FO derandomization sometimes damages security

Daniel J. Bernstein

University of Illinois at Chicago, Department of Computer Science, USA
Academia Sinica, Institute of Information Science, Taiwan

Abstract. FO derandomization is a common step in protecting against chosen-ciphertext attacks. There are theorems qualitatively stating that FO derandomization preserves ROM OW-CPA security. However, quantitatively, these theorems are loose, allowing the possibility of the derandomized security level being considerably smaller than the original security level. Many cryptosystems rely on FO derandomization without adjusting parameters to account for this looseness.

This paper proves, for two examples of a randomized ROM PKE, that derandomizing the PKE degrades ROM OW-CPA security by a factor close to the number of hash queries. The first example can be explained by the size of the message space of the PKE; the second cannot. This paper also gives a concrete example of a randomized non-ROM PKE that appears to have the same properties regarding known attacks.

As a spinoff, this paper presents a 2^{88}-guess attack exploiting derandomization to break one out of 2^{40} ciphertexts for a FrodoKEM-640 public key. This attack contradicts the official FrodoKEM claim that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin". The official responses to this attack so far include (1) renaming FrodoKEM as "ephemeral FrodoKEM" and (2) proposing a newly patched "FrodoKEM".

This paper does not involve new cryptanalysis: the attacks are straightforward. What is new is finding examples where derandomization damages security.

Keywords: public-key encryption · Fujisaki–Okamoto transformation · T transformation

1 Introduction

Fujisaki–Okamoto [42] proposed modularizing the task of designing a hopefully-IND-CCA2 PKE into two tasks:

- Design a hopefully-one-way PKE. This is a simpler task: one does not have to worry about distinguishers or about chosen-ciphertext attacks.
- Apply a generic transform, now called the "FO transform", to obtain a hopefully-IND-CCA2 PKE.

The usual argument for safety of the resulting PKE is as follows: (1) we believe, based on cryptanalysis, that the original PKE is in fact one-way ("OW-CPA"); (2) there is an FO *theorem* saying that if the original PKE is OW-CPA then the transformed PKE is ROM

This work was funded by the Deutsche Forschungsgemeinschaft under Germany's Excellence Strategy—EXC 2092 CASA—390781972 "Cyber Security in the Age of Large-Scale Adversaries"; by the U.S. National Science Foundation under grant 2037867; by the Cisco University Research Program; and by the Taiwan's Executive Yuan Data Safety and Talent Cultivation Project (AS-KPQ-109-DSTCP). "Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation" (or other funding agencies). Permanent ID of this document: a6c406911ef1c325541ff3cce0667d57aaffc91d. Date: 2024.12.30.

E-mail: djb@cr.yp.to (Daniel J. Bernstein)

IND-CCA2; (3) we believe that there are no IND-CCA2 attacks more effective than ROM IND-CCA2 attacks.

However, even if the first and third steps in this argument are correct, a closer look shows that the FO theorem in the second step is not tight. The ROM IND-CCA2 advantage could be polynomially higher than the success probability of OW-CPA attacks against the original PKE.

Subsequent work reviewed in Section 1.3 has produced tight ROM theorems for some FO variants. A tight theorem typically follows one of three paths: (1) assume a *deterministic* OW-CPA PKE; (2) assume an *IND-CPA* PKE; (3) require a much less efficient iterated transform. But consider the following setting: a transform having similar efficiency to the original FO transform is applied to a *randomized* PKE, and one wants to deduce security from an OW-CPA assumption rather than making a stronger IND-CPA assumption. This setting appears frequently, and the theorems available for this setting are unsatisfactory.

1.1 Examples of randomized PKEs

It is useful to consider concrete examples of randomized PKEs to understand (1) the extent to which FO is used and (2) why assuming IND-CPA is unsatisfactory.

Consider the DH NIKE from [39], written here in additive notation as commonly used for elliptic curves: Alice publishes aG, where a is secret and G is public; Bob publishes bG, where b is secret; Alice and Bob now share a secret abG. Relabeling Bob's "public key" bG as "ciphertext" converts this NIKE into a KEM. Simple additive encryption of a message M converts this KEM into a PKE, the ElGamal PKE from [40], with ciphertext $(bG, M + abG)$. Decryption in this PKE recovers only M, not the randomness b that was used in encryption.

Similarly, in post-quantum cryptography, a typical construction of a hopefully-IND-CCA2 lattice-based PKE (see, e.g., [6]) starts from a randomized "noisy DH" PKE, with aG and bG replaced by $aG + e$ and $bG + d$ (or $aG + e$ and $Gb + d$ in the non-commutative case; see, e.g., [8] and [5]). The construction then applies a reasonably efficient FO variant to convert this hopefully-OW-CPA PKE into a hopefully-IND-CCA2 PKE.

For original DH, common practice skips ElGamal's PKE, skips FO, takes the KEM described above, and modifies the KEM by simply applying a hash function as proposed by Shoup in [74], so the KEM ciphertext is bG and the KEM session key is $H(abG)$. But—even if one believes that this DH KEM is secure—it is not so easy to skip FO for typical lattice-based "noisy DH" PKEs. Alice computes $a(bG + d) = abG + ad$, while Bob computes $b(aG + e) = abG + be$, which is different. Alice applies an error-correction process to suppress the difference between ad and be, extracting the correct M in the end, but the same error-correction process allows easy chosen-ciphertext attacks that add small modifications to $bG + d$; see, e.g., [44] and [79]. It is thus unsurprising that FO is used pervasively in post-quantum cryptography.

Assuming IND-CPA for the underlying PKEs is more risky than assuming merely OW-CPA. The central issue is that distinguishing problems such as IND-CPA offer more attack avenues than search problems such as OW-CPA. As Goldreich wrote in [43]: "What concerns us about the DDH assumption is the fact that this assumption refers to a setting that is less simple than usual (e.g., DDH is less simple than DH), which makes this assumption harder to evaluate." See [13, Sections 6.2–6.3] for examples illustrating that mathematical algorithm designers focus primarily on search problems. The occasional studies of the extra risks of distinguishing problems have produced some easy breaks (e.g., DDH is broken when there are "cofactors", as noted in, e.g., [27, Section 1.1]; similarly, decisional LWE for the polynomial $x^n - 1$ is broken by the factor $x - 1$ of $x^n - 1$) and some more subtle breaks (e.g., some elliptic curves have efficient pairings, so, as pointed out in [62], DDH is broken for those curves), which is worrisome. So it is important to ask, and the literature asks, whether an OW-CPA assumption suffices for a tight proof.

1.2 Is there a guarantee of ROM IND-CCA2 security for an efficient transform of a randomized OW-CPA PKE?

The best ROM theorems available for this setting (see, e.g., [45, Section 3.3]) say that the IND-CCA2 advantage against the transformed system is bounded by about $2q$ times the OW-CPA success probability against the original system, when there are at most q calls to the oracle used for derandomization and at most q calls to other oracles.

In short, OW-CPA for the PKE implies ROM IND-CCA2 for the KEM *except* for a looseness gap, a factor $2q$. The following paragraphs consider four ways that this gap is potentially unimportant.

Is the gap quantitatively too small to worry about? Any cryptosystem with enough security margin to survive years of technology development will be able to survive having attack costs reduced by a factor 2. However, a factor q is much more problematic. An attacker devoting just 65 megawatts to hashing using 7nm technology would carry out 2^{85} hashes per year; Bitcoin carried out 2^{93} hashes in 2023. See [35] for an example of a 65-megawatt spy center; [17, Appendix J] for the performance of readily available 7nm hashing equipment; and [26] for the Bitcoin numbers.

Are cryptosystem proposals applying looseness adjustments? One response to looseness gaps is to adjust cryptosystem parameters to compensate, avoiding a disconnect between proofs and proposed parameters. Bellare–Rogaway wrote in [11, Section 1.2]: "We reiterate the crucial point: if the reduction proving security is 'loose,' like the one above, the efficiency of the scheme is impacted, because we must move to a larger security parameter."

However, many, perhaps most, of the cryptosystems with loose proofs proposed after [11] do not apply looseness adjustments. One of the examples listed in [55, Section 5.4] is MQDSS [37], a "provably secure" round-2 submission to the NIST Post-Quantum Cryptography Standardization Project ("NISTPQC") that was broken by [52]. MQDSS would have avoided the break from [52] if it had—at whatever cost in efficiency—adjusted parameters to account for the looseness of its proofs.

Random sampling of other NISTPQC submissions quickly finds many more examples of submissions without looseness adjustments, including Kyber [8], which was selected for standardization by NIST, and FrodoKEM [5], which according to [28] is under consideration for standardization by ISO.

Are cryptosystem proposals using deterministic PKEs, avoiding the risk? Another response to the lack of tight proofs for derandomization is to avoid this setting: start from a deterministic PKE so as to be able to use tight proofs, as recommended in [18, Appendix F]. However, as noted in [56], "only 2.5 out of 17 second round NIST proposals for public-key encryption schemes" took this approach: Classic McEliece [4], NTRU [36], and the Streamlined NTRU Prime option within NTRU Prime [20].

Various applications are now using the latter KEMs: Streamlined NTRU Prime has been used by OpenSSH by default since April 2022 [1]; NTRU-HRSS has been used for Google's internal communication since November 2022 [48]; Classic McEliece is deployed in various applications listed in [2]; also, [33] says that ISO already has a Classic McEliece draft. For those applications, this is the end of the story. But this leaves open the security question for KEMs built from randomized PKEs, such as Kyber and FrodoKEM.

Are there better proofs? Another response is to search for a tight theorem, eliminating the q factor in the current theorems. But perhaps the q factor cannot be eliminated. One scenario to consider is that the CCA transform maintains security without this being

provable; many truths are unprovable. Another scenario to consider, an example of what Menezes [58] calls the "nightmare scenario" for loose proofs, is that CCA transforms make attacks q times easier.

1.3 Overview of FO improvements

Shoup [74, Section 4.2] (see also [75, Section 3]) proposed constructing a hopefully-IND-CCA2 PKE by constructing a hopefully-IND-CCA2 KEM and constructing a hopefully-secure DEM. KEMs are simpler than PKEs, and hopefully-secure DEMs are readily available from symmetric cryptography.

Dent [38] proposed constructing a hopefully-IND-CCA2 KEM by combining a hopefully-OW-CPA PKE with a CCA transform, analogous to the FO transform but in the simpler KEM context. Modern KEM constructions typically follow this structure, although the details of the CCA transform vary.

