$?

  This analogy is convenient when it holds, and informative when it fails.



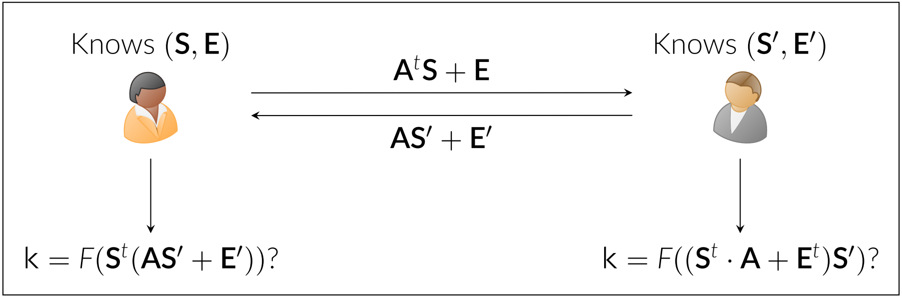
Noisy Diffie-Hellman

Diffie-Hellman



A few comments:

- It works thanks to the commutativity of exponentiation: $(g^a)^b = (g^b)^a = g^{ab}$
- Nice properties: fully non-interactive, can be used with static keys, etc.



- Parties have to "pick a side", but that isn't the big issue.
- Bigger problem: and agree on a shared secret *up to some additive noise*:

$$\mathbf{S}^t(\mathbf{A}\mathbf{S}' + \mathbf{E}') = \mathbf{S}^t\mathbf{A}\mathbf{S}' + \underline{\mathbf{S}^t\mathbf{E}'} \approx \mathbf{S}^t\mathbf{A}\mathbf{S}' + \underline{\mathbf{E}^t\mathbf{S}'} = (\mathbf{S}^t \cdot \mathbf{A} + \mathbf{E}^t)\mathbf{S}' \qquad (1)$$

- Natural idea: keep most significant bits (MSB) of $\mathbf{S}^t\mathbf{A}\mathbf{S}' + \underline{\mathbf{S}^t\mathbf{E}'}$ and $\mathbf{S}^t\mathbf{A}\mathbf{S}' + \underline{\mathbf{E}^t\mathbf{S}'}$.
  - But does it work?

Why keeping the MSBs doesn't work

For this example, assume $q = 2^K$. Suppose that at a given coefficient:

1. $\mathbf{S}^t\mathbf{A}\mathbf{S}'$ is extremely close to $q/2$ or 0
2. $\mathbf{S}^t\mathbf{E}' > 0$
3. $\mathbf{E}^t\mathbf{S}' < 0$

Then  and  might not agree on the MSB of that coefficient.

A few comments:

- →  and  have no way to detect that they disagreed on a coefficient.
- → Discarding coefficients that are "too close to $q/2$" doesn't work.
- → One naive workaround is to set $q = \Omega(2^\lambda)$, but this raises its own problems.

The results of [GKRS20] suggest that if $q = o(2^\lambda)$, then finding a function F such that

$$F(\mathbf{S}^t(\mathbf{A}\mathbf{S}' + \mathbf{E}')) = F((\mathbf{S}^t \cdot \mathbf{A} + \mathbf{E}^t)\mathbf{S}') \tag{2}$$

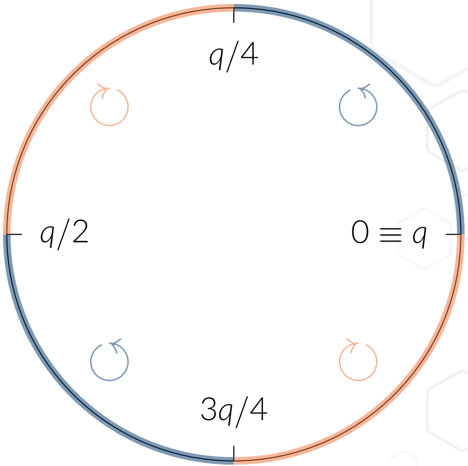
is a difficult problem.

