

TWO LAWS OF PUBLIC-KEY CRYPTOGRAPHY

TREY LI

ABSTRACT. We discover two laws governing public-key key exchange and public-key encryption. The first states that a non-interactive key-exchange scheme is secure only if it is kleptographically insecure. The second states that a public-key encryption scheme is secure only if it is kleptographically insecure. They give two impossibility results: no kleptographically secure non-interactive key exchange with pseudorandom shared keys exists, and no kleptographically secure public-key encryption with pseudorandom ciphertexts exists. More precisely, no non-interactive key-exchange scheme can simultaneously establish pseudorandom shared keys and support black-box execution in which the user observes only the public keys and the final shared keys output by the black box, and no public-key encryption scheme can simultaneously have pseudorandom ciphertexts and support black-box execution in which the user observes only the public keys and plaintexts input to the black box and the ciphertexts output by it. Our argument turns the very shield of a primitive into a spear against itself, showing that the security guarantee of the primitive is precisely what enables a kleptographic attack. We instantiate the laws with two post-quantum schemes: CSIDH and Regev encryption.

1. INTRODUCTION

Diffie and Hellman introduced public-key cryptography in 1976 [1]. The insight is to perform cryptographic tasks without preshared secrets. Two typical examples are public-key key exchange and public-key encryption. In public-key key exchange, two parties remotely establish a common secret. In public-key encryption, two parties remotely perform private communication without relying on a preshared common secret.

Young and Yung introduced the Secretly Embedded Trapdoor with Universal Protection (SETUP) in 1996 [2] and introduced kleptography in 1997 [3]. The insight is to use cryptography against cryptography itself. A cryptographic implementation may be modified so that it secretly gives an advantage to a kleptographer while still appearing to behave honestly. In a key-exchange SETUP, the key-exchange device is treated as a black box. The user sees only the public keys and the final shared keys output by the device. In a public-key encryption SETUP, the encryption device is treated as a black box. The user sees only the public key and plaintext input to the device and the ciphertext output by it. Previous work on kleptography [2, 3], algorithm-substitution attacks [4–6], public-key steganography [7], and anamorphic encryption [8–10] has developed methods for constructing kleptographic subversions, as well as mechanisms for resisting them [11, 12]. They primarily focus on how to achieve attacks or defenses, and for this purpose may employ additional assumptions, cryptographic primitives, or mechanisms beyond the properties of the underlying scheme itself.

We consider a more fundamental question about the nature of security and attacks. We study the phenomena of public-key security and kleptographic insecurity from an intrinsic point of view. We ask whether there are laws governing the relation between public-key cryptography and kleptography. We ask whether kleptographic insecurity is an accidental property of particular constructions or an intrinsic consequence of public-key cryptography. We ask whether natural properties of a public-key primitive already force the existence of the kleptographic phenomenon. Our results give positive answers.

Throughout the paper, we work mostly at the most basic abstract level. However, the insight can already be seen concretely in Diffie-Hellman key exchange. In the usual mathematical setting, Alice and Bob exchange g^a and g^b and derive the shared key $\text{KDF}(g^{ab})$ which is a pseudorandom binary string. Suppose Alice performs the key exchange inside a black-box device. A malicious Eve, who may be the device vendor, can embed her public key g^e into the device before selling it to Alice. When the device samples a and computes g^a for Alice, it additionally computes $\text{KDF}(g^{ae})$. Alice then sends g^a to Bob over a public channel. Eve observes g^a and computes the same key $\text{KDF}(g^{ae})$ as in Alice’s device. Precisely because of the security of the key-exchange scheme, $\text{KDF}(g^{ae})$ is pseudorandom to Alice and other observers. The device can then use this hidden pseudorandom string against the key-exchange scheme itself. It can use $\text{KDF}(g^{ae})$ to signal a bit of Alice and Bob’s shared key $\text{KDF}(g^{ab})$ through later public outputs and thus break the security of the key-exchange scheme. Moreover, if $\text{KDF}(g^{ae})$ is sufficiently long, it can even use $\text{KDF}(g^{ae})$ as the randomness for Alice’s next key generation: Since Eve knows the same value $\text{KDF}(g^{ae})$, she can reproduce that key generation and recover Alice’s next secret key. The success of the kleptographic attack is guaranteed precisely by the security of the key-exchange scheme itself and by nothing more.