One of Dent's theorems [38, Theorem 8] obtains a ROM IND-CCA2 KEM *tightly* from any deterministic OW-CPA PKE, using what is now known as "plaintext confirmation". Persichetti [66, Section 5.3] introduced, for a particular deterministic PKE, a different strategy for tight ROM IND-CCA2 KEM proofs from OW-CPA, using what is now known as "implicit rejection". Hofheinz–Hövelmanns–Kiltz [45] generalized the implicit-rejection theorem to handle any deterministic OW-CPA PKE.

Hofheinz–Hövelmanns–Kiltz also observed that a wide range of FO variants for randomized PKEs factor into two simpler pieces, and presented state-of-the-art proofs factored analogously. The first piece is always the following transform T, called derandomization. The transform is given a PKE C and a public hash function H. The transform outputs a deterministic PKE $T(C, H)$, which is the same as C except that it uses $H(M)$ as the coins used to encrypt M. For example, if C is the ElGamal cryptosystem with a random choice of the DH scalar b, then $T(C, H)$ instead chooses $b = H(M)$.

Hofheinz–Hövelmanns–Kiltz proved tight ROM OW-CPA security of $T(C, H)$ assuming IND-CPA security of C, and proved loose ROM OW-CPA security of $T(C, H)$ assuming OW-CPA security of C. The loss factor is approximately q. An ℓ-fold iteration in [45, Section 3.4] obtains ROM IND-CPA security from OW-CPA security with loss factor only $q^{1/\ell}$ but makes ciphertexts ℓ times longer.

Further work on this topic includes allowing decryption failures in the PKE (see [45]), analyzing QROM IND-CCA2 security rather than just ROM IND-CCA2 security (see [45], [71], [49], [24], and [56]), additional factorizations of the transforms and proofs (see [71] and [22]), and various efforts to formally verify proofs (see, e.g., [77]).

All ROM IND-CCA2 theorems available today that start by merely assuming an OW-CPA PKE have loss factor at least $q^{1/\ell}$ with ℓ-fold ciphertext expansion; this is, *a fortiori*, also the case for QROM IND-CCA2 theorems. In particular, if the application is unwilling to incur a doubling of ciphertext size, the loss factor is at least q. Are better proofs possible? Or could it be that derandomization, the T transform, really does degrade OW-CPA security by a factor q?

1.4 Contributions of this paper

This paper reports the discovery of PKE examples where derandomization degrades OW-CPA security against standard attack strategies by a factor close to q. The PKE examples are reasonably simple, and the analyses are conceptually straightforward—no new cryptanalysis. The core novelty in this paper is *finding the examples*.

Three examples are presented. Example 2 is a ROM PKE with the feature of a *proof* that derandomization degrades OW-CPA security; this is a proof regarding *all* attacks, not just known attacks. Example 1 is a warmup for Example 2. For Example 3, there is

no proof that the known attacks are optimal, but this example has the feature of being a concrete non-ROM PKE.

Section 2 proves, for both Example 1 and Example 2, that derandomization degrades OW-CPA security by a factor close to q. The derandomized OW-CPA success chance for Example 1 is $(q+1)/\#\text{Plaintexts}$, which one might argue does not qualify as "looseness" since cryptosystem designers can be told to choose #Plaintexts to be very large. However, for Example 2, the OW-CPA success chance is far above $(q+1)/\#\text{Plaintexts}$.

Section 3 constructs Example 3, a non-ROM PKE for which derandomization appears, based on an analysis of known attacks, to degrade pre-quantum OW-CPA security by a factor close to q for every reasonable choice of hash function, where now q is the number of attack operations. As in Example 2, the success probabilities here are far above $(q+1)/\#\text{Plaintexts}$.

As a spinoff, Section 4 breaks a security claim for a proposed high-profile KEM, specifically the claim in [5, page 44] that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin". The break exploits internal derandomization in exactly the same way as the attack against Example 1. Quantitatively, 2^{128} efficient guesses break a ciphertext sent to a FrodoKEM-640 public key, and 2^{88} efficient guesses break one of 2^{40} ciphertexts sent to a FrodoKEM-640 public key.

The question of whether the original randomized FrodoPKE is close to q times more secure than the derandomized system is a complicated question about lattice attacks; analyzing lattice attacks is outside the scope of this paper. Sections 2 and 3 analyze attacks exploiting derandomization, and analyze attacks against randomized PKEs; Section 4 analyzes only attacks exploiting derandomization.

1.5 Priority dates

The first version of this paper was posted in July 2021, including Example 1, Example 2, and Example 3. The break of the FrodoKEM security claim quoted above was announced in October 2022.

1.6 Broader consequences and open questions

It is tempting to imagine that one can selectively disregard limitations in proofs, treating a proof as evidence for something stronger than what the proof actually says. There is, for example, a proof that derandomization loosely preserves OW-CPA security; surely this is evidence that derandomization tightly preserves OW-CPA security. There is also a tight OW-CPA proof for derandomization under a stronger IND-CPA assumption; surely this is further evidence for the same hypothesis. But the hypothesis is incorrect: it is disproven by Example 1. More subtly, even if one assumes large #Plaintexts and designates $(q+1)/\#\text{Plaintexts}$ as tight, the hypothesis is disproven by Example 2.

Formally, one way to avoid this contradiction would be to retreat to the weaker hypothesis that derandomization tightly preserves OW-CPA security for *non-ROM* PKEs. But this hypothesis cannot have a relativizable proof,[1] given this paper's ROM examples. The evidence provided *for* this hypothesis, namely extrapolation from weaker proofs, is relativizable, and relativizing the extrapolation produces a false statement, so the

[1] Retreating to statements about non-ROM PKEs also raises questions regarding whether these statements logically compose. Consider, e.g., the "noisy DH" lattice-based PKE mentioned above, presumably the target of cryptanalysis. This PKE chooses G randomly, but a real proposal typically chooses G as hash output from a short public string. This is not a problem for ROM analyses: one steps through (1) a ROM PKE that chooses G as random-oracle output, (2) a derandomized ROM PKE, (3) a ROM KEM including a CCA conversion, and, if desired, (4) a ROM PKE including this KEM and a DEM. If one of these steps is only for non-ROM cryptosystems, then logically one has to figure out whether this step can be at the beginning of the chain. If two steps are only for non-ROM cryptosystems then it is not at all clear what to do.

evidence is weak. Meanwhile this hypothesis implies a much better OW-CPA attack against Example 3, begging the question of what this attack is. This hypothesis would also say that randomized FrodoPKE-640 is as weak as derandomized FrodoPKE-640; this is possible but again raises the question of justification.

A time-tested approach to managing cryptographic risks requires all proof gaps to be filled with detailed cryptanalysis—or with better proofs. Could an OW-CPA assumption plus a small extra assumption produce a tight proof? Tight proofs are already known assuming IND-CPA, but IND-CPA assumptions are more risky than OW-CPA assumptions. Perhaps some sort of intermediate assumption can be identified that (1) suffices for a tight IND-CCA2 proof, (2) eliminates all of this paper's examples, and at the same time (3) follows from OW-CPA for some proposed cryptosystems, or at least is a simpler cryptanalytic target than IND-CPA. This paper's examples could help guide the search for such a proof, the same way that existing proofs helped guide the search for this paper's examples.

Regarding cryptanalysis, there are many randomized hopefully-OW-CPA PKE proposals in the literature. The obvious cryptanalytic challenge, in the absence of a tight proof for derandomization, is to understand the impact that derandomization has upon the security of each proposal. Often randomized PKEs are packaged with specific CCA transforms, but the analysis is important in any case. The CCA transforms are generic "plug and play" components, proposed for use with any PKE meeting specified rules; if this would degrade the security of a particular PKE then there needs to be a warning regarding this degradation. As an illustration of how one cannot assume stability in choices of CCA transforms, NIST's 2023 draft "ML-KEM" standard for Kyber [60] switched to a CCA transform that did not match any version of the Kyber submission to NISTPQC.

1.7 Terminology

The reader is assumed to be familiar with the standard abbreviations "DH" for "Diffie–Hellman", "DDH" for "decisional Diffie–Hellman", "KEM" for "key-encapsulation mechanism", "NIKE" for "non-interactive-key-exchange scheme", "OW-CPA" for "one-wayness against chosen-plaintext attacks", "IND-CPA" for "indistinguishability against chosen-plaintext attacks", "IND-CCA2" for "indistinguishability against chosen-ciphertext attacks" (with the "2" emphasizing that chosen-ciphertext queries are allowed at any moment), "PKE" for "public-key-encryption scheme", and "ROM" for "random-oracle model".

As pointed out in [22], the OW-CPA terminology is misleading: having "chosen-plaintext attacks" in the name "suggests, incorrectly, that the attacker is permitted to choose plaintexts". Saying that an attack can internally choose and encrypt its own plaintexts does not justify the CPA terminology: those plaintexts are not secrets, and internal attack details are invisible in the OW-CPA definition. It is important for this paper's proofs that the secret plaintext encrypted in the OW-CPA definition is *not* influenced by the attacker. The IND-CPA definition is different: it flips a coin between two attacker-chosen plaintexts.

Some examples of IND-CPA being broken, without OW-CPA being broken, exploit the extra risks of distinguishers (IND) compared to search (OW); this is captured in the standard terminology. Other examples exploit the attacker's ability to choose plaintexts; this is not reflected in the standard terminology. To capture this difference, [22] renames "OW-CPA" as "OW-Passive".

An argument against this renaming says that security reviewers are overloaded, and that this is a serious problem for ongoing efforts to evaluate the security of post-quantum cryptosystems. (As context, the study [15] says that 48% of the 69 round-1 NISTPQC submissions are broken by now, 25% of the submissions unbroken during round 1 are broken by now, and 36% of the submissions selected by NIST for round 2 in 2019 are broken by now.) Changing terminology adds to this load, at least in the short term,

perhaps outweighing the advantages of more descriptive terminology. This paper says "OW-CPA".

1.8 Acknowledgments

The author is indebted to Michel Abdalla, Kai-Min Chung, Nils Fleischhacker, Kathrin Hövelmanns, Andreas Hülsing, Eike Kiltz, Tanja Lange, Christian Majenz, Giulio Malavolta, and Christian Schaffner for various discussions that shed light on the topic of this paper. Part of this work was carried out during a visit to the Simons Institute for the Theory of Computing, and part of this work was carried out during a visit to Academia Sinica.

2 Derandomizing a generic information-leaking PKE

This section

- defines $\mathrm{GenericPKE}_{a,b,c,h}$, a randomized ROM PKE;
- specifies parameter choices for Example 1 and Example 2;
- shows that the q-query ROM OW-CPA insecurity of $\mathrm{GenericPKE}_{a,b,c,h}$ is exactly $1/2^b + q/2^{b+h}$, assuming $q \leq 2^b - 1$; and
- shows that the q-query ROM OW-CPA insecurity of the derandomized PKE $\mathrm{TGenericPKE}_{a,b,c,h}$ is at least $(q+1)/2^b$, again assuming $q \leq 2^b - 1$.

