CCA-Secure (Puncturable) KEMs from Encryption With Non-Negligible Decryption Errors

Valerio Cini, Sebastian Ramacher, Daniel Slamanig, Christoph Striecks

ASIACRYPT 2020, December 7

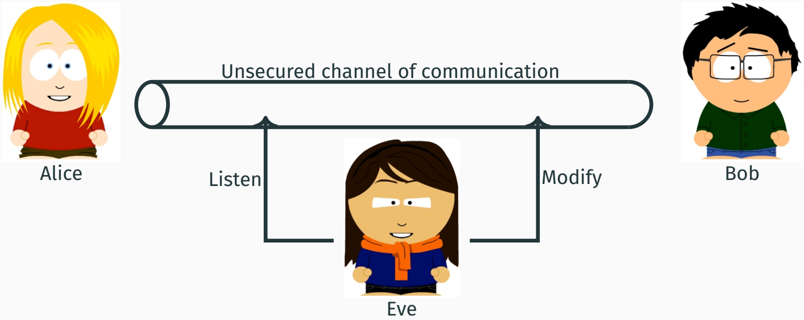
AIT Austrian Institute of Technology

Talk Overview

- Main Definitions and Models
- Decryption Error
- Our Compiler
- Evaluation
- Further Results & Conclusions

Main Definition and Models

Confidentiality = Indistinguishability



Motivation

Public key encryption allows two parties to communicate securely even when no prior secret shared key is available to them.

It is extremely useful for establishing secure communications over the Internet: e.g., the TLS protocol.

Perfect Correctness

Definition

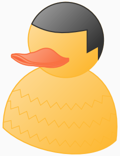
We say that a public-key encryption (PKE) scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is perfectly correct if the following holds:

- for every message $M \in \mathcal{M}$, for every pair $(\mathsf{pk}, \mathsf{sk})$ generated by KeyGen on input λ, and all possible coin tosses of Enc and Dec, it should hold that $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M)) = M$.

Analogous definition for key encapsulation mechanisms (KEMs).

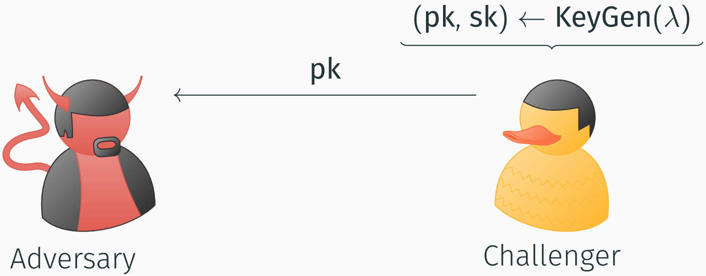
Security Games: IND-CPA

Adversary



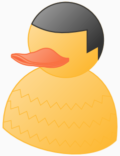
Challenger

Security Games: IND-CPA



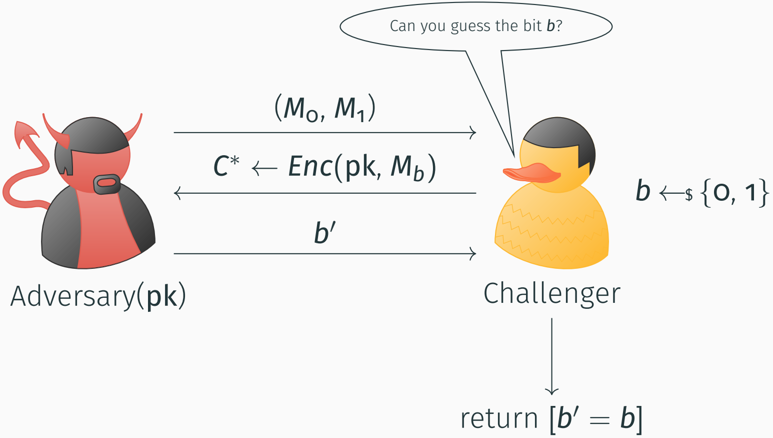
Security Games: IND-CPA

Adversary($\mathbf{pk}$)