2. LAW OF PUBLIC-KEY KEY EXCHANGE

In this paper, by public-key key exchange we always mean non-interactive key exchange (NIKE) [13].

Definition 1 (KE). *A public-key key-exchange scheme consists of two (possibly identical) key-generation algorithms KeyGen_A and KeyGen_B , and two (possibly identical) shared-key algorithms Shared_A and Shared_B . Alice and Bob generate key pairs $(sk_A, pk_A) \leftarrow \text{KeyGen}_A$ and $(sk_B, pk_B) \leftarrow \text{KeyGen}_B$, respectively, exchange pk_A and pk_B , and derive a common key as $\text{Shared}_A(sk_A, pk_A, pk_B) = \text{Shared}_B(sk_B, pk_A, pk_B)$.*

This formulation is not merely conventional but unavoidable. In particular, all six elements are necessary: sk_A , sk_B , pk_A , pk_B , a nontrivial dependence between sk_A and pk_A , and a nontrivial dependence between sk_B and pk_B . First, neither pk_A nor pk_B can be missing. If Alice sends no public information pk_A to Bob, then Bob has no information that distinguishes Alice from a third party. Alice therefore has no advantage over a third party in establishing a secret with Bob. The same argument applies to Bob. Second, neither sk_A nor sk_B can be missing. If Alice has no secret information sk_A , then everything available to her is also available to a third party. She therefore has no advantage over the third party in establishing a secret with Bob. The same argument applies to Bob. Third, pk_A must depend nontrivially on sk_A , and pk_B must depend nontrivially on sk_B . If pk_A were independent of sk_A , then knowing sk_A would give Alice no advantage in using pk_A to establish a secret with Bob. The same argument applies to (sk_B, pk_B) . Finally, the secret key must not be efficiently recoverable from the public key. Otherwise, a third party could recover the secret key and obtain the same advantage as the legitimate party.

Definition 2 (KE-Security). *A public-key key-exchange scheme is secure if, for any polynomial number of independently generated key pairs, any polynomial-size collection of shared keys established between distinct pairs of parties is jointly computationally indistinguishable from independent uniformly random keys of the same lengths, given all public keys.*

This definition of security is basic and natural: If public-key key exchange is intended to replace the private distribution of independently generated random keys, then the shared keys it establishes should be jointly indistinguishable from independent random keys.

We next define key-exchange SETUP through three basic axioms. Before that, we first define black-box execution of a key exchange protocol for one participant.

Definition 3 (Black-Box KE Execution). *In a black-box execution for Alice, the device first runs $(sk_A, pk_A) \leftarrow \text{KeyGen}_A$ and outputs pk_A . It then receives Bob's honestly generated public key pk_B and outputs $K = \text{Shared}_A(sk_A, pk_A, pk_B)$. The user observes only pk_A , pk_B , and K . The secret key sk_A , the internal randomness, and the internal state of the device are hidden. Over multiple executions, the observable transcript consists only of the corresponding sequence of public keys and final shared keys.*

Definition 4 (KE-SETUP). *A key-exchange SETUP is a stateful subverted implementation of a secure key-exchange scheme satisfying the following three properties:*

- (1) *it breaks the security of the scheme for the kleptographer;*
- (2) *over any polynomial number of black-box executions, its observable transcript is computationally indistinguishable from that of an honest implementation;*
- (3) *no secret of the kleptographer is revealed even if the device is fully reverse-engineered.*

A secure key-exchange scheme is said to be kleptographically insecure if it admits a key-exchange SETUP. It is said to be kleptographically secure if it is not kleptographically insecure.

The following theorem shows that every secure public-key key-exchange scheme is kleptographically insecure.