Derandomization thus degrades ROM OW-CPA insecurity of this ROM PKE by a factor at least $(q+1)2^h/(q+2^h)$. This factor is very close to $q+1$, under the reasonable assumption that 2^h is much larger than q. The message space Plaintexts for $\mathrm{GenericPKE}_{a,b,c,h}$ has size 2^{a+b}.

2.1 Overview of the PKE construction

If decryption time is irrelevant and unconstrained, the question "What is a simple example of a randomized ROM PKE?" is straightforwardly answered by the special case $a = 0$ of this section: $\mathrm{GenericPKE}_{0,b,c,h}$ feeds a b-bit plaintext and h bits of randomness to a random oracle, producing a c-bit ciphertext.

The further question "Why does OW-CPA not imply IND-CPA?" is standard. One of the standard answers is a PKE transformation that adds information to plaintexts, say an a-bit string ℓ, and adds the same information to ciphertexts. This has no effect on OW-CPA, but it breaks IND-CPA as soon as $a > 0$; if OW-CPA is achievable then this separates IND-CPA from OW-CPA. Applying this transformation to $\mathrm{GenericPKE}_{0,b,c,h}$ produces $\mathrm{GenericPKE}_{a,b,c,h}$.

Building PKEs from random oracles is not a new idea. See, e.g., the more complicated ROM PKEs used by Shoup in [76] to provide "strong evidence that the OAEP construction is not sound": those PKEs are secure, but applying OAEP to those PKEs destroys all security.[2] However, the consequences of such PKEs for derandomization do not appear to have been observed before.

[2] Shoup also gave a counting argument that the insecurity of OAEP on average over all oracles implies the existence of specific oracles relative to which OAEP is insecure.

2.2 Parameter requirements and examples

This paper restricts attention to b chosen so that 2^b is larger than the number of ROM queries q allowed for the attacker. The comparisons of attack probabilities further assume that h is chosen so that 2^h is much larger than q; for concreteness, the reader can take $h = 2b$. Finally, the PKE construction requires $c \geq b + h$.

Example 1, this paper's first example, is $\text{GenericPKE}_{0,b,c,h}$, specializing this section to $a = 0$. For this example, #Plaintexts $= 2^b$, so the original OW-CPA attack probability is only about 1/#Plaintexts, and the derandomized OW-CPA attack probability is only $(q + 1)$/#Plaintexts; as noted in Section 1, this does not contradict typical notions of tightness.

Example 2 is $\text{GenericPKE}_{b,b,c,h}$, instead specializing this section to $a = b$. For this example, #Plaintexts $= 2^{2b}$, so the original OW-CPA attack probability is approximately the square root of 1/#Plaintexts, and the derandomized OW-CPA attack probability is approximately q times larger than that.

2.3 The PKE

This subsection defines $\text{GenericPKE}_{a,b,c,h}$. This is a ROM PKE, using an oracle for a uniform random injective function F from $\{0,1\}^{b+h}$ to $\{0,1\}^c$. ROM success probabilities are by definition averaged over all choices of the oracle, along with all coin flips in algorithms.

Readers who prefer to work solely with uniform random functions, without injectivity constraints, can restrict attention to c much larger than $2(b + h)$, take F as a uniform random function, and observe that F is overwhelmingly likely to be injective. However, this would complicate the theorem statements to account for the tiny correctness error in the resulting PKE and the tiny chance of collisions spoiling the attack.

Definition 2.3.1. *Let a, b, c, h be nonnegative integers with $c \geq b+h$. Let F be a uniform random injective function from $\{0,1\}^{b+h}$ to $\{0,1\}^c$. Then* $\text{GenericPKE}_{a,b,c,h}(F)$ *is defined as*

$$(\text{PublicKeys}, \text{PrivateKeys}, \text{Plaintexts}, \text{Ciphertexts}, \text{KeyGen}, \text{Encrypt}, \text{Decrypt})$$

with the following components:

- PublicKeys $= \{0,1\}^0 = \{()\}$.
- PrivateKeys $= \{0,1\}^0 = \{()\}$.
- Plaintexts $= \{0,1\}^a \times \{0,1\}^b$.
- Ciphertexts $= \{0,1\}^a \times \{0,1\}^c$.
- KeyGen *is the following algorithm:*
  - *Input the empty string* $()$.
  - *Output* $((), ())$.
- Encrypt *is the following algorithm:*
  - *Input* $((\ell, m), p) \in (\{0,1\}^a \times \{0,1\}^b) \times \{()\}$.
  - *Generate a uniform random* $r \in \{0,1\}^h$.
  - *Output* $(\ell, F(m, r)) \in \{0,1\}^a \times \{0,1\}^c$.
- Decrypt *is the following algorithm:*

- *Input* $((\ell, z), s) \in (\{0,1\}^a \times \{0,1\}^c) \times \{()\}$.
- *Search all* $(m, r) \in \{0,1\}^b \times \{0,1\}^h$ *in lexicographic order.*
- *Output* (ℓ, m) *for the first* (m, r) *such that* $F(m, r) = z$.
- *If no such* (m, r) *exists, output* $\perp$.

The decryption algorithm is very slow, but the PKE definition in (e.g.) [45, Section 2.1] places no requirements upon the speed of decryption. More to the point, decryption is irrelevant to the OW-CPA security definition. This choice of decryption algorithm also removes the need for any randomness in private keys. There is also no need for any randomness in public keys, since there is enough randomness in F.

This paper's discovery of examples of derandomization damaging OW-CPA security began with systematic simplification. The simplifications described in the previous paragraph are part of this, and are used in constructing and proving Example 1 and Example 2. For examples of concrete PKEs with fast decryption where the same attack strategy works, see Sections 3 and 4. The advantage of this section's construction of a ROM PKE is that, as noted in Section 1.4, FO derandomization is *proven* to degrade OW-CPA security of this PKE by a factor close to q against *all* attacks.

Theorem 2.3.2. *Under the assumptions of Definition 2.3.1,* $\text{GenericPKE}_{a,b,c,h}(F)$ *is a correct PKE.*

Some of the FO proofs cited in Section 1.3 require correct PKEs. Some can handle decryption failures but become vacuous when decryption failures are frequent. Proving correctness avoids concerns about these issues.

Proof. Syntactic requirements: PublicKeys, PrivateKeys, Plaintexts, Ciphertexts are nonempty finite sets; $\perp \notin$ Plaintexts; KeyGen is an algorithm mapping $\{()\}$ to PublicKeys $\times$ PrivateKeys; Encrypt is an algorithm mapping Plaintexts $\times$ PublicKeys to Ciphertexts; Decrypt is an algorithm mapping Ciphertexts $\times$ PrivateKeys to Plaintexts $\cup$ $\{\perp\}$.

Correctness: Say KeyGen() outputs (p, s); $M \in$ Plaintexts; and Encrypt(M, p) outputs C. By definition of Plaintexts, $M = (\ell, m)$ for some $\ell \in \{0,1\}^a$ and $m \in \{0,1\}^b$. By definition of Encrypt, there is some $r \in \{0,1\}^h$ such that $C = (\ell, z)$ with $z = F(m, r)$. By assumption F is injective, so this (m, r) is the unique preimage of z under F. The search in Decrypt finds this preimage and outputs M as desired. □

2.4 Attacking the PKE

This subsection defines an OW-CPA attack against $\text{GenericPKE}_{a,b,c,h}$, and shows that the attack has success probability exactly $1/2^b + q/2^{b+h}$.

Definition 2.4.1. *Under the assumptions of Definition 2.3.1, let* q *be an element of* $\{0, 1, \ldots, 2^b - 1\}$, *and define* $\text{GenericAttack}_{a,b,c,h,q}(F)$ *as the following algorithm:*

- *Input* $(p, (\ell, z)) \in \{()\} \times (\{0,1\}^a \times \{0,1\}^c)$.
- *Generate a uniform random sequence of distinct elements* $m_0, m_1, \ldots, m_q$ *of* $\{0,1\}^b$.
- *Generate a uniform random sequence of elements* $r_1, \ldots, r_q$ *of* $\{0,1\}^h$.
- *For each* $i \in \{1, 2, \ldots, q\}$ *in increasing order: If* $F(m_i, r_i) = z$, *output* (ℓ, m_i) *and stop.*
- *Output* (ℓ, m_0).

Theorem 2.4.2. *Under the assumptions of Definition 2.4.1, the algorithm* $\text{GenericAttack}_{a,b,c,h,q}(F)$ *uses at most* q *calls to the* F *oracle and has ROM OW-CPA success probability* $1/2^b + q/2^{b+h}$ *against* $\text{GenericPKE}_{a,b,c,h}(F)$.

Proof. The algorithm calls the F oracle for $F(m_1, r_1)$; then, if $F(m_1, r_1) \neq z$, for $F(m_2, r_2)$; and so on through $F(m_q, r_q)$. This is at most q calls, and there are no other calls.

By definition the OW-CPA success probability of A against $\text{GenericPKE}_{a,b,c,h}(F)$ is the chance that the following game outputs 1: compute $(p, s) \leftarrow \text{KeyGen}()$; generate a uniform random $M \in \text{Plaintexts}$; compute $C \leftarrow \text{Encrypt}(M, p)$; output 1 if $A(p, C) = M$.

Write M as (ℓ, m). Then $C = (\ell, z)$ where $z = F(m, r)$ for some $r \in \{0,1\}^h$, by definition of Encrypt.

There is probability exactly $1/2^b$ that m_0 inside $A = \text{GenericAttack}_{a,b,c,h,q}(F)$ matches m. If this occurs then by distinctness none of $m_1, \ldots, m_q$ match m, so, by injectivity of F, none of the outputs $F(m_i, r_i)$ match z, so A does not stop early, so A outputs $(\ell, m_0) = (\ell, m) = M$, and the OW-CPA game outputs 1.

There is also, for each $i \in \{1, 2, \ldots, q\}$, probability exactly $1/2^{b+h}$ that (m_i, r_i) inside A matches (m, r). If this occurs then by distinctness none of $m_1, \ldots, m_{i-1}$ match m, so, by injectivity of F, none of the outputs $F(m_1, r_1)$ through $F(m_{i-1}, r_{i-1})$ match z, so A does not stop before reaching this i; A then tries $F(m_i, r_i)$, which matches $F(m, r) = z$, so A outputs $(\ell, m_i) = (\ell, m) = M$, and again the OW-CPA game outputs 1.