The solution: reconciliation [DXL12, Pei14]

For each coefficient x, sends 1 additional "reconciliation bit" $\texttt{rec}(x)$: ↺ / ↻

- If we identify $\mathbb{Z}_q$ with a circle, ↺ / ↻ indicates in which direction rounded the coefficient.
- This allows and to agree on "*borderline*" coefficients. For example, if for a given x:
  1. has $0.501 \cdot q$
  2. has $0.499 \cdot q$

  Then sends ↺ and rounds to 1 (instead of 0 if there was no reconciliation bit).

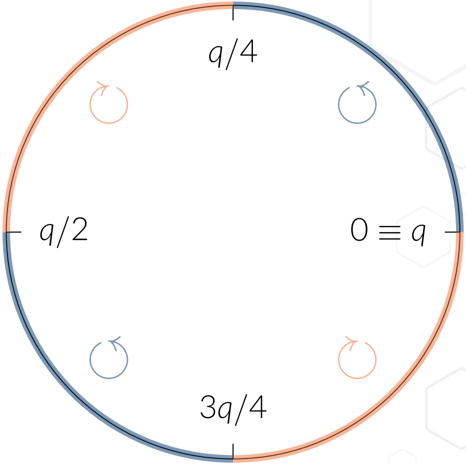


The solution: reconciliation [DXL12, Pei14]

For each coefficient x, sends 1 additional "reconciliation bit" $\mathtt{rec}(x)$: ↺ / ↻

- → If we identify $\mathbb{Z}_q$ with a circle, ↺ / ↻ indicates in which direction rounded the coefficient.
- → This allows and to agree on "*borderline*" coefficients. For example, if for a given x:
  1. has $0.501 \cdot q$
  2. has $0.499 \cdot q$

  Then sends ↺ and rounds to 1 (instead of 0 if there was no reconciliation bit).



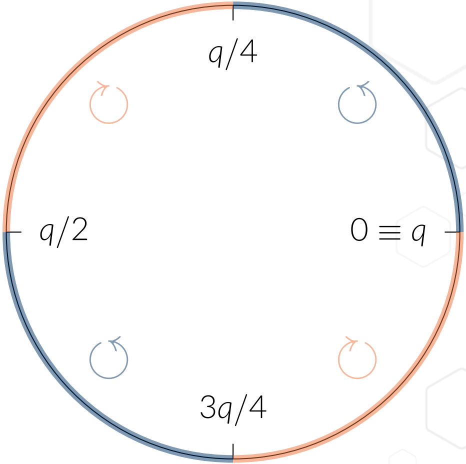
Question: Given $q = 2^\kappa$ and $x \leftarrow \mathbb{Z}_q$, does $\mathtt{rec}(x)$ leak information about MSB(x)?

The solution: reconciliation [DXL12, Pei14]

For each coefficient x, sends 1 additional "reconciliation bit" $\texttt{rec}(x)$: $\circlearrowleft$ / $\circlearrowright$

- If we identify $\mathbb{Z}_q$ with a circle, $\circlearrowleft$ / $\circlearrowright$ indicates in which direction rounded the coefficient.
- This allows and to agree on "*borderline*" coefficients. For example, if for a given x:
  1. has $0.501 \cdot q$
  2. has $0.499 \cdot q$

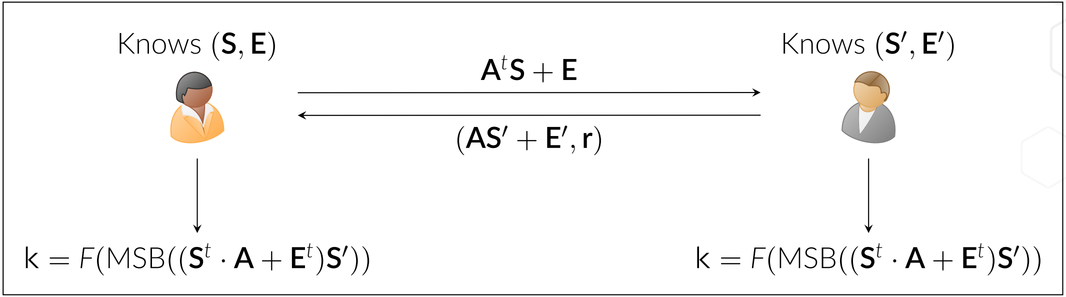
  Then sends $\circlearrowleft$ and rounds to 1 (instead of 0 if there was no reconciliation bit).