Theorem 1 (First Law of Public-Key Cryptography). *A public-key key-exchange scheme is secure¹ only if it is kleptographically insecure.*

Proof. Let Eve be the kleptographer, who may be a device vendor. She generates a legitimate key pair $(sk_E, pk_E) \leftarrow \text{KeyGen}_B$ and embeds pk_E into the device sold to Alice. In Alice's first execution, the device honestly generates $(sk_A^{(1)}, pk_A^{(1)})$ and outputs $pk_A^{(1)}$. In addition, it computes $z = \text{Shared}_A(sk_A^{(1)}, pk_A^{(1)}, pk_E)$. Eve

¹In the sense of Definition 2.

observes $pk_A^{(1)}$ and computes $z' = \text{Shared}_B(sk_E, pk_A^{(1)}, pk_E)$. By key-exchange correctness, $z = z'$. After receiving Bob's public key $pk_B^{(1)}$, the device computes the legitimate shared key $K_{AB}^{(1)} = \text{Shared}_A(sk_A^{(1)}, pk_A^{(1)}, pk_B^{(1)})$. By key-exchange security, z and $K_{AB}^{(1)}$ are jointly computationally indistinguishable from two independent uniformly random keys given all public keys. In particular, z remains computationally indistinguishable from uniform even given $K_{AB}^{(1)}$ and all public information.

In order to break the security of the scheme for Eve, the device now tries to leak one bit of $K_{AB}^{(1)}$ to Eve. Let $b \in \{0, 1\}$ be the first bit of $K_{AB}^{(1)}$, and let r be the first bit of z . It sets $c = b \oplus r$. In the next two executions, the device independently generates two fresh key pairs $(sk_A^{(2)}, pk_A^{(2)})$ and $(sk_A^{(3)}, pk_A^{(3)})$. Fix a public canonical total order on public keys. If $c = 0$, the device outputs the two public keys in increasing order. If $c = 1$, it outputs them in decreasing order. Eve observes the order and recovers c . Since Eve knows r , she then recovers $b = c \oplus r$. The only exception that fails the recovery of c is when the two public keys $pk_A^{(2)}$ and $pk_A^{(3)}$ are equal. However, the key-exchange security already guarantees that the probability that two independently generated public keys are equal is negligible. In fact, if this is non-negligible, then given an honestly generated public key pk_A , an adversary could independently generate (sk', pk') and obtain $pk' = pk_A$ with non-negligible probability. When this happens, key-exchange correctness allows the adversary to use sk' to compute the shared key associated with pk_A and any honest peer public key, contradicting key-exchange security. Thus Eve recovers b with overwhelming probability and breaks the security of the first execution.

It remains to prove undetectability. By key-exchange security, given all relevant public keys, including pk_E , the pair $(z, K_{AB}^{(1)})$ is computationally indistinguishable from two independent uniformly random keys of the same lengths. We may therefore replace only these two values by independent uniform keys before the next two executions; by efficient post-processing, this changes the complete observable transcript only computationally negligibly. In this hybrid, b and r are independent uniform bits, so $c = b \oplus r$ is a uniform bit independent of the two fresh key pairs generated for the next two executions. Since these two key pairs are independently and identically distributed, outputting them in increasing order when $c = 0$ and decreasing order when $c = 1$ gives exactly the same distribution as outputting two independently generated honest key pairs in their original order. The subsequent shared-key computations are therefore distributed exactly as in honest executions. Restoring the real values of z and $K_{AB}^{(1)}$ changes the transcript only computationally negligibly by key-exchange security. Hence the complete observable behavior of the device is computationally indistinguishable from that of an honest implementation.

Finally, only pk_E is stored in the device. If sk_E could be efficiently recovered from pk_E , then the shared key between Eve and any honest party could be efficiently computed from public information, contradicting key-exchange security. Thus reverse engineering the device reveals no secret of Eve. The construction is therefore a key-exchange SETUP. $\square$

When the shared key is sufficiently long, the First Law admits a stronger form.