Conversely, these events are the only way for the OW-CPA game to output 1: if $A(p, C) = M$ then either A outputs $(\ell, m_0) = M$ in the last step, in which case $m_0 = m$, or it outputs some $(\ell, m_i) = M$ in the previous step, in which case $m_i = m$.

Finally, these events are disjoint by distinctness of $m_0, \ldots, m_q$, so they occur with total probability $1/2^b + q/2^{b+h}$. □

2.5 Optimality of the attack

This subsection shows that, given its number of calls to the F oracle, the attack above reaches the maximum possible OW-CPA success probability against $\text{GenericPKE}_{a,b,c,h}$. The fact that no attack can do better than probability $1/2^b + q/2^{b+h}$ against this PKE is what matters for seeing that derandomization damages security by a factor close to q; the fact that the specific attack above reaches probability $1/2^b + q/2^{b+h}$ shows that this OW-CPA analysis is complete.

The optimality proof relies on the fact that M is generated uniformly at random in the OW-CPA game, and that r is generated uniformly at random in Encrypt. These facts were not used in Theorem 2.4.2.

Theorem 2.5.1. *Under the assumptions of Definition 2.3.1, let* q *be an element of* $\{0, 1, \ldots, 2^b - 1\}$. *Every algorithm that uses at most* q *distinct calls to the* F *oracle has ROM OW-CPA success probability at most* $1/2^b + q/2^{b+h}$ *against* $\text{GenericPKE}_{a,b,c,h}(F)$.

Proof. Let A be an algorithm using at most q distinct calls to the F oracle. Modify A to count the number of distinct oracle inputs and, just before stopping, add extra calls to F on uniform random inputs until the count reaches q; this will terminate since the domain of F has size $2^{b+h} \geq 2^b > q$. Now A makes exactly q distinct oracle calls.

In the OW-CPA attack game for A, there are $\prod_{0 \leq j < 2^{b+h}} (2^c - j)$ equally likely possibilities for the injective function F; then 2^{a+b} equally likely possibilities for (ℓ, m) from Plaintexts; and 2^h equally likely possibilities for r inside $\text{Encrypt}(M, p)$, determining $C = (\ell, z)$ where $z = F(m, r)$.

A's initial view (p, C) reveals ℓ but provides no information about (m, r): for each choice of (m, r), there are exactly $\prod_{1 \leq j < 2^{b+h}} (2^c - j)$ choices of F satisfying $z = F(m, r)$. A's first oracle query (m_1, r_1), assuming $q \geq 1$, thus has $(m_1, r_1) = (m, r)$ with probability $1/2^{b+h}$, and $(m_1, r_1) \neq (m, r)$ with probability $1 - 1/2^{b+h}$.

Now condition on $(m_1, r_1) \neq (m, r)$. A's view after the oracle response z_1 provides no further information about (m, r): for each of the $2^{b+h}-1$ choices of $(m, r) \neq (m_1, r_1)$, there are exactly $\prod_{2 \leq j < 2^{b+h}} (2^c - j)$ choices of F satisfying $z = F(m, r)$ and $z_1 = F(m_1, r_1)$. A's second distinct oracle query (m_2, r_2), assuming $q \geq 2$, thus has $(m_2, r_2) = (m, r)$ with conditional probability $1/(2^{b+h}-1)$. The non-conditional probability that $(m_1, r_1) \neq (m, r)$ and $(m_2, r_2) \neq (m, r)$ is $1 - 2/2^{b+h}$.

Continue in the same way through all q distinct oracle queries. By induction, the total probability that $(m_1, r_1) \neq (m, r)$ and so on through $(m_i, r_i) \neq (m, r)$ is $1 - i/2^{b+h}$. A's view after oracle responses $z_1, \ldots, z_i$ provides no further information about (m, r): there are $2^{b+h} - i$ choices of (m, r) different from $(m_1, r_1), \ldots, (m_i, r_i)$, each produced by the same number of choices of F. A's next distinct oracle query (m_{i+1}, r_{i+1}), assuming $q \geq i + 1$, thus has $(m_{i+1}, r_{i+1}) = (m, r)$ with conditional probability $1/(2^{b+h} - i)$ if (m_{i+1}, r_{i+1}), i.e., non-conditional probability $1/2^{b+h}$, completing the induction for $i + 1$.

In particular, the total probability that $(m_1, r_1) \neq (m, r)$ and so on through $(m_q, r_q) \neq (m, r)$ is $1 - q/2^{b+h}$, and if this occurs then A's view after all q oracle responses provides no further information about (m, r). There are $2^{b+h} - q$ choices of (m, r) remaining at this point, and *at most* 2^h of them have (ℓ, m) matching the output from A, so A succeeds with conditional probability at most $2^h/(2^{b+h} - q)$; i.e., the non-conditional probability that A succeeds with $(m_1, r_1) \neq (m, r)$ and so on through $(m_q, r_q) \neq (m, r)$ is at most $1/2^b$. Meanwhile the probability that A succeeds with (m, r) matching one of $(m_1, r_1), \ldots, (m_q, r_q)$ is at most $q/2^{b+h}$. The total probability that A succeeds is at most $1/2^b + q/2^{b+h}$. □

2.6 The derandomized PKE

To keep this paper self-contained, this subsection defines $\text{TGenericPKE}_{a,b,c,h}$. The transformation from $\text{GenericPKE}_{a,b,c,h}$ to $\text{TGenericPKE}_{a,b,c,h}$ is an example of the standard T derandomization process from the literature.

Definition 2.6.1. *Under the assumptions of Definition 2.3.1, let H be a uniform random function from $\{0,1\}^{a+b}$ to $\{0,1\}^h$, and assume that F and H are independent. Then* $\text{TGenericPKE}_{a,b,c,h}(F, H)$ *is defined as*

$$(\text{PublicKeys}, \text{PrivateKeys}, \text{Plaintexts}, \text{Ciphertexts}, \text{KeyGen}, \text{TEncrypt}, \text{Decrypt})$$

where TEncrypt *is the following algorithm:*

- *Input* $((\ell, m), p) \in (\{0,1\}^a \times \{0,1\}^b) \times \{()\}$.
- *Compute* $r = H(\ell, m) \in \{0,1\}^h$.
- *Output* $(\ell, F(m, r)) \in \{0,1\}^a \times \{0,1\}^c$.

$\text{TGenericPKE}_{a,b,c,h}$ is the same as $\text{GenericPKE}_{a,b,c,h}$ except for replacing Encrypt with TEncrypt. The only difference between Encrypt and TEncrypt is that Encrypt generates r uniformly at random while TEncrypt generates r as $H(M)$, where $M = (\ell, m)$ is the plaintext being encrypted.

2.7 Attacking the derandomized PKE

This subsection defines an OW-CPA attack against $\text{TGenericPKE}_{a,b,c,h}$, and shows that the attack has success probability $(q+1)/2^b$, where q is the number of calls to the H oracle and also the number of calls to the F oracle. This completes the proof that derandomizing $\text{GenericPKE}_{a,b,c,h}$ damages security by a factor close to q.

To also complete the analysis of OW-CPA security of $\mathrm{TGenericPKE}_{a,b,c,h}$, one could ask for a proof that the following attack is optimal, but it is easier to observe that near-optimality follows from composing existing T theorems with Theorem 2.5.1.

Definition 2.7.1. *Under the assumptions of Definition 2.6.1, let q be an element of $\{0, 1, \ldots, 2^b - 1\}$, and define $\mathrm{TGenericAttack}_{a,b,c,h,q}(F, H)$ as the following algorithm:*

- *Input $(p, (\ell, z)) \in \{()\} \times (\{0,1\}^a \times \{0,1\}^c)$.*
- *Generate a uniform random sequence of distinct elements $m_0, m_1, \ldots, m_q$ of $\{0,1\}^b$.*
- *For each $i \in \{1, 2, \ldots, q\}$ in increasing order: If $F(m_i, H(\ell, m_i)) = z$, output (ℓ, m_i) and stop.*
- *Output (ℓ, m_0).*

In $\mathrm{TGenericAttack}_{a,b,c,h,q}$, each of the guesses $m_1, \ldots, m_q$ is correct with chance $1/2^b$—which, again, is much larger than $1/\#\mathrm{Plaintexts} = 1/2^{a+b}$ when a is large—and, critically, derandomization allows each of these guesses to be checked efficiently. For comparison, in $\mathrm{GenericAttack}_{a,b,c,h,q}$, each of the guesses $m_1, \ldots, m_q$ is correct with chance $1/2^b$, but checking a guess for m involves also guessing r, reducing the success chance of each guess to $1/2^{b+h}$.

Theorem 2.7.2. *Under the assumptions of Definition 2.7.1, the algorithm $\mathrm{TGenericAttack}_{a,b,c,h,q}(F)$ uses at most q calls to the F oracle, uses at most q calls to the H oracle, and has ROM OW-CPA success probability $(q+1)/2^b$ against $\mathrm{TGenericPKE}_{a,b,c,h}(F)$.*

Proof. As in Theorem 2.4.2, except that r_i is replaced by $H(\ell, m_i)$ and the success probabilities are adjusted accordingly. Full details are spelled out here to aid in verification.

The algorithm calls the H oracle and then the F oracle for $F(m_1, H(\ell, m_1))$; then, if the output was not z, for $F(m_2, H(\ell, m_2))$; and so on. This is at most q calls to H and at most q calls to F. There are no other oracle calls.

By definition the OW-CPA success probability of A against $\mathrm{TGenericPKE}_{a,b,c,h}(F)$ is the chance that the following game outputs 1: compute $(p, s) \leftarrow \mathrm{KeyGen}()$; generate a uniform random $M \in \mathrm{Plaintexts}$; compute $C \leftarrow \mathrm{TEncrypt}(M, p)$; output 1 if $A(p, C) = M$.

Write M as (ℓ, m). Then $C = (\ell, z)$ where $z = F(m, H(\ell, m))$, by definition of TEncrypt.

There is probability exactly $1/2^b$ that m_0 inside $A = \mathrm{TGenericAttack}_{a,b,c,h,q}(F)$ matches m. If this occurs then by distinctness none of $m_1, \ldots, m_q$ match m, so, by injectivity of F, none of the outputs $F(m_i, H(\ell, m_i))$ match z, so A does not stop early, so A outputs $(\ell, m_0) = (\ell, m) = M$, and the OW-CPA game outputs 1.