Question: Given $q = 2^\kappa$ and $x \leftarrow \mathbb{Z}_q$, does $\texttt{rec}(x)$ leak information about $\mathrm{MSB}(x)$?
Answer: No. Computation below for the case $(\texttt{rec}(x) = \circlearrowleft)$:

$$\mathbb{P}_{x \leftarrow \mathbb{Z}_p}[\mathrm{MSB}(x) = 0 \mid \texttt{rec}(x) = \circlearrowleft] = \frac{\mathbb{P}_{x \leftarrow \mathbb{Z}_p}[(\mathrm{MSB}(x) = 0) \wedge (\texttt{rec}(x) = \circlearrowleft)]}{\mathbb{P}_{x \leftarrow \mathbb{Z}_p}[\texttt{rec}(x) = \circlearrowleft]} = \frac{q/4}{q/2} = \frac{1}{2}$$

Noisy Diffie-Hellman: now with reconciliation



Variations on the reconciliation idea:

- → More coefficients than bits of k: use error-correcting codes [ADPS16]
- → Less coefficients than bits of k: encode several bits per coefficient [BCD+16]
- → Simply discard coefficients that are borderline for  [Saa17]

Noisy Diffie-Hellman: now with reconciliation

Knows $(\mathbf{S}, \mathbf{E})$

$$\xrightarrow{\mathbf{A}^t\mathbf{S} + \mathbf{E}}$$

$$\xleftarrow[(\mathbf{A}\mathbf{S}' + \mathbf{E}', \mathbf{r})]{}$$

Knows $(\mathbf{S}', \mathbf{E}')$

$$\mathrm{k} = F(\mathrm{MSB}((\mathbf{S}^t \cdot \mathbf{A} + \mathbf{E}^t)\mathbf{S}'))$$

$$\mathrm{k} = F(\mathrm{MSB}((\mathbf{S}^t \cdot \mathbf{A} + \mathbf{E}^t)\mathbf{S}'))$$

Some fundamental differences with classical Diffie-Hellman:

- → Reconciliation makes the protocol interactive
- → Cannot be used with static keys due to an attack by Fluhrer [Flu16]

Noisy El Gamal

El Gamal

Keygen$(g \in \mathbb{G})$

1. Sample $x \leftarrow \mathbb{Z}_{|\mathbb{G}|}$
2. $h \leftarrow g^x$
3. $\mathsf{dk} := x, \mathsf{ek} := h$

Enc$(\mathbf{msg}, \mathbf{ek})$

1. Sample $r \leftarrow \mathbb{Z}_{|\mathbb{G}|}$
2. $u \leftarrow g^r$
3. $v \leftarrow h^r \cdot \mathtt{msg}$
4. $\mathsf{c} := (u, v)$

Dec$(\mathbf{c}, \mathbf{dk})$

1. $\mathtt{msg} \leftarrow v \cdot u^{-x}$

"Noisy" El Gamal [LPR10, LP11]

Keygen$(\mathbf{A} \in \mathcal{R}_q^{m \times m})$

1. Sample short $\mathbf{S}, \mathbf{E}$
2. $\mathbf{B} \leftarrow \mathbf{AS} + \mathbf{E}$
3. $\mathsf{dk} := (\mathbf{S}, \mathbf{E}), \mathsf{ek} := \mathbf{B}$

Enc$(\mathbf{msg}, \mathbf{ek})$

1. Sample short $\mathbf{R}, \mathbf{E}', \mathbf{E}''$
2. $\mathbf{U} \leftarrow \mathbf{RA} + \mathbf{E}'$
3. $\mathbf{V} \leftarrow \mathbf{RB} + \mathbf{E}'' + \mathtt{Encode}(\mathtt{msg})$
4. $\mathsf{c} := (\mathbf{U}, \mathbf{V})$

Dec$(\mathbf{c}, \mathbf{dk})$

1. $\mathtt{msg} \leftarrow \mathtt{Decode}(\mathbf{V} - \mathbf{US})$

Correctness

Decryption is successful since:

- **El Gamal:** $v \cdot u^{-x} = (h^r \cdot \mathtt{msg}) \cdot (g^r)^{-x} = \mathtt{msg}$
- **Noisy El Gamal:**

$$\begin{aligned} \mathbf{V} - \mathbf{U}\mathbf{S} &= \big(\mathbf{R}(\mathbf{A}\mathbf{S} + \mathbf{E}) + \mathbf{E}'' + \mathtt{Encode}(\mathtt{msg})\big) - (\mathbf{R}\mathbf{A} + \mathbf{E}')\mathbf{S} && (3) \\ &= \mathtt{Encode}(\mathtt{msg}) + (\mathbf{R}\mathbf{E} + \mathbf{E}'' - \mathbf{E}'\mathbf{S}) && (4) \end{aligned}$$

The recipient recovers $\mathtt{msg}$ as long as $(\mathbf{R}\mathbf{E} + \mathbf{E}'' - \mathbf{E}'\mathbf{S})$ remains small.

Exercise (solutions in the next slide)

Let us note/specify:

- a. For a matrix $\mathbf{X}$, $\|\mathbf{X}\|_{\max}$ is the max in absolute norm of its entries (or the coefficients of their entries if these are polynomials).
- b. Any unspecified matrix dimension of $\mathbf{R}, \mathbf{S}, \mathbf{E}, \mathbf{E}', \mathbf{E}''$ is n.
- c. $r = \|\mathbf{R}\|_{\max}, s = \|\mathbf{S}\|_{\max}, e = \|\mathbf{E}\|_{\max}, e' = \|\mathbf{E}'\|_{\max}, e'' = \|\mathbf{E}''\|_{\max}$

Exercise

Give a formula using $m, n, q, r, s, e, e', e''$ so that perfect correctness is guaranteed:

1. Assuming the base ring $\mathcal{R}$ is $\mathbb{Z}$, the modulus q is 2^k, $\mathtt{msg} = \{0,1\}^{n \times n}$, $\mathtt{Encode}(\mathtt{msg}) := q/2 \cdot \mathtt{msg}$ and $\mathtt{Decode}(\mathbf{M})$ rounds each coefficient of $\mathbf{M}$ to the closest value in $\{0, q/2\} \in \mathbb{Z}_q$.
2. Same conditions but the base ring $\mathcal{R}$ is $\mathbb{Z}[x]/(x^d+1)$ and $\mathtt{msg} = \{\{0,1\}^d\}^{n \times n}$.

Remember: $\mathtt{Dec}(\mathtt{msg}, \mathtt{dk}) = \mathtt{Decode}(\mathtt{Encode}(\mathtt{msg}) + (\mathbf{RE} + \mathbf{E}'' - \mathbf{E}'\mathbf{S}))$.

Solutions for the previous slide

Note that due to condition b, we have $\mathbf{S}, \mathbf{E} \in \mathcal{R}_q^{m \times n}$, $\mathbf{R}, \mathbf{E}' \in \mathcal{R}_q^{n \times m}$ and $\mathbf{E}'' \in \mathcal{R}_q^{n \times n}$.

Answer to 1

Perfect correctness is guaranteed if this value is $\leq q/4$:

$$\begin{aligned} \|\mathbf{RE} + \mathbf{E}'' - \mathbf{E}'\mathbf{S}\|_{\max} &\leq \|\mathbf{RE}\|_{\max} + \|\mathbf{E}''\|_{\max} + \|\mathbf{E}'\mathbf{S}\|_{\max} \\ &\leq mre + e'' + me's \end{aligned}$$

Answer to 2

Note that for two polynomials $f, g \in \mathbb{Z}[x]/(x^d + 1)$, we have $\|fg\|_\infty \leq d\|f\|_\infty \|g\|_\infty$. This gives the following condition on correctness:

$$d(mre + e'' + me's) \leq q/4$$

Note: these conditions are tight.