Theorem 2 (Strong Form of the First Law). *A secure public-key key-exchange scheme whose shared-key length is at least the number of random bits required by one of its key-generation algorithms admits a key-exchange SETUP that reveals the full secret key of the corresponding party in all subsequent executions.*

Proof. Eve generates a legitimate key pair $(sk_E, pk_E) \leftarrow \text{KeyGen}_B$ and embeds pk_E into Alice's device. In the first execution, the device honestly generates $(sk_A^{(1)}, pk_A^{(1)}) \leftarrow \text{KeyGen}_A(1^\lambda)$ and outputs $pk_A^{(1)}$. In addition, it computes $z = \text{Shared}_A(sk_A^{(1)}, pk_A^{(1)}, pk_E)$. Eve observes $pk_A^{(1)}$ and computes the same value $z = \text{Shared}_B(sk_E, pk_A^{(1)}, pk_E)$ by key-exchange correctness. After receiving Bob's public key $pk_B^{(1)}$, the device computes the legitimate shared key $K_{AB}^{(1)} = \text{Shared}_A(sk_A^{(1)}, pk_A^{(1)}, pk_B^{(1)})$. By key-exchange security, z and $K_{AB}^{(1)}$ are jointly computationally indistinguishable from two independent uniformly random keys given all public keys. In particular, z remains computationally indistinguishable from uniform even given $K_{AB}^{(1)}$ and the entire public information. Let $\alpha(\lambda)$ be the number of random bits required by KeyGen_A . As usual, we view KeyGen_A as a deterministic algorithm once its uniformly random binary tape is fixed. By assumption of the theorem, the length of z is at least $\alpha(\lambda)$.

The device takes the first $\alpha(\lambda)$ bits of z as the random tape for KeyGen_A and computes $(sk_A^{(2)}, pk_A^{(2)}) = \text{KeyGen}_A(1^\lambda; z_1 \cdots z_{\alpha(\lambda)})$. The device may then erase z and retain only $(sk_A^{(2)}, pk_A^{(2)})$. In the second execution,

it outputs $pk_A^{(2)}$. Eve already knows z , so she reconstructs the same random tape, runs KeyGen_A herself, and recovers $sk_A^{(2)}$. After observing Bob's second public key, she can compute the corresponding shared key and break the security of the second execution.

The same procedure can be repeated. In each execution, the device computes the hidden shared key between Alice's current secret key and pk_E and uses its first $\alpha(\lambda)$ bits as the random tape for the next key generation. Eve computes the same hidden shared key from sk_E and Alice's current public key and therefore recovers every subsequently generated secret key.

It remains to prove undetectability over any polynomial number $T(\lambda)$ of executions. Define hybrids $H_0, \dots, H_{T(\lambda)-1}$, where H_j is obtained from the real execution by replacing the first j hidden shared keys used as random tapes for subsequent key generation by independent uniform strings of the same lengths. Consider the transition from H_{j-1} to H_j . In H_{j-1} , the random tape used to generate Alice's current key pair is uniform, so the current key pair has the honest distribution and is independent of the other honestly generated key pairs. By key-exchange security, the current hidden shared key remains computationally indistinguishable from a fresh uniform key even given the current legitimate shared key and all public keys. Since all later behavior of the device is efficiently computed from these values and fresh randomness, replacing the hidden shared key by a uniform key changes the complete observable transcript only computationally negligibly. Thus adjacent hybrids are computationally indistinguishable. In the final hybrid, every random tape used for subsequent key generation is uniform, so the resulting sequence of key pairs and shared keys has the honest distribution. A polynomial hybrid argument therefore shows that the complete observable transcript of the subverted device is computationally indistinguishable from that of an honest implementation.

Finally, the only Eve-specific value stored in the device is the public key pk_E . Eve's secret key sk_E never enters the device. Efficient recovery of sk_E from pk_E would contradict key-exchange security. Thus reverse engineering the device reveals no secret of Eve. The construction is therefore a key-exchange SETUP that recovers the corresponding party's subsequent secret keys. $\square$

Instantiation for CSIDH. We now instantiate the strong form of the First Law with the post-quantum key-exchange scheme CSIDH [14]. To the best of our knowledge, this is the first kleptographic subversion of CSIDH and, more generally, of isogeny-based commutative group-action key exchange. The construction, however, is not specific to CSIDH. CSIDH serves only as a concrete post-quantum realization of the universal law.