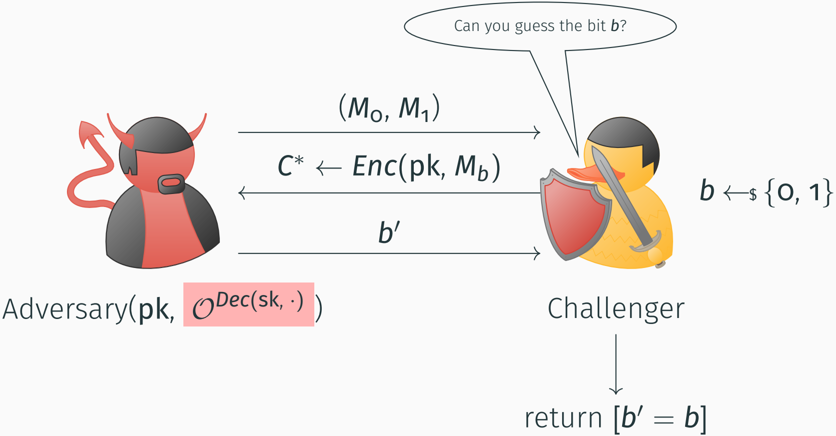


Challenger

Security Games: IND-CPA



Security Games: IND-CCA



Adversary Advantage

Definition

The adversary advantage in game $\mathbf{x} \in \{\text{IND-CPA}, \text{IND-CCA}\}$, is:

$$Adv^{\mathsf{x}}(\mathcal{A}) = \left| \Pr[\mathcal{A} \text{ wins in the } \mathbf{x} \text{ game}] - \frac{1}{2} \right| .$$

Why CCA Security?

CCA de-facto security standard nowadays; particular important in practice:

Why CCA Security?

CCA de-facto security standard nowadays; particular important in practice:

- Daniel Bleichenbacher, "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1" [Ble98],

and many follow up works.

Why CCA Security?

CCA de-facto security standard nowadays; particular important in practice:

- Daniel Bleichenbacher, "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1" [Ble98],

and many follow up works.

Such an attack enables an adversary to completely recover the original message for any ciphertext of its choice!

Decryption Error

Imperfect Decryption

Some PKE schemes (especially those post-quantum) do not have perfect decryption.

Imperfect Decryption

Some PKE schemes (especially those post-quantum) do not have perfect decryption.

When the ability to correctly decrypt valid ciphertexts is dependent on the secret key, the result of the decryption process can leak information about the secret key (e.g., [BS20] and [DRV20]).

Trade-offs

Decryption error can be naively decreased by increasing the parameters of the PKE.

- security (e.g., different ways of sampling error in lattice-based PKEs),
- efficiency,
  - size (e.g., public-key, ciphertext),
  - runtime,
- decryption error.

Imperfect Decryption: First Definition [DNR04]

Definition

A PKE Π is DNR-$\delta(\cdot)$-correct if we have that

$$\Pr[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, M)) \neq M] \leq \delta(\lambda),$$

where the probability is taken over the choice of keypairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$, $M \in \mathcal{M}$, and over the random coins of Enc and Dec.

Imperfect Decryption: Final Definition [HHK17]

Definition

A PKE Π is $\delta(\cdot)$-correct if

$$\mathbb{E}\left[\max_{M \in \mathcal{M}} \Pr[C \leftarrow \mathsf{Enc}(\mathsf{pk}, M) : \mathsf{Dec}(\mathsf{sk}, C) \neq M]\right] \leq \delta(\lambda),$$

where the expected value is taken over $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(\lambda)$.

Taking the maximum over all possible messages we obtain an upper-bound for the “decryption error” of any single message.

Our Compiler

Main Idea

- start from a IND-CPA secure PKE with non-negligible correctness error (e.g., 128 bit security level, error $> 2^{-128}$):
  - easier to construct,
  - concrete security of parameters set can be studied in depth,

Main Idea

- start from a IND-CPA secure PKE with non-negligible correctness error (e.g., 128 bit security level, error $> 2^{-128}$):
  - easier to construct,
  - concrete security of parameters set can be studied in depth,
- reduce the error to a negligible value (i.e., $< 2^{-128}$),

Main Idea

- start from a IND-CPA secure PKE with non-negligible correctness error (e.g., 128 bit security level, error $> 2^{-128}$):
  - easier to construct,
  - concrete security of parameters set can be studied in depth,
- reduce the error to a negligible value (i.e., $< 2^{-128}$),

- transform it into an IND-CCA secure PKE/KEM, preserving the negligible correctness error.