Security definitions

IND-CPA Experiment

1. $(\mathtt{ek}, \mathtt{dk}) \leftarrow \mathtt{Keygen}()$
2. $b \leftarrow \{0, 1\}$
3. $(\mathtt{msg}_0, \mathtt{msg}_1, \mathtt{st}) \leftarrow \mathcal{A}(\mathtt{ek})$
4. $\mathtt{c} \leftarrow \mathtt{Enc}(\mathtt{msg}_b, \mathtt{ek})$
5. $b' \leftarrow \mathcal{A}(\mathtt{ek}, \mathtt{c}, \mathtt{st})$
6. If $(b = b')$ return 1, else return 0.

IND-CCA Experiment

1. $(\mathtt{ek}, \mathtt{dk}) \leftarrow \mathtt{Keygen}()$
2. $b \leftarrow \{0, 1\}$
3. $(\mathtt{msg}_0, \mathtt{msg}_1, \mathtt{st}) \leftarrow \mathcal{A}(\mathtt{ek})$
4. $\mathtt{c} \leftarrow \mathtt{Enc}(\mathtt{msg}_b, \mathtt{ek})$
5. $b' \leftarrow \mathcal{A}^{Oracle_{\mathtt{Decaps}}(\cdot)}(\mathtt{ek}, \mathtt{c}, \mathtt{st})$
6. If $(b = b')$ return 1, else return 0.

Note: $Oracle_{\mathtt{Decaps}}(\cdot)$ is a decryption oracle for any ciphertext $\mathtt{c}' \neq \mathtt{c}$.

The advantage of an adversary $\mathcal{A}$ in either experiment is:

$$\left| \mathbb{P}[\text{The game outputs } 1] - \frac{1}{2} \right|.$$

IND-CPA security

Let us note $\texttt{Encode}(\texttt{msg}) = \mathbf{M}$. These three views are indistinguishable:

Real view: $(\mathbf{A} \text{ unif}, \mathbf{B} = \mathbf{AS} + \mathbf{E}, \mathbf{U} = \mathbf{RA} + \mathbf{E}', \mathbf{V} = \mathbf{RB} + \mathbf{E}'' + \mathbf{M})$

Hybrid 1: $(\mathbf{A} \text{ unif}, \mathbf{B} \text{ unif}, \mathbf{U} = \mathbf{RA} + \mathbf{E}', \mathbf{V} = \mathbf{RB} + \mathbf{E}'' + \mathbf{M})$
$\Leftrightarrow ([\mathbf{A}\|\mathbf{B}] \text{ unif}, [\mathbf{U}\|\mathbf{V}] = \mathbf{R}[\mathbf{A}\|\mathbf{B}] + [\mathbf{E}'\|\mathbf{E}''] + [\mathbf{0}\|\mathbf{M}])$

Hybrid 2: $([\mathbf{A}\|\mathbf{B}] \text{ unif}, [\mathbf{U}\|\mathbf{V}] \text{ unif})$

Proof outline:

- *(Real-world $\approx_c$ Hybrid 1)* under LWE
- *(Hybrid 1 $\approx_c$ Hybrid 2)* under LWE

From Blueprints to Concrete Schemes

IND-CCA security: the problem

The "Noisy El Gamal" scheme is only IND-CPA secure. Example of a CCA attack:

- Remember that $\mathbf{E}^* = (\mathbf{RE} + \mathbf{E}'' - \mathbf{E}'\mathbf{S})$ must be small in order to have $\mathtt{Decode}(\mathtt{Encode}(\mathtt{msg}) + \mathbf{E}^*) = \mathtt{msg}$.
- $\mathcal{A}$ can set $\mathbf{R} = \mathbf{0}$, $\mathbf{E}' = \mathbf{I}$,[1] and $\mathbf{E}''$ arbitrarily when computing $\mathtt{c} = \mathtt{Enc}(\mathtt{msg}, \mathtt{ek})$.
- By checking on which coefficients $\mathtt{Dec}(\mathtt{c}, \mathtt{dk})$ and $\mathtt{msg}$ differ, the attacker can learn which coefficients of $(\mathbf{E}'' - \mathbf{S})$ are "too large" and gradually recover $\mathbf{S}$.