There is also, for each $i \in \{1, 2, \ldots, q\}$, probability exactly $1/2^b$ that m_i inside A matches m. If this occurs then by distinctness none of $m_1, \ldots, m_{i-1}$ match m, so, by injectivity of F, none of the outputs $F(m_1, H(\ell, m_1))$ through $F(m_{i-1}, H(\ell, m_{i-1}))$ match z, so A does not stop before reaching this i; A then tries $F(m_i, H(\ell, m_i))$, which matches $F(m, H(\ell, m)) = z$, so A outputs $(\ell, m_i) = (\ell, m) = M$, and again the OW-CPA game outputs 1.

Conversely, these events are the only way for the OW-CPA game to output 1: if $A(p, C) = M$ then either A outputs $(\ell, m_0) = M$ in the last step, in which case $m_0 = m$, or it outputs some $(\ell, m_i) = M$ in the previous step, in which case $m_i = m$.

Finally, these events are disjoint by distinctness of $m_0, \ldots, m_q$, so they occur with total probability $(q+1)/2^b$. □

3 Derandomizing a concrete PKE

Every ROM proof raises the question of whether the conclusion is an artifact of the ROM, i.e., whether extrapolating to concrete non-ROM proposals produces incorrect conclusions. Proofs generally do not address this question, so one falls back on cryptanalysis, searching for attacks against concrete proposals.

This section gives an example of a concrete PKE for which derandomization damages the pre-quantum OW-CPA security of the PKE against known attacks, in a way not explained by #Plaintexts. The damage is quantitatively similar to what happens in the second example in Section 2: derandomization makes known attacks easier by a factor growing linearly with the number of operations available to the attacker.

This is *not* a theorem regarding all attacks; it is conceivable that better attacks could change the status of this PKE. A close inspection also shows that, as in other areas of cryptanalysis (see generally [**17**, Appendix B]), the attack analyses rely on unproven conjectures. But any argument that derandomization is not risky (for large #Plaintexts) needs to explain how the argument is compatible not just with the proven ROM examples from Section 2 but also with the concrete example in this section.

This example is selected to rely entirely on well-known design techniques and well-known cryptanalytic techniques, reducing the chance of errors in the attack analysis. One could instead systematically survey previously published examples of PKEs and explore whether derandomization degrades the security of those PKEs. In general, this would be asking for new cryptanalysis, but, as Section 4 illustrates, there are cases where "derandomization attacks", meaning attacks exploiting derandomization, turn out to be as easy to write down as they are in this paper.

3.1 Is ElGamal an example?

Consider again the ElGamal PKE as in Section 1.1, with public key aG and ciphertext $(bG, M + abG)$, with a standard group as the plaintext space. Assume for simplicity that $\langle G \rangle$ is the whole group, not a proper subgroup.

As in Section 2, the attacker can enumerate guesses for (M, b), and, if this fails, output a final guess for M. Checking q guesses for (M, b) has success chance $q/\#\langle G \rangle^2$ and takes q simple operations. The final guess for M has success chance $1/\#\langle G \rangle$, which is dominant under the reasonable assumption that q is small compared to $\#\langle G \rangle$. Derandomization, choosing b as a hash of M, increases the success chance to $(q+1)/\#\langle G \rangle$.

One can object that this is not a tightness problem for large #Plaintexts: the attack has success chance only $(q+1)/\#\text{Plaintexts}$. However, modifying the PKE as in Section 2 to include additional information in plaintexts, leaked through ciphertexts, makes #Plaintexts much larger than $\#\langle G \rangle$, removing this objection. What matters is the success-probability ratio between attacks against the derandomized system and attacks against the original system.

A more serious objection is that there are much better attacks that instead spend q operations trying to compute the discrete logarithm a of aG. Even for our (conjecturally) strongest groups, generic attacks have success probability on the scale of $q^2/\#\langle G \rangle$, which is much larger than the probabilities $1/\#\langle G \rangle$ and $(q+1)/\#\langle G \rangle$ mentioned above. One is then faced with the question of whether derandomization allows q-operation attacks with higher success probability. This question does not appear to have been addressed in the cryptanalytic literature, so this paper moves on to another example.

3.2 Minimizing randomness in ElGamal plaintexts

A standard design technique in cryptography is to

- identify options for a specific component of a cryptographic system,

- restrict attention to options that reach a specified security level against known attacks according to a specified security metric, and
- choose the smallest option in a specified size metric.

The smallest option is typically described as being "efficient", while larger options are described as "wasting resources", being "overkill", etc. Consider, e.g., [7, Section 5] proposing usage of reduced-round ciphers "for a future where less energy is wasted on computing superfluous rounds".

Often this minimization of a cryptographic component is combined with an argument that larger options do not increase overall system security[3] beyond the specified constraint, given attacks against other components of the system. The larger options are then criticized as, e.g., being "unbalanced". NIST's official key-size recommendations for many years stated [9, Section 5.6.3] that combining "non-comparable strength" algorithms was "generally discouraged". For users of 256-bit ECC, this discouraged use of AES-256 and encouraged use of AES-128 instead, based on a security metric where AES-128 has "comparable strength" to 256-bit ECC while AES-256 has much higher strength.

This design approach often reduces security, for example because the specified security metric was too narrow. See, e.g., [12] showing that NIST's comparison between AES-128 and 256-bit ECC relies on considering only high-probability single-target attacks and fails when one considers a broader class of attacks. This section exploits a similar gap between different notions of security, after applying the following minimization to one component of the ElGamal PKE.

Consider the typical use of a PKE to communicate a random k-bit session key to achieve "k bits of security": for example, an AES-128 key for $k = 128$. ElGamal's plaintext M is not simply a k-bit key: it is a full-size group element, with many more than k bits of entropy—typically at least $2k$ bits, and sometimes even more to protect against known or suspected improvements in discrete-log attacks.

It is straightforward—see Example 3 below—to modify the ElGamal PKE for an "optimally efficient" plaintext space, the set $\{0,1\}^k$ of k-bit strings, exactly the set of session keys that the user wants to communicate. For comparison, the original message-space size "wastes precious randomness resources"; it is "overkill"; it is "unbalanced", since security of the whole PKE is certainly far below the group size.

This ElGamal modification is a simple example of cryptographic-component minimization. The component at issue is Plaintexts, the set of plaintexts. The specified security requirement for this component is that a guess for a secret (uniform random) plaintext succeeds with chance at most $1/2^k$. The size metric for this component is #Plaintexts. Certainly 2^k is smaller than $\#\langle G\rangle$. This ElGamal minimization is not new (see, e.g., [32, Section 5.1], using ElGamal to encrypt an encoding of a short session key); what is new here is the connection to derandomization.

3.3 Example 3: encoded-plaintext elliptic-curve ElGamal

Consider, in general, replacing ElGamal's $M + abG$ with $E(M) + abG$, where E is a public injection from Plaintexts to $\langle G\rangle$, easy to compute and easy to invert.

The special case Plaintexts $= \langle G\rangle$, with E as the identity map, is the original ElGamal system. As explained above, the generalization allows more "efficient" (meaning smaller) choices of #Plaintexts: specifically, #Plaintexts $= 2^k$ while $\#\langle G\rangle$ remains much larger than 2^k.

Take, in particular, Plaintexts $= \{0,1\}^k$, and define E as the composition of the following three steps:

- Zero-pad the k-bit input to $2h \geq k$ bits.

[3] Meanwhile the overall system *cost* rarely appears in the efficiency analysis.

- Map a $2h$-bit string (x_0, x_1) to a $2h$-bit string (x_4, x_5) defined by $x_2 = x_0 \oplus H(x_1)$, $x_3 = x_1 \oplus H(x_2)$, $x_4 = x_2 \oplus H(x_3)$, and $x_5 = x_3 \oplus H(x_4)$ where H is a standard h-bit hash function.
- Use Elligator [21] to map a $2h$-bit string invertibly to a point on a $(2h+1)$-bit elliptic curve.

Finally, Example 3 is this cryptosystem with a extra bits added into plaintexts and copied into ciphertexts, so that $\#\text{Plaintexts} = 2^{a+k}$.

The middle step in E is an example of what Rivest [69] dubbed an "all-or-nothing transform". This particular transform is from earlier work by Johnson–Matyas–Peyravian [50], adding more rounds to the transform used by Bellare–Rogaway [10] inside OAEP. If H were secret then this transform would instead be called a 4-round Feistel cipher.

When the elliptic curve is chosen according to standard criteria, the best discrete-log attack known has success probability on the scale of $q^2/\#\langle G\rangle \approx q^2/2^{2h+1}$ after q simple operations. If the discrete-log computation fails, a final guess for M succeeds with probability $1/2^k$. If parameters are chosen so that $2^k > q$ and, e.g., $2h > 3k + 10$ then the overall success probability is only slightly above $1/2^k$.

For the derandomized version of the same PKE, the attacker does much better by trying q guesses for M. The success probability of this attack is $q/2^k$ (plus $1/2^k$ if a random final output is included); i.e., approximately q times larger than the success probability of the attack against the randomized PKE. Instead of spending effort on a low-probability discrete-log computation, the attacker spends the same effort exploiting derandomization to check higher-probability guesses for M.

This is not the end of the analysis, since one still has to check whether there is a better attack against the randomized PKE. Standard curve criteria allow small cofactors, such as 4 or 8, and Elligator requires a cofactor. It is well known that the ElGamal PKE is not IND-CPA in the presence of these cofactors: the attacker learns the bottom 2 or 3 bits of a and b, partitioning the set of curve points $E(M)$ into 4 or 8 immediately recognizable classes. However, this merely allows the attacker to exclude approximately 3/4 or 7/8 of the possibilities for M (assuming E is well distributed across classes). This lets the attacker reach success probability approximately $4/2^k$ or $8/2^k$ by checking (on average) 4 or 8 possibilities for the final guess M, but this is not a powerful enough distinguisher to allow productive use of q guesses for M.

Could there be a stronger DDH attack? If the curve happens to allow a fast pairing then one can much more reliably check a guess for M—in other words, a guess for abG—by checking whether the pairing output for (G, abG) matches the pairing output for (aG, bG). However, standard curve criteria eliminate all curves where efficient pairings are known.

Section 1 noted the relatively low cryptanalytic attention to distinguishers as a reason that making IND-CPA assumptions is riskier than making OW-CPA assumptions. For the same reason, it is risky to assume that there is no DDH attack strong enough to invalidate this example. However, derandomization damages security of this example against *known* attacks.

3.4 Variants

One can replace the elliptic curve above with a multiplicative group $(\mathbf{Z}/p)^*$, where p is prime, and replace Elligator with simply viewing a $2h$-bit string as an integer between 1 and 2^{2h}. Known discrete-log attacks take time subexponential in $\log p$, but it is straightforward to take $\log p$ large enough that these attacks have success chance below $1/2^k$, assuming standard conjectures.