Direct Product Compiler [DNR04]

- parallel repetition of encryption of the same message under different randomness reduces decryption error exponentially,

$$\mathsf{Enc}'(\mathsf{pk}, M) := (\mathsf{Enc}(\mathsf{pk}, M; r_1), \ldots, \mathsf{Enc}(\mathsf{pk}, M; r_\ell)),$$

Direct Product Compiler [DNR04]

- parallel repetition of encryption of the same message under different randomness reduces decryption error exponentially,

$$\mathsf{Enc}'(\mathsf{pk}, M) := (\mathsf{Enc}(\mathsf{pk}, M; r_1), \ldots, \mathsf{Enc}(\mathsf{pk}, M; r_\ell)),$$

- majority vote is needed to decide which message obtain is the correct one,

Direct Product Compiler [DNR04]

- parallel repetition of encryption of the same message under different randomness reduces decryption error exponentially,

$$\mathsf{Enc}'(\mathsf{pk}, M) := (\mathsf{Enc}(\mathsf{pk}, M; r_1), \ldots, \mathsf{Enc}(\mathsf{pk}, M; r_\ell)),$$

- majority vote is needed to decide which message obtain is the correct one,
- even if the underlying PKE is IND-CCA secure, the so obtained PKE’ is not.

Correctness Error Direct Product Compiler

δ	$\delta'(2)$	$\delta'(3)$	$\delta'(4)$
2^{-32}	$\approx 2^{-32}$	$\approx 2^{-63}$	$\approx 2^{-94}$
2^{-64}	$\approx 2^{-64}$	$\approx 2^{-127}$	$\approx 2^{-190}$
2^{-96}	$\approx 2^{-96}$	$\approx 2^{-191}$	$\approx 2^{-284}$

Table 1: Estimation of the correctness error for the direct product compilers. $\delta'(\ell)$ denotes the correctness error for ℓ ciphertexts.

FO Transform [FO99]

It is a generic transformation that converts “any” CPA-secure PKE into a CCA-secure KEM.

FO Transform [FO99]

It is a generic transformation that converts "any" CPA-secure PKE into a CCA-secure KEM.

rPKE IND-CPA —— FO transform ——→ KEM IND-CCA

FO Transform [FO99]

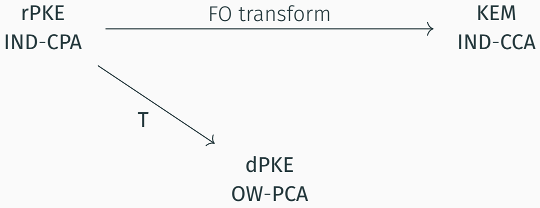
It is a generic transformation that converts “any” CPA-secure PKE into a CCA-secure KEM.

rPKE IND-CPA —— FO transform ——→ KEM IND-CCA

The transformation is modular [HHK17]: it can be viewed as the composition of two different ones.

FO Transform [FO99]

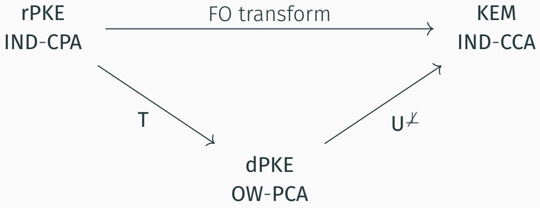
It is a generic transformation that converts “any” CPA-secure PKE into a CCA-secure KEM.



The transformation is modular [HHK17]: it can be viewed as the composition of two different ones.