See also *key-mismatch attacks* [DFR18, BGRR19], with an even weaker attack model.

[1] For simplicity, we assume that $\mathbf{E}'$ is square, which is e.g. the case in Ring-LWE schemes.

IND-CCA security: the solution

A generic solution: CPA-to-CCA transforms.

- *Generically* transform an IND-CPA scheme into an IND-CCA scheme.
- The resulting IND-CCA scheme is not necessarily a PKE, can also be e.g. a KEM.
- Fujisaki-Okamoto transforms [FO99a, FO99b] and their variants are the most common ones. High-level idea:
  - During encryption, generate the encryption randomness (here $\mathbf{R}, \mathbf{E}', \mathbf{E}''$) by passing $\mathtt{msg}$ (and $\mathtt{ek}$) into a PRF: $(\mathbf{R}, \mathbf{E}', \mathbf{E}'') := F(\mathtt{msg}, \mathtt{ek})$.
  - During decryption, recompute $(\mathbf{R}, \mathbf{E}', \mathbf{E}'')$ and therefore $\mathtt{c}$ from $\mathtt{msg}$ (and $\mathtt{ek}$). If they don't match, abort or output a pseudo-random shared key (in case of a KEM).

Decryption failures

Some lattice-based schemes suffer from decryption failures: a small portion of $(\mathbf{R}, \mathbf{E}', \mathbf{E}'')$ may lead to an incorrect decryption (error or different message) for dk.

- **Why do they happen?** Perfect correctness may require larger parameters.
- **Can this be an issue?** Yes:
  - Once a decryption failures is found, it is easier to find others. [DRV20]
  - Decryption failures can be exploited to mount key-recovery attacks [DGJ+19, DVV19, GJY19].
- **Note:** They can also impact code-based and rank-based schemes.
- Also work against IND-CCA: brute-force `msg` until decryption failures are found.

Decryption failures

Some lattice-based schemes suffer from decryption failures: a small portion of $(\mathbf{R}, \mathbf{E}', \mathbf{E}'')$ may lead to an incorrect decryption (error or different message) for $\mathtt{dk}$.

- **Why do they happen?** Perfect correctness may require larger parameters.
- **Can this be an issue?** Yes:
  - Once a decryption failures is found, it is easier to find others. [DRV20]
  - Decryption failures can be exploited to mount key-recovery attacks [DGJ+19, DVV19, GJY19].
- **Note:** They can also impact code-based and rank-based schemes.
- Also work against IND-CCA: brute-force $\mathtt{msg}$ until decryption failures are found.

The solution: set the probability p of decryption failures to be $\leq 2^{-\kappa}$.

- $p = 0$: NTRU [CDH+20], NTRU Prime [BBC+20]
- $p \leq 2^{-\kappa}$: Kyber [SAB+20], Saber [DKR+20], FrodoKEM [NAB+20]
- One can reduce p by embedding an error-correcting code in $\mathtt{msg}$, but it is risky:
  - Side-channel attacks [DTVV19]
  - Theoretical attacks [DVV19, GJY19]

Questions?

Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe.
Post-quantum key exchange - A new hope.
In Thorsten Holz and Stefan Savage, editors, *USENIX Security 2016*, pages 327–343. USENIX Association, August 2016.

Daniel J. Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang.
NTRU Prime.
Technical report, National Institute of Standards and Technology, 2020.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila.
Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE.
In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1006–1018. ACM Press, October 2016.

Aurélie Bauer, Henri Gilbert, Guénaël Renault, and Mélissa Rossi.
Assessment of the key-reuse resilience of NewHope.
In Mitsuru Matsui, editor, *CT-RSA 2019*, volume 11405 of *LNCS*, pages 272–292. Springer, Heidelberg, March 2019.

Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, Tsunekazu Saito, Takashi Yamakawa, and Keita Xagawa.
NTRU.
Technical report, National Institute of Standards and Technology, 2020.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

Jintai Ding, Scott R. Fluhrer, and Saraswathy RV.
Complete attack on RLWE key exchange with reused keys, without signal leakage.
In Willy Susilo and Guomin Yang, editors, *ACISP 18*, volume 10946 of *LNCS*, pages 467–486. Springer, Heidelberg, July 2018.