Let $p = 4 \prod_{i=1}^n \ell_i - 1$ be a prime with $\ell_1, \dots, \ell_n$ distinct small odd primes. CSIDH uses Montgomery curves $E_A : y^2 = x^3 + Ax^2 + x$ over $\mathbb{F}_p$, with base curve $E_0 : y^2 = x^3 + x$. Let π be the p -power Frobenius and $\mathcal{O} = \text{End}_{\mathbb{F}_p}(E_0) = \mathbb{Z}[\pi] \cong \mathbb{Z}[\sqrt{-p}]$ the associated quadratic order. Each ℓ_i splits in $\mathcal{O}$ as $(\ell_i) = \mathfrak{l}_i \bar{\mathfrak{l}}_i$, where $\mathfrak{l}_i = (\ell_i, \pi - 1)$ and $\bar{\mathfrak{l}}_i = (\ell_i, \pi + 1)$. Let $\mathcal{E} = \{[\mathfrak{a}] \star E_0 : [\mathfrak{a}] \in \text{cl}(\mathcal{O})\}$ be the CSIDH orbit of E_0 . The class group $\text{cl}(\mathcal{O})$ acts freely and transitively on $\mathcal{E}$. For $e \in \mathbb{Z}^n$, write $[e] = \prod_{i=1}^n [\mathfrak{l}_i]^{e_i}$. Then $[e][f] = [e + f]$ and $[e] \star ([f] \star E) = ([e][f]) \star E$ for all $e, f \in \mathbb{Z}^n$ and $E \in \mathcal{E}$.

The group-action inverse problem (GAIP) is to recover $[e]$ from $(E_0, [e] \star E_0)$, while the group-action computational Diffie–Hellman problem (GACDH) is to compute $([e][f]) \star E_0$ from $(E_0, [e] \star E_0, [f] \star E_0)$. In CSIDH, Alice and Bob independently sample secret vectors $e, f \in \{-m, \dots, m\}^n$ (where m is a small positive integer typically 5), exchange the public curves $[e] \star E_0$ and $[f] \star E_0$, and obtain the same shared curve $([e][f]) \star E_0$ by commutativity of the class-group action. Assume that the final key-derivation step makes the resulting CSIDH key exchange satisfy the joint pseudorandomness requirement of Definition 2. Let the output of the CSIDH shared-key algorithm be a $\beta(\lambda)$ -bit shared key. Let $\alpha(\lambda)$ be the number of random bits required by Alice's key-generation algorithm, and suppose that $\beta(\lambda) \geq \alpha(\lambda)$.²

²This condition can be satisfied naturally for CSIDH-512. For the original CSIDH-512 parameter set, KeyGen samples a secret exponent vector in $\{-5, \dots, 5\}^{74}$, so the intended secret-key distribution has size $11^{74} \approx 2^{256}$. A fixed-length implementation may take a 512-bit uniform random tape and deterministically map it to this distribution with statistical distance at most about 2^{-256} ; the resulting exponent vector is the secret key. We take this negligibly perturbed seeded implementation as the honest KeyGen in the instantiation; its secret-key distribution is negligibly close to the conventional uniform distribution on $\{-5, \dots, 5\}^{74}$. The raw shared curve is represented by its Montgomery coefficient in $\mathbb{F}_p$ for a 511-bit prime p , and is therefore represented in about 512 bits. A final key-derivation function may output a 512-bit pseudorandom shared key, which can then be used directly as the 512-bit random tape for the next invocation of KeyGen . Note that the randomness of any randomized key-generation algorithm may be represented by a uniform binary tape followed by a deterministic transformation to the required distribution; when exact fixed-length sampling is impossible, a sufficiently long fixed tape gives negligible statistical error.

Let Eve generate a legitimate CSIDH key pair $(x, E_x = [x] \star E_0)$ and embed the public curve E_x into Alice's device. In the first execution, the device honestly generates Alice's key pair $(e^{(1)}, [e^{(1)}] \star E_0) \leftarrow \text{KeyGen}_A(1^\lambda; r^{(1)})$ and outputs the public curve $[e^{(1)}] \star E_0$. In addition, it computes the hidden shared curve $z^{(1)} = [e^{(1)}] \star E_x$. By commutativity of the class-group action, $z^{(1)} = [e^{(1)}] \star ([x] \star E_0) = [x] \star ([e^{(1)}] \star E_0)$. The device then obtains the corresponding hidden shared key $K_{DE}^{(1)} = \text{Shared}_A(e^{(1)}, [e^{(1)}] \star E_0, E_x)$, where the final key-derivation step is part of the legitimate CSIDH key exchange.