If $(p-1)/2$ is also prime then the cofactor is just 2. If also $p > 2^{2h+1}$ then one can square each integer modulo p and work in the subgroup of squares, with cofactor 1; this encoding function in the ElGamal context appears in, e.g., [41, Section 2.2].

Finally, one can construct examples that build a group element M in two parts, where one part ℓ is leaked through a larger cofactor while the other part m is limited to 2^k possibilities. This avoids the need to insert an extra string ℓ into plaintexts and ciphertexts. It is easy to construct multiplicative groups with a specified cofactor, by searching for primes p in an arithmetic progression. For elliptic-curve groups, the techniques of Bröker–Stevenhagen [30] allow efficient construction of a group of order N, given any N that factors into powers of a small number of known primes.

4 The derandomization inside FrodoKEM

It is claimed in [5, page 44] that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin". This claim refers to the round-3 FrodoKEM proposal to NISTPQC. That proposal includes three proposed parameter sets:[4] FrodoKEM-640, FrodoKEM-976, and FrodoKEM-1344, targeting the security levels of brute-force search for an AES-128 key, an AES-192 key, and an AES-256 key respectively.

Section 4.1 disproves this claim. Concretely, breaking IND-CCA2 for FrodoKEM-640 takes only 2^{128} guesses, with each guess being only a few bits more expensive than an AES-128 encryption. As in Example 1 (unlike Examples 2 and 3), this limit on the security level is explained by #Plaintexts. The attack is a simple brute-force search through PKE plaintexts, exploiting derandomization to be able to check each plaintext guess. Similar comments apply to FrodoKEM-976 (2^{192} guesses) and FrodoKEM-1344 (2^{256} guesses).

Section 4.2 applies the attack to a batch of targets. In short, if 2^{40} ciphertexts are sent to a FrodoKEM-640 public key, then breaking one of the ciphertexts takes only 2^{88} guesses. Section 4.2 also evaluates two different responses to multi-target attacks.

The consequences of this multi-target attack are sufficiently severe that, in response to the attack, the FrodoKEM team renamed FrodoKEM as "ephemeral FrodoKEM" and proposed a new patched "FrodoKEM". However, there has still been no erratum for the claim that "the FrodoKEM parameter sets comfortably match their target security levels with a large margin". The patch also does not rescue the claim.

4.1 Disproving the security claim at issue

FrodoPKE-640 has only 2^{128} plaintexts. Because of derandomization, there are only 2^{128} possible FrodoKEM-640 ciphertexts for any particular public key, and an attacker can check a plaintext guess against a ciphertext, exactly as in Example 1 from Section 2. Consequently, FrodoKEM-640 is vulnerable to a straightforward brute-force search through the 2^{128} plaintexts. Comparing to AES-128 thus boils down to comparing the cost of testing a FrodoKEM-640 plaintext guess to the cost of testing an AES-128 plaintext guess.

A reader who has heard that FrodoKEM is very slow might guess that this cost ratio is enough to rescue the claimed "large margin". Seeing that this is incorrect requires looking more closely at the attack details.

FrodoKEM-640 encryption involves multiplying a 640×640 matrix modulo 2^{15} by a 640×8 matrix, but testing whether a FrodoKEM-640 plaintext matches a given ciphertext is easier than this. It almost always suffices to test just one ciphertext position, using just 640 multiplications; only 1 in every 2^{15} guesses will pass this test. Even better, since FrodoKEM uses a power-of-2 modulus, one can use just 640 ANDs and 639 XORs to compute and check just the bottom bit of one ciphertext position. Only half of all guesses will pass this test, only half of those will pass the same test at another ciphertext position, and so on. There is still some overhead, for example to generate about 10KB of SHAKE

[4] Actually six, since each parameter set has one version using AES for matrix generation and one version using SHAKE for matrix generation. The difference has very little effect on the performance of the attack here.

output relevant to the first ciphertext positions (out of about 20KB overall); this accounts for about 2^{23} bit operations, since 1600-bit Keccak uses $1600 \cdot 84$ bit operations.

To do better, note that the IND-CCA2 definition provides a session key to the attacker, so one can skip all the multiplications involved in FrodoKEM and simply check the session-key hash. Most of the hash computation for the session key can be precomputed, since FrodoKEM's hash input puts the secret message *after* the public ciphertext. (For comparison, one rule of thumb for the order of hash inputs—see, e.g., [14]—is that whatever is least likely to be predictable by the attacker should come first.) This reduces the cost of an IND-CCA2 break to just two SHAKE blocks per message guess, around 2^{18} bit operations.

Compared to slightly under 2^{15} bit operations to test an AES-128 key (see [17]), this leaves FrodoKEM-640 with about 3 bits of security margin, not a "large margin". In a restricted attack model that sees only ciphertexts, FrodoKEM-640 has only about 8 bits of security margin, still not a "large margin".

Quantum attacks. Grover search is trivially applicable here, for example taking only about 2^{64} iterations for FrodoKEM-640 rather than 2^{128} iterations. Quantifying qubit operations per iteration is beyond the scope of this paper. For comparison, [5, Table 10] indicates only about a 10% difference between pre-quantum and post-quantum security levels for lattice attacks, and [5, page 44] says that "obtaining a quantum speed-up for sieving is rather tenuous".

Recall that, according to [28], FrodoKEM is under consideration by ISO. According to [16], ISO asks post-quantum encryption systems to achieve a "minimum security strength of 128 bits" in the "quantum model". The model does not seem to be publicly specified, but it would in any case be interesting to quantify how much impact derandomization attacks have upon FrodoKEM parameter choices if a user sets a minimum post-quantum security level rather than setting a minimum pre-quantum security level.

4.2 Multi-target attacks

The search through 2^{128} plaintexts in Section 4.1 is billions of times more expensive than a year of Bitcoin—obviously not something an attacker will carry out in the near future. However, deployed cryptosystems are normally used more than once, and attacking a large batch of targets can be more cost-effective than attacking a single target.

It is generally easy to prove that an attack breaking one of T targets cannot be more than $T\times$ better than a single-target attack. This is not comforting when a single-target attack has a security margin smaller than a factor T. For the type of attack considered here, breaking one of T ciphertexts sent to a single public key is in fact close to $T\times$ more efficient than a single-target attack—the "nightmare scenario" for a loose proof.

Concretely, assume that 2^{40} ciphertexts are sent to a FrodoKEM-640 public key and intercepted by an attacker. The attacker can then try 2^{88} guesses for the underlying FrodoPKE-640 plaintext, and compute 128 bits of the ciphertext for each guess. With chance $1-(1-2^{88}/2^{128})^{2^{40}} \approx 1-\exp(-1) = 0.632\ldots$, the guesses will match at least one of the 2^{40} target ciphertexts. One can use distinguished-point techniques to eliminate essentially all of the memory-access costs of checking for a match; see [63].

A simple, comprehensive response to a looseness factor T in theorems regarding T-target security is to increase parameters correspondingly, as in [4, Section 6.1]. This is a special case of the looseness adjustment covered in Section 1.2. For example, if the goal is 2^{128} multi-target security with at most $T = 2^{64}$ targets, then it suffices to set a goal of 2^{192} single-target security. AES-192, FrodoKEM-976, etc. claim to reach the latter goal. Of course, single-target security is sometimes misevaluated (as illustrated by FrodoKEM-976 incorrectly claiming to have a "large margin" beyond AES-192), but single-target security still has the advantage of being simpler to evaluate than multi-target security.

A more complicated, error-prone response is to patch specific cryptosystems to try to avoid multi-target weaknesses. Consider, e.g., NIST's stated reason [65] for continuing to recommend AES-128 in the context of post-quantum KEMs:

> Finally, you seem to be advocating that NIST respond to the possibility of multikey attacks by withdrawing AES128, rather than by advocating for modes of operation that have built-in multi-key security. Given that
>
> 1) AES-128 is the most widely used block cipher at present, and it has never come even close to being practically attacked based on an insufficiently large key size.
> 2) Most widely-used high-volume protocols, where multi-key security is actually a concern (e.g. TLS and IPSec) already have built-in protections against multi-key attacks.
>
> It seems premature to pull AES128.

The protections mentioned here are randomizing the inputs to various protocols using AES so that any specific input is not encrypted under many AES keys. This takes much more work to analyze than moving up to AES-192 (or AES-256). It also fails to protect a broader system that uses FrodoKEM-640 to communicate AES-128 keys in the first place: the attack directly breaks FrodoKEM-640 ciphertexts, independently of how the resulting session keys are used. Similarly, patching FrodoKEM-640 to try to stop multi-target attacks is more error-prone than moving up to FrodoKEM-976.

References

[1] — (no editor listed), *OpenSSH 9.0 release notes* (2022). URL: https://www.openssh.com/txt/release-9.0. Citations in this document: §1.2.

[2] — (no editor listed), *McEliece resource list* (2023). URL: https://mceliece.org. Citations in this document: §1.2.

[3] Carlisle Adams, Jan Camenisch (editors), *Selected areas in cryptography—SAC 2017, 24th international conference, Ottawa, ON, Canada, August 16–18, 2017, revised selected papers*, Lecture Notes in Computer Science, 10719, Springer, 2018. ISBN 978-3-319-72564-2. DOI: 10.1007/978-3-319-72565-9. See [19].

[4] Martin R. Albrecht, Daniel J. Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Kenneth G. Paterson, Edoardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, Jakub Szefer, Cen Jung Tjhai, Martin Tomlinson, Wen Wang, *Classic McEliece: conservative code-based cryptography: guide for security reviewers* (2022). URL: https://classic.mceliece.org/mceliece-security-20221023.pdf. Citations in this document: §1.2, §4.2.

[5] Erdem Alkim, Joppe W. Bos, Léo Ducas, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Chris Peikert, Ananth Raghunathan, Douglas Stebila, *FrodoKEM: Learning With Errors key encapsulation: algorithm specifications and supporting documentation* (2021). URL: https://web.archive.org/web/20220119174856/https://frodokem.org/files/FrodoKEM-specification-20210604.pdf. Citations in this document: §1.1, §1.2, §1.4, §4, §4.1, §4.1.

[6] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, Peter Schwabe, *Post-quantum key exchange—a new hope*, in USENIX 2016 [47] (2016), 327–343. URL: https://eprint.iacr.org/2015/1092. Citations in this document: §1.1.

[7] Jean-Philippe Aumasson, *Too much crypto* (2019). URL: https://eprint.iacr.org/2019/1492. Citations in this document: §3.2.