FO Transform [FO99]

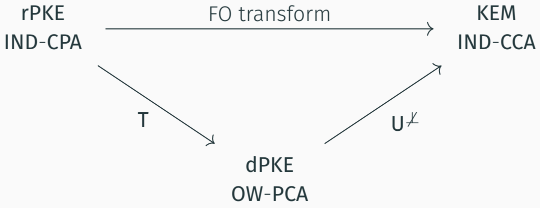
It is a generic transformation that converts "any" CPA-secure PKE into a CCA-secure KEM.



The transformation is modular [HHK17]: it can be viewed as the composition of two different ones.

FO Transform [FO99]

It is a generic transformation that converts “any” CPA-secure PKE into a CCA-secure KEM.



The transformation is modular [HHK17]: it can be viewed as the composition of two different ones.

The FO transform requires negligible correctness error of the underlying PKE.

The Transformation $\mathsf{T}^\star$ (1/2)

We compute ℓ independent encryptions of the same message M under the same public key pk using randomness $\mathsf{G}(M, i)$, $i \in [\ell]$, where G is a RO (random oracle).

The Transformation $\mathsf{T}^\star$ (1/2)

We compute ℓ independent encryptions of the same message M under the same public key pk using randomness $\mathsf{G}(M, i)$, $i \in [\ell]$, where G is a RO (random oracle).

The resulting de-randomized PKE Π' has then correctness error $\delta' := \delta^\ell$, where ℓ is chosen in a way that δ^ℓ is negligible.

The Transformation $\mathsf{T}^\star$ (1/2)

We compute ℓ independent encryptions of the same message M under the same public key pk using randomness $\mathsf{G}(M, i)$, $i \in [\ell]$, where G is a RO (random oracle).

The resulting de-randomized PKE Π' has then correctness error $\delta' := \delta^\ell$, where ℓ is chosen in a way that δ^ℓ is negligible.

During the decryption correctness of the message can be checked via the de-randomization: it allows us to control if the ciphertext was modified and to get rid of the majority vote.

The Transformation $\mathsf{T}^\star$ (2/2)

$\Pi'.\mathsf{Enc}(\mathsf{pk}, M)$	$\Pi'.\mathsf{Dec}(\mathsf{sk}, C)$
for $i = 1, \ldots, \ell$ **do**	$\mathtt{res} \leftarrow \bot$, $\mathtt{check} \leftarrow \bot$
$\quad C_i := \Pi.\mathsf{Enc}(\mathsf{pk}, M; \mathsf{G}(M, i))$	**for** $i = 1, \ldots, \ell$ **do**
$C := (C_1, \ldots, C_\ell)$	$\quad \mathtt{res}[i] := \Pi.\mathsf{Dec}(\mathsf{sk}, C_i)$
return C	**for** $i \in [\ell]$ s.t. $\mathtt{res}[i] \neq \bot$ **do**
	$\quad$ **if** $\forall j \in [\ell] : C_j = \Pi.\mathsf{Enc}(\mathsf{pk}, \mathtt{res}[i], \mathsf{G}(\mathtt{res}[i], j))$
	$\quad\quad \mathtt{check} \leftarrow i$
	if $\mathtt{check} \neq \bot$
	$\quad$ **return** $\mathtt{res}[\mathtt{check}]$
	return $\bot$

To the resulting PKE Π' we can then directly apply the transformation $\mathsf{U}^{\not\perp}$ from the modular analysis of the FO transform [HHK17], to obtain an IND-CCA secure KEM with negligible correctness error in the (Q)ROM.