Eve, after observing the public curve $[e^{(1)}] \star E_0$, computes the same hidden shared curve as $z^{(1)} = [x] \star ([e^{(1)}] \star E_0)$ and hence obtains the same hidden shared key $K_{DE}^{(1)} = \text{Shared}_B(x, [e^{(1)}] \star E_0, E_x)$ by correctness of the key exchange. After receiving Bob's first public curve $[f^{(1)}] \star E_0$, the device also computes the legitimate shared key $K_{AB}^{(1)} = \text{Shared}_A(e^{(1)}, [e^{(1)}] \star E_0, [f^{(1)}] \star E_0)$. By the joint pseudorandomness assumption, $K_{DE}^{(1)}$ and $K_{AB}^{(1)}$ are jointly computationally indistinguishable from two independent uniform $\beta(\lambda)$ -bit strings given all public curves. In particular, $K_{DE}^{(1)}$ remains computationally indistinguishable from uniform even given the complete observable first execution. Since $\beta(\lambda) \geq \alpha(\lambda)$, the device uses the first $\alpha(\lambda)$ bits of $K_{DE}^{(1)}$ as the random tape $r^{(2)}$ for the next key generation and computes $(e^{(2)}, [e^{(2)}] \star E_0) = \text{KeyGen}_A(1^\lambda; r^{(2)})$. It may then erase $z^{(1)}$, $K_{DE}^{(1)}$, and $r^{(2)}$, retaining only the state required for the second CSIDH execution.

In the second execution, the device outputs the previously generated public curve $[e^{(2)}] \star E_0$. Eve already knows $K_{DE}^{(1)}$ and therefore reconstructs the same random tape $r^{(2)}$. By running $\text{KeyGen}_A(1^\lambda; r^{(2)})$ herself, she recovers Alice's entire second-session secret key $e^{(2)}$. Consequently, after observing Bob's second public curve $[f^{(2)}] \star E_0$, Eve can compute the second-session shared curve $[e^{(2)}] \star ([f^{(2)}] \star E_0)$ and hence the corresponding shared key. The same construction may be iterated, allowing Eve to recover every subsequently generated secret key.

Undetectability follows by the same sequential-hybrid argument as in Theorem 2. The first public curve $[e^{(1)}] \star E_0$ is distributed exactly as in an honest CSIDH execution. Replacing the first hidden shared key $K_{DE}^{(1)}$ by a uniform $\beta(\lambda)$ -bit string makes $r^{(2)}$ an honest uniform random tape for KeyGen_A . Inductively, after the preceding hidden shared keys have been replaced by uniform strings, the current CSIDH key pair has the honest distribution, so joint key-exchange security permits replacement of the current hidden shared key by an independent uniform string. Repeating this replacement over any polynomial number of executions yields a final hybrid with the honest transcript distribution. Hence the complete observable transcript of the subverted CSIDH device is computationally indistinguishable from that of an honest implementation.

Finally, the only information belonging to Eve that resides in the device is the public curve $E_x = [x] \star E_0$; Eve's secret key x never enters the device. Recovering x from (E_0, E_x) implies recovering $[x]$ from (E_0, E_x) , which is GAIP and is infeasible under the security assumption underlying the scheme itself. Thus even complete reverse engineering of the device reveals no secret of Eve. The construction is therefore a kleptographic SETUP that realises the secret-key recovery guaranteed by the Strong Form of the First Law.

3. LAW OF PUBLIC-KEY ENCRYPTION

We now turn to public-key encryption for the second law.