[8] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, Damien Stehlé, *CRYSTALS-Kyber: Algorithm specifications and supporting documentation (version 3.02)* (2021). URL: https://web.archive.org/web/20211215150153/https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf. Citations in this document: §1.1, §1.2.

[9] Elaine Barker, William Barker, William Burr, William Polk, Miles Smid, *Recommendation for key management—part 1: general (revised)*, NIST Special Publication 800-57 (2007). URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-57p1r2007.pdf. Citations in this document: §3.2.

[10] Mihir Bellare, Phillip Rogaway, *Optimal asymmetric encryption—how to encrypt with RSA*, in Eurocrypt 1994 [72] (1995), 92–111. URL: https://cseweb.ucsd.edu/~mihir/pubs.html. DOI: 10.1007/BFB0053428. Citations in this document: §3.3.

[11] Mihir Bellare, Phillip Rogaway, *The exact security of digital signatures: how to sign with RSA and Rabin*, in Eurocrypt 1996 [57] (1996), 399–416. URL: https://cseweb.ucsd.edu/~mihir/pubs.html. DOI: 10.1007/3-540-68339-9_34. Citations in this document: §1.2, §1.2.

[12] Daniel J. Bernstein, *Break a dozen secret keys, get a million more for free* (2015). URL: https://blog.cr.yp.to/20151120-batchattacks.html. Citations in this document: §3.2.

[13] Daniel J. Bernstein, *Comparing proofs of security for lattice-based encryption*, Second PQC Standardization Conference (2019). URL: https://cr.yp.to/papers.html#latticeproofs. Citations in this document: §1.1.

[14] Daniel J. Bernstein, *Re: Was: eddsa (un)suited for mandatory to implement ciphersuite?* (2022). URL: https://mailarchive.ietf.org/arch/msg/cfrg/GRigAYvZ8-Z8qmxJ1jOiKR8eLyQ/. Citations in this document: §4.1.

[15] Daniel J. Bernstein, *Quantifying risks in cryptographic selection processes* (2023). URL: https://cr.yp.to/papers.html#qrcsp. Citations in this document: §1.7.

[16] Daniel J. Bernstein, *Ensuring time for security review* (2023). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/W2VOzy0wz_E/m/4yArnctiAQAJ. Citations in this document: §4.1.

[17] Daniel J. Bernstein, Tung Chou, *CryptAttackTester: high-assurance attack analysis*, in Crypto 2024 [68] (2024), 141–182. URL: https://cr.yp.to/papers.html#cat. DOI: 10.1007/978-3-031-68391-6_5. Citations in this document: §1.2, §3, §4.1.

[18] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal, *NTRU Prime: reducing attack surface at low cost*, full version of [19] (2017). URL: https://ntruprime.cr.yp.to/papers.html. Citations in this document: §1.2.

[19] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal, *NTRU Prime: reducing attack surface at low cost*, in SAC 2017 [**3**], abbreviated version of [**18**] (2018), 235–260. DOI: 10.1007/978-3-319-72565-9_12.

[20] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, Christine van Vredendaal, *NTRU Prime: round 2* (2019). URL: https://web.archive.org/web/20240303202622/https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions. Citations in this document: §1.2.

[21] Daniel J. Bernstein, Mike Hamburg, Anna Krasnova, Tanja Lange, *Elligator: elliptic-curve points indistinguishable from uniform random strings*, in CCS 2013 [**70**] (2013), 967–980. URL: https://eprint.iacr.org/2013/325. DOI: 10.1145/2508859.2516734. Citations in this document: §3.3.

[22] Daniel J. Bernstein, Edoardo Persichetti, *Towards KEM unification* (2018). URL: https://cr.yp.to/papers.html#tightkem. Citations in this document: §1.3, §1.7, §1.7.

[23] Eli Biham (editor), *Fast software encryption, 4th international workshop, FSE '97, Haifa, Israel, January 20–22, 1997, proceedings*, Springer, 1997. ISBN 3-540-63247-6. DOI: 10.1007/BFB0052329. See [69].

[24] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, Edoardo Persichetti, *Tighter proofs of CCA security in the quantum random oracle model*, in TCC 2019 [**46**] (2019), 61–90. URL: https://eprint.iacr.org/2019/590. DOI: 10.1007/978-3-030-36033-7_3. Citations in this document: §1.3.

[25] Mario Blaum, Patrick G. Farrell, Henk C. A. van Tilborg (editors), *Information, coding and mathematics*, Kluwer International Series in Engineering and Computer Science, 687, Kluwer, 2002. MR 2005a:94003. See [79].

[26] Blockchain.com, *Total hash rate (TH/s)* (2023). URL: https://www.blockchain.com/explorer/charts/hash-rate. Citations in this document: §1.2.

[27] Dan Boneh, *The decision Diffie-Hellman problem*, in ANTS 1998 [**31**] (1998), 48–63. URL: https://crypto.stanford.edu/~dabo/pubs/papers/DDH.pdf. DOI: 10.1007/BFB0054851. Citations in this document: §1.1.

[28] Joppe Bos, Christine Cloostermans, Joost Renes, Olivier Bronchain, Frank Custers, *Conservative post-quantum security with FrodoKEM* (2023). URL: https://web.archive.org/web/20241001200453/https://www.nxp.com/company/about-nxp/smarter-world-blog/BL-POST-QUANTUM-SECURITY-WITH-FRODOKEM. Citations in this document: §1.2, §4.1.

[29] Colin Boyd, Leonie Simpson (editors), *Information security and privacy—18th Australasian conference, ACISP 2013, Brisbane, Australia, July 1–3, 2013. proceedings*, Springer, 2013. ISBN 978-3-642-39058-6. DOI: 10.1007/978-3-642-39059-3. See [41].

[30] Reinier Bröker, Peter Stevenhagen, *Efficient CM-constructions of elliptic curves over finite fields*, Mathematics of Computation **76** (2007), 2161–2179. URL: https://www.ams.org/journals/mcom/2007-76-260/S0025-5718-07-01980-1/S0025-5718-07-01980-1.pdf. DOI: 10.1090/S0025-5718-07-01980-1. Citations in this document: §3.4.

[31] Joe Buhler (editor), *Algorithmic number theory, third international symposium, ANTS-III, Portland, Oregon, USA, June 21–25, 1998, proceedings*, 1423, Springer, 1998. ISBN 3-540-64657-4. DOI: 10.1007/BFB0054849. See [27].

[32] Jon Callas, Lutz Donnerhacke, Hal Finney, David Shaw, Rodney Thayer, *OpenPGP message format* (2007). URL: https://datatracker.ietf.org/doc/html/rfc4880. Citations in this document: §3.2.

[33] Peter Campbell, *Re: [CFRG] Classic McEliece* (2023). URL: https://mailarchive.ietf.org/arch/msg/cfrg/m07j-Gsrb6KWyQxNSKkdSxSiu8M/. Citations in this document: §1.2.

[34] Anne Canteaut, Yuval Ishai (editors), *Advances in cryptology—EUROCRYPT 2020—39th annual international conference on the theory and applications of cryptographic techniques, Zagreb, Croatia, May 10–14, 2020, proceedings, part III*, Springer, 2020. ISBN 978-3-030-45726-6. DOI: 10.1007/978-3-030-45727-3. See [56].

[35] Rory Carroll, *Welcome to Utah, the NSA's desert home for eavesdropping on America* (2013). URL: https://www.theguardian.com/world/2013/jun/14/nsa-utah-data-facility. Citations in this document: §1.2.

[36] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hulsing, Joost Rijneveld, John M. Schanck, Peter Schwabe, William Whyte, Zhenfei Zhang, *NTRU: algorithm specifications and supporting documentation* (2019). URL: https://web.archive.org/web/20240303202622/https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions. Citations in this document: §1.2.

[37] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, Peter Schwabe, *MQDSS specifications* (2019). URL: https://web.archive.org/web/20220120235746/https://mqdss.org/files/MQDSS_Ver2.pdf. Citations in this document: §1.2.

[38] Alexander W. Dent, *A designer's guide to KEMs*, in Cirencester 2003 [**64**] (2003), 133–151. URL: https://eprint.iacr.org/2002/174. DOI: 10.1007/978-3-540-40974-8_12. Citations in this document: §1.3, §1.3.

[39] Whitfield Diffie, Martin Hellman, *New directions in cryptography*, IEEE Transactions on Information Theory **22** (1976), 644–654. ISSN 0018-9448. MR 55:10141. URL: https://ee.stanford.edu/~hellman/publications/24.pdf. DOI: 10.1109/TIT.1976.1055638. Citations in this document: §1.1.

[40] Taher ElGamal, *A public key cryptosystem and a signature scheme based on discrete logarithms*, IEEE Transactions on Information Theory **31** (1985), 469–472. ISSN 0018-9448. MR 86j:94045. DOI: 10.1109/TIT.1985.1057074. Citations in this document: §1.1.

[41] Pierre-Alain Fouque, Antoine Joux, Mehdi Tibouchi, *Injective encodings to elliptic curves*, in ACISP 2013 [**29**] (2013), 203–218. URL: https://eprint.iacr.org/2013/373. DOI: 10.1007/978-3-642-39059-3_14. Citations in this document: §3.4.

[42] Eiichiro Fujisaki, Tatsuaki Okamoto, *Secure integration of asymmetric and symmetric encryption schemes*, in Crypto 1999 [**80**] (1999), 537–554. URL: https://link.springer.com/content/pdf/10.1007/3-540-48405-1_34.pdf. DOI: 10.1007/3-540-48405-1_34. Citations in this document: §1.

[43] Oded Goldreich, *On post-modern cryptography* (2006). URL: https://eprint.iacr.org/2006/461. Citations in this document: §1.1.

[44] Chris Hall, Ian Goldberg, Bruce Schneier, *Reaction attacks against several public-key cryptosystems*, in ICICS 1999 [**78**] (1999), 2–12. URL: https://cypherpunks.ca/~iang/pubs/paper-reaction-attacks.pdf. DOI: 10.1007/978-3-540-47942-0_2. Citations in this document: §1.1.

[45] Dennis Hofheinz, Kathrin Hövelmanns, Eike Kiltz, *A modular analysis of the Fujisaki-Okamoto transformation*, in TCC 2017-1 [**51**] (2017), 341–371. URL: https://eprint.iacr.org/2017/604. DOI: 10.1007/978-3-319-70500-2_12. Citations in this document: §1.2, §1.3, §1.3, §1.3, §1.3, §2.3.

[46] Dennis Hofheinz, Alon Rosen (editors), *Theory of cryptography—17th international conference, TCC 2019, Nuremberg, Germany, December 1–5, 2019, proceedings, part II*, Lecture Notes in Computer Science, 11892, Springer, 2019. ISBN 978-3-030-36032-0. DOI: 10.1007/978-3-030-36033-7. See [24].