Runtime and bandwidth overheads

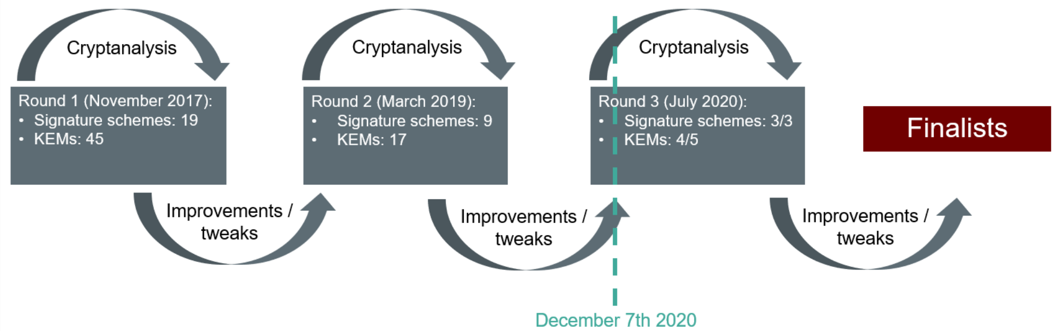
	$\|pk\|$	$\|C\|$	KeyGen	Enc	Dec
$C_{p,y}$	1 (r) / ℓ (d)	ℓ	1 (r) / ℓ (d)	ℓ	ℓ
$C^{\star}_{p,d}$	ℓ'	ℓ'	ℓ'	ℓ'	ℓ'
$T^{\star}$	1	ℓ'	1	ℓ'	ℓ'^2 / ℓ' ($\perp$)

Table 2: Comparison of the runtime and bandwidth overheads of $C_{p,y}$, $y \in \{r, d\}$, with ℓ ciphertexts and $T^{\star}$ and $C^{\star}_{p,d}$ with ℓ' ciphertexts such that $\ell \geq \ell' + 1$.

Evaluation

NIST Post-Quantum Competition

- Important competitions for cryptographic schemes in the past: AES, SHA-1, SHA-3;
- Now running a Post-Quantum Cryptography Standardization project: Signatures and PKE/KEMs.



Question

How does increasing from 1 to ℓ ciphertexts compare to increasing the parameters at comparable resulting decryption errors for (existing) round-2 submissions in the NIST PQC?

N.B. Our work took place before Round-3 started.

Code-based PKEs/KEMs

Code-based Round 2 Submissions

Encryption/KEMs	assumption	problem
Classic McEliece	codes	Goppa
NTS-KEM	codes	Goppa
BIKE	codes	short Hamming
HQC	codes	short Hamming
LEDAcrypt	codes	short Hamming
ROLLO	codes	low rank
RQC	codes	low rank

Code-based Round 2 Submissions

Encryption / KEMs	assumption	problem
Classic McEliece	codes	Goppa
NTS-KEM	codes	Goppa
BIKE[1]	codes	short Hamming
HQC	codes	short Hamming
LEDAcrypt	codes	short Hamming
ROLLO	codes	low rank
RQC	codes	low rank

[1] BIKE is a Round-3 Alternate Candidate.

Evaluation (1/2)

KEM	δ	pk	C	$\sum$	KeyGen	Encaps	Decaps
O[ROLLO-I-L1,5]	2^{-150}	465	2325	2790	0.10	0.02 /0.10	0.26 /1.30
ROLLO-II-L1	2^{-128}	1546	1674	3220	0.69	0.08	0.53
O[ROLLO-I-L3,4]	2^{-128}	590	2360	2950	0.13	0.02 / 0.08	0.42 /1.68
ROLLO-II-L3	2^{-128}	2020	2148	4168	0.83	0.09	0.69
O[ROLLO-I-L5,4]	2^{-168}	947	7576	8523	0.20	0.03 /0.12	0.78 /3.12
ROLLO-II-L5	2^{-128}	2493	2621	5114	0.79	0.10	0.84
O[BIKE-2-L1,3]	2^{-147}	10163	30489	40652	4.79	0.14 /0.42	3.29 /9.88
BIKE-2-CCA-L1	2^{-128}	11779	12035	23814	6.32	0.20	4.12

Table 3: Sizes (in bytes) and runtimes (in ms and millions of cycles for BIKE), where O denotes the transformed scheme. Runtimes are taken from the optimized implementations, if available, and are only intra-scheme comparable.

Lattice-based PKEs/KEMs

Round 2 Submissions (2/2)

Encryption/KEMs	assumption	problem
Crystals-Kyber	lattice	MLWE
Saber	lattice	MLWR
FrodoKEM	lattice	LWE
Round 5	lattice	LWR
LAC	lattice	RLWE
NewHope	lattice	RLWE
Three Bears	lattice	IMLWE
NTRU	lattice	NTRU
NTRUprime	lattice	NTRU

Round 2 Submissions (2/2)

Encryption/KEMs	assumption	problem
Crystals-Kyber	lattice	MLWE
Saber	lattice	MLWR
FrodoKEM[2]	lattice	LWE
Round 5	lattice	LWR
LAC	lattice	RLWE
NewHope	lattice	RLWE
Three Bears	lattice	IMLWE
NTRU	lattice	NTRU
NTRUprime	lattice	NTRU

[2] FrodoKEM is a Round-3 Alternate Candidate.