Jan-Pieter D'Anvers, Qian Guo, Thomas Johansson, Alexander Nilsson, Frederik Vercauteren, and Ingrid Verbauwhede.
Decryption failure attacks on IND-CCA secure lattice-based schemes.
In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part II*, volume 11443 of *LNCS*, pages 565–598. Springer, Heidelberg, April 2019.

Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso.
SABER.
Technical report, National Institute of Standards and Technology, 2020.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

Jan-Pieter D'Anvers, Mélissa Rossi, and Fernando Virdia.
(One) failure is not an option: Bootstrapping the search for failures in lattice-based encryption schemes.
In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part III*, volume 12107 of *LNCS*, pages 3–33. Springer, Heidelberg, May 2020.

Jan-Pieter D'Anvers, Marcel Tiepelt, Frederik Vercauteren, and Ingrid Verbauwhede.
Timing attacks on error correcting codes in post-quantum schemes.
In Begül Bilgin, Svetla Petkova-Nikova, and Vincent Rijmen, editors, *TIS@CCS*, pages 2–9. ACM, 2019.

Jan-Pieter D'Anvers, Frederik Vercauteren, and Ingrid Verbauwhede.
The impact of error dependencies on ring/mod-LWE/LWR based schemes.
In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 10th International Conference, PQCrypto 2019*, pages 103–115. Springer, Heidelberg, 2019.

Jintai Ding, Xiang Xie, and Xiaodong Lin.
A simple provably secure key exchange scheme based on the learning with errors problem.
Cryptology ePrint Archive, Report 2012/688, 2012.
https://eprint.iacr.org/2012/688.

Scott Fluhrer.
Cryptanalysis of ring-LWE based key exchange with key share reuse.
Cryptology ePrint Archive, Report 2016/085, 2016.
https://eprint.iacr.org/2016/085.

Eiichiro Fujisaki and Tatsuaki Okamoto.
How to enhance the security of public-key encryption at minimum cost.
In Hideki Imai and Yuliang Zheng, editors, *PKC'99*, volume 1560 of *LNCS*, pages 53–68. Springer, Heidelberg, March 1999.

Eiichiro Fujisaki and Tatsuaki Okamoto.
Secure integration of asymmetric and symmetric encryption schemes.
In Michael J. Wiener, editor, *CRYPTO'99*, volume 1666 of *LNCS*, pages 537–554. Springer, Heidelberg, August 1999.

Qian Guo, Thomas Johansson, and Jing Yang.
A novel CCA attack using decryption errors against LAC.
In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 82–111. Springer, Heidelberg, December 2019.

Siyao Guo, Pritish Kamath, Alon Rosen, and Katerina Sotiraki.
Limits on the efficiency of (ring) LWE based non-interactive key exchange.
In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 374–395. Springer, Heidelberg, May 2020.

Richard Lindner and Chris Peikert.
Better key sizes (and attacks) for LWE-based encryption.
In Aggelos Kiayias, editor, *CT-RSA 2011*, volume 6558 of *LNCS*, pages 319–339. Springer, Heidelberg, February 2011.

Vadim Lyubashevsky, Chris Peikert, and Oded Regev.
On ideal lattices and learning with errors over rings.
In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 1–23. Springer, Heidelberg, May / June 2010.

Michael Naehrig, Erdem Alkim, Joppe Bos, Léo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila.
FrodoKEM.
Technical report, National Institute of Standards and Technology, 2020.
available at `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

Chris Peikert.
Lattice cryptography for the internet.
In Michele Mosca, editor, *Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014*, pages 197–219. Springer, Heidelberg, October 2014.

Markku-Juhani O. Saarinen.
HILA5: On reliability, reconciliation, and error correction for ring-LWE encryption.
In Carlisle Adams and Jan Camenisch, editors, *SAC 2017*, volume 10719 of *LNCS*, pages 192–212. Springer, Heidelberg, August 2017.

Peter Schwabe, Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Gregor Seiler, and Damien Stehlé.

CRYSTALS-KYBER.
Technical report, National Institute of Standards and Technology, 2020.
available at https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.