[47] Thorsten Holz, Stefan Savage (editors), *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10–12, 2016*, USENIX Assocation, 2016. See [6].

[48] ISE Crypto PQC working group, *Securing tomorrow today: Why Google now protects its internal communications from quantum threats* (2022). URL: https://web.archive.org/web/20240107055046/https://cloud.google.com/blog/products/identity-security/why-google-now-uses-post-quantum-cryptography-for-internal-comms. Citations in this document: §1.2.

[49] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, Zhi Ma, *IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited*, in Crypto 2018 [**73**] (2018), 96–125. URL: https://eprint.iacr.org/2017/1096. DOI: 10.1007/978-3-319-96878-0_4. Citations in this document: §1.3.

[50] Don B. Johnson, Stephen M. Matyas, Mohammad Peyravian, *Encryption of long blocks using a short-block encryption procedure* (1996). URL: https://web.archive.org/web/20001017235026/https://grouper.ieee.org/groups/1363/P1363a/contributions/peyrav.pdf. Citations in this document: §3.3.

[51] Yael Kalai, Leonid Reyzin (editors), *Theory of cryptography—15th international conference, TCC 2017, Baltimore, MD, USA, November 12–15, 2017, proceedings, part I*, Lecture Notes in Computer Science, 10677, Springer, 2017. ISBN 978-3-319-70499-9. DOI: 10.1007/978-3-319-70500-2. See [45].

[52] Daniel Kales, Greg Zaverucha, *Forgery attacks on MQDSSv2.0* (2019). URL: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-2/official-comments/MQDSS-round2-official-comment.pdf. Citations in this document: §1.2, §1.2.

[53] Joe Kilian (editor), *Advances in cryptology—CRYPTO 2001, 21st annual international cryptology conference, Santa Barbara, California, USA, August 19–23, 2001, proceedings*, Lecture Notes in Computer Science, 2139, Springer, 2001. ISBN 3-540-42456-3. MR 2003d:94002. DOI: 10.1007/3-540-44647-8. See [76].

[54] Kwangjo Kim (editor), *Public key cryptography, 4th international workshop on practice and theory in public key cryptography, PKC 2001, Cheju Island, Korea, February 13–15, 2001, proceedings*, 1992, Springer, 2001. ISBN 3-540-41658-7. DOI: 10.1007/3-540-44586-2. See [62].

[55] Neal Koblitz, Alfred Menezes, *Critical perspectives on provable security: fifteen years of "another look" papers*, Advances in Mathematics of Communications **13** (2019), 517–558. URL: https://eprint.iacr.org/2019/1336. DOI: 10.3934/AMC.2019034. Citations in this document: §1.2.

[56] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, Shifeng Sun, *Measure-Rewind-Measure: tighter quantum random oracle model proofs for one-way to hiding and CCA security*, in Eurocrypt 2020 [**34**] (2020), 703–728. URL: https://eprint.iacr.org/2021/454. DOI: 10.1007/978-3-030-45727-3_24. Citations in this document: §1.2, §1.3.

[57] Ueli M. Maurer (editor), *Advances in cryptology—EUROCRYPT '96: proceedings of the fifteenth international conference on the theory and application of cryptographic techniques held in Saragossa, May 12–16, 1996*, Lecture Notes in Computer Science, 1070, Springer, 1996. ISBN 3-540-61186-X. MR 97g:94002. DOI: 10.1007/3-540-68339-9. See [11].

[58] Alfred Menezes, *Another look at provable security* (2012). URL: https://www.iacr.org/conferences/eurocrypt2012/Program/Weds/Menezes.pdf. Citations in this document: §1.2.

[59] Shiho Moriai, Huaxiong Wang (editors), *Advances in cryptology—ASIACRYPT 2020—26th international conference on the theory and application of cryptology and information security, Daejeon, South Korea, December 7–11, 2020, proceedings, part I*, Springer, 2020. ISBN 978-3-030-64836-7. DOI: 10.1007/978-3-030-64837-4. See [77].

[60] National Institute of Standards and Technology, *FIPS 203 (Draft): Module-lattice-based key-encapsulation mechanism standard* (2023). URL: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.203.ipd.pdf. Citations in this document: §1.6.

[61] Jesper Buus Nielsen, Vincent Rijmen (editors), *Advances in cryptology—EUROCRYPT 2018—37th annual international conference on the theory and applications of cryptographic techniques, Tel Aviv, Israel, April 29–May 3, 2018, proceedings, part III*, Lecture Notes in Computer Science, 10822, Springer, 2018. ISBN 978-3-319-78371-0. DOI: 10.1007/978-3-319-78372-7. See [71].

[62] Tatsuaki Okamoto, David Pointcheval, *The gap-problems: a new class of problems for the security of cryptographic schemes*, in PKC 2001 [**54**] (2001), 104–118. URL: https://www.di.ens.fr/david.pointcheval/Documents/Papers/2001_pkc.pdf. DOI: 10.1007/3-540-44586-2_8. Citations in this document: §1.1.

[63] Paul C. van Oorschot, Michael Wiener, *Parallel collision search with cryptanalytic applications*, Journal of Cryptology **12** (1999), 1–28. ISSN 0933–2790. URL: https://people.scs.carleton.ca/~paulv/papers/pubs.html. DOI: 10.1007/PL00003816. Citations in this document: §4.2.

[64] Kenneth G. Paterson (editor), *Cryptography and coding, 9th IMA international conference, Cirencester, UK, December 16–18, 2003, proceedings*, Lecture Notes in Computer Science, 2898, Springer, 2003. ISBN 3-540-20663-9. DOI: 10.1007/B93924. See [38].

[65] Ray Perlner, *Re: post-quantum benchmarking and RNGs* (2017). URL: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/PW6GF-wHGFE/m/nnAnuUAUAAAJ. Citations in this document: §4.2.

[66] Edoardo Persichetti, *Improving the efficiency of code-based cryptography*, Ph.D. thesis, University of Auckland, 2012. URL: https://www.math.auckland.ac.nz/~sgal018/EdoardoPhD.pdf. Citations in this document: §1.3.

[67] Bart Preneel (editor), *Advances in cryptology—EUROCRYPT 2000, international conference on the theory and application of cryptographic techniques, Bruges, Belgium, May 14–18, 2000, proceeding*, Lecture Notes in Computer Science, 1807, Springer, 2000. ISBN 3-540-67517-5. DOI: 10.1007/3-540-45539-6. See [74].

[68] Leonid Reyzin, Douglas Stebila (editors), *Advances in cryptology—CRYPTO 2024—44th annual international cryptology conference, Santa Barbara, CA, USA, August 18–22, 2024, proceedings, part VI*, 14925, Springer, 2024. ISBN 978-3-031-68390-9. DOI: 10.1007/978-3-031-68391-6. See [17].

[69] Ronald L. Rivest, *All-or-nothing encryption and the package transform*, in FSE 1997 [**23**] (1997), 210–218. URL: https://people.csail.mit.edu/rivest/pubs/Riv97d.pdf. DOI: 10.1007/BFB0052348. Citations in this document: §3.3.

[70] Ahmad-Reza Sadeghi, Virgil D. Gligor, Moti Yung (editors), *2013 ACM SIGSAC conference on computer and communications security, CCS'13, Berlin, Germany, November 4–8, 2013*, ACM, 2013. ISBN 978-1-4503-2477-9. See [21].

[71] Tsunekazu Saito, Keita Xagawa, Takashi Yamakawa, *Tightly-secure key-encapsulation mechanism in the quantum random oracle model*, in Eurocrypt 2018 [**61**] (2018), 520–551. URL: https://eprint.iacr.org/2017/1005. DOI: 10.1007/978-3-319-78372-7_17. Citations in this document: §1.3, §1.3.

[72] Alfredo De Santis (editor), *Advances in cryptology—EUROCRYPT '94, workshop on the theory and application of cryptographic techniques, Perugia, Italy, May 9–12, 1994, proceedings*, Lecture Notes in Computer Science, 950, Springer, 1995. ISBN 3-540-60176-7. MR 98h:94001. DOI: 10.1007/BFB0053418. See [10].

[73] Hovav Shacham, Alexandra Boldyreva (editors), *Advances in cryptology—CRYPTO 2018—38th annual international cryptology conference, Santa Barbara, CA, USA, August 19–23, 2018, proceedings, part III*, Springer, 2018. ISBN 978-3-319-96877-3. DOI: 10.1007/978-3-319-96878-0. See [49].

[74] Victor Shoup, *Using hash functions as a hedge against chosen ciphertext attack*, in Eurocrypt 2000 [**67**] (2000), 275–288. URL: https://shoup.net/papers/hedge.pdf. DOI: 10.1007/3-540-45539-6_19. Citations in this document: §1.1, §1.3.

[75] Victor Shoup, *A proposal for an ISO standard for public key encryption*, version 2.1 (2001). URL: http://shoup.net/papers/iso-2_1.pdf. Citations in this document: §1.3.

[76] Victor Shoup, *OAEP reconsidered*, in Crypto 2001 [**53**] (2001), 239–259. URL: https://shoup.net/papers/oaep.pdf. DOI: 10.1007/3-540-44647-8_15. Citations in this document: §2.1.

[77] Dominique Unruh, *Post-quantum verification of Fujisaki-Okamoto*, in Asiacrypt 2020 [**59**] (2020), 321–352. URL: https://eprint.iacr.org/2020/962. DOI: 10.1007/978-3-030-64837-4_11. Citations in this document: §1.3.

[78] Vijay Varadharajan, Yi Mu (editors), *Information and communication security, second international conference, ICICS'99, Sydney, Australia, November 9–11, 1999, proceedings*, Springer, 1999. ISBN 3-540-66682-6. DOI: 10.1007/B72329. See [44].

[79] Eric R. Verheul, Jeroen M. Doumen, Henk C. A. van Tilborg, *Sloppy Alice attacks! Adaptive chosen ciphertext attacks on the McEliece public-key cryptosystem*, in [**25**] (2002), 99–119. MR 2005b:94041. URL: https://ris.utwente.nl/ws/portalfiles/portal/298823820/Verheul_sloppy.pdf. Citations in this document: §1.1.

[80] Michael J. Wiener (editor), *Advances in cryptology—CRYPTO '99, 19th annual international cryptology conference, Santa Barbara, California, USA, August 15–19, 1999, proceedings*, Lecture Notes in Computer Science, 1666, Springer, 1999. ISBN 3-540-66347-9. DOI: 10.1007/3-540-48405-1. See [42].