Evaluation (2/2)

KEM	δ	pk	C	$\sum$	KeyGen	Encaps	Decaps
O[R5N1-3-PKE-cpa,2]	2^{-130}	8834	17732	26566	6.69	10.10/20.20	10.38/20.75
R5N1-3-KEM-cca	2^{-144}	9660	9732	19392	6.78	10.20	10.60
O[FrodoCCS-Rec.,4]	2^{-155}	11280	45152	56432	2.94	3.48/13.94	10.79/43.16
FrodoKEM-640-AES	2^{-138}	9616	9720	19336	1.38	1.86	1.75

Table 4: Sizes (in bytes) and runtimes, where O denotes the transformed scheme. FrodoCCS refers to the FrodoKEM version precedent to the NIST competition. Runtimes are taken from the optimized implementations if available.

Further Results & Conclusions

Further Results

Bloom Filter KEMs:

- recent primitive proposed by Derler et al. [Der+18],
- building block to construct fully forward-secret 0-RTT key exchange protocols [Gün+17],
- required perfect decryption of underlying building block (hinders post-quantum instantiations).

Further Results

Bloom Filter KEMs:

- recent primitive proposed by Derler et al. [Der+18],
- building block to construct fully forward-secret 0-RTT key exchange protocols [Gün+17],
- required perfect decryption of underlying building block (hinders post-quantum instantiations).

We extended the work generically and showed that one can construct BFKEMs from any IBE and even base it upon IBEs with a (non-)negligible correctness error:

- first post-quantum CCA-secure BFKEM.

Conclusions

- generic way to deal with the error from weaker schemes (i.e., IND-CPA secure ones with non-negligible error) which are easier to design,
- all involved algorithms are easily parallelizable,
- our approach performs well in context of code-based schemes but gives less advantage for lattice-based ones.
- first post-quantum CCA-secure Bloom Filter KEM

Open Questions

- extending analysis to other constructions?
- code- VS lattice-based schemes: why the compiler performs so differently?

Thank you for your attention!

(full version of the Asiacrypt'20 paper to appear soon on ePrint)

Supported by:

References

[Ble98] D. Bleichenbacher. "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS# 1". In: *Annual International Cryptology Conference*. Springer. 1998, pp. 1–12.

[BS20] N. Bindel and J. M. Schanck. "Decryption failure is more likely after success". In: *International Conference on Post-Quantum Cryptography*. Springer. 2020, pp. 206–225.

[Der+18] D. Derler et al. "Bloom Filter Encryption and Applications to Efficient Forward-Secret 0-RTT Key Exchange". In: 2018, pp. 425–455. DOI: 10.1007/978-3-319-78372-7_14.

[DNR04] C. Dwork, M. Naor, and O. Reingold. "Immunizing encryption schemes from decryption errors". In: *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2004, pp. 342–360.

[DRV20] J.-P. D'Anvers, M. Rossi, and F. Virdia. "(One) Failure Is Not an Option: Bootstrapping the Search for Failures in Lattice-Based Encryption Schemes". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2020, pp. 3–33.

[FO99] E. Fujisaki and T. Okamoto. “Secure integration of asymmetric and symmetric encryption schemes”. In: *Annual International Cryptology Conference*. Springer. 1999, pp. 537–554.

[Gün+17] F. Günther et al. “0-RTT Key Exchange with Full Forward Secrecy”. In: 2017, pp. 519–548. DOI: 10.1007/978-3-319-56617-7_18.

[HHK17] D. Hofheinz, K. Hövelmanns, and E. Kiltz. “A modular analysis of the Fujisaki-Okamoto transformation”. In: *Theory of Cryptography Conference*. Springer. 2017, pp. 341–371.