Definition 5 (PKE). *A public-key encryption scheme consists of three algorithms KeyGen , Enc , and Dec and a ciphertext space $\mathcal{C}_\lambda$. Key generation produces $(sk, pk) \leftarrow \text{KeyGen}(1^\lambda)$. For an admissible message m , randomized encryption produces a ciphertext $c \leftarrow \text{Enc}(pk, m) \in \mathcal{C}_\lambda$, and decryption satisfies $\text{Dec}(sk, c) = m$ except with negligible probability.*

Definition 6 (PKE-Security). *A public-key encryption scheme is secure if, for every probabilistic polynomial-time algorithm $\mathcal{A}$ that outputs an admissible message $m \leftarrow \mathcal{A}(pk)$, the distributions $(pk, m, \text{Enc}(pk, m))$ and $(pk, m, U_{n(\lambda)})$ are computationally indistinguishable, where $(sk, pk) \leftarrow \text{KeyGen}(1^\lambda)$, $U_{n(\lambda)}$ is uniform over $\{0, 1\}^{n(\lambda)}$, and $n(\lambda)$ is the length of the binary representation of a ciphertext.*

This definition of security is natural: Since a ciphertext is publicly observable, a strong form of hiding is that the ciphertext itself reveals no efficiently detectable structure beyond its length. In the simplest case $\mathcal{C}_\lambda = \{0, 1\}^{n(\lambda)}$, this means that an honest ciphertext is computationally indistinguishable from a uniformly random string of the same length. This requirement has appeared previously in the literature as part of IND\$-CPA security [7], where ciphertexts are required to be indistinguishable from uniformly random bit strings of the same length.

Definition 7 (Black-Box Encryption Execution). *In a black-box encryption execution, the device receives an honestly generated recipient public key pk and an admissible plaintext m , runs $c \leftarrow \text{Enc}(pk, m)$, and outputs c . The user observes only pk , m , and c . The internal randomness and the internal state of the device are hidden. Over multiple executions, the observable transcript consists only of the corresponding sequence of public keys, plaintexts, and ciphertexts.*

Definition 8 (PKE-SETUP). *A public-key encryption $SETUP$ is a stateful subverted implementation of a public-key encryption scheme satisfying the following three properties:*

- (1) *it breaks the security of the scheme for the kleptographer;*
- (2) *over any polynomial number of black-box executions, its observable transcript is computationally indistinguishable from that of an honest implementation;*
- (3) *no secret of the kleptographer is revealed even if the device is fully reverse-engineered.*

A public-key encryption scheme is said to be kleptographically insecure if it admits a public-key encryption $SETUP$. It is said to be kleptographically secure if it is not kleptographically insecure.

The following theorem shows that every secure public-key encryption scheme is kleptographically insecure.

Theorem 3 (Second Law of Public-Key Cryptography). *A public-key encryption scheme is secure³ only if it is kleptographically insecure.*

Proof. Eve generates an independent legitimate key pair $(sk_E, pk_E) \leftarrow \text{KeyGen}(1^\lambda)$ and embeds pk_E into the encryption device. Consider a target execution in which the device is asked to encrypt a message $m^{(1)}$ under an honestly generated recipient public key $pk^{(1)}$. The device behaves honestly and outputs $c^{(1)} \leftarrow \text{Enc}(pk^{(1)}, m^{(1)})$. Internally, it also computes a carrier $D \leftarrow \text{Enc}(pk_E, m^{(1)})$ and writes $D = D_1 D_2 \cdots D_N$, where $N = n(\lambda)$.

Now let $\pi : \{0, 1\}^{n(\lambda)} \rightarrow \{0, 1\}$ return the first bit of a ciphertext. For $b \in \{0, 1\}$, let $\text{RS}(pk, m, b)$ be a rejection-sampling algorithm which repeatedly samples $c \leftarrow \text{Enc}(pk, m)$ until $\pi(c) = b$, and outputs the first accepted ciphertext. To obtain a strict polynomial-time algorithm, the sampler stops after λ trials and outputs one final honest encryption if no ciphertext has been accepted.

During the next N encryption executions, suppose the i -th request is to encrypt an admissible message $m^{(i+1)}$ under an honestly generated recipient public key $pk^{(i+1)}$. The device outputs $c^{(i+1)} \leftarrow \text{RS}(pk^{(i+1)}, m^{(i+1)}, D_i)$. Every accepted ciphertext is an output of the honest encryption algorithm on the requested message, so correctness is preserved.

Eve observes the ciphertexts and reads $D_i = \pi(c^{(i+1)})$ for each i , except with negligible probability caused by the polynomial-time cutoff. After collecting all N bits, she reconstructs D and computes $m^{(1)} = \text{Dec}(sk_E, D)$. Thus Eve recovers the complete plaintext of the target encryption.

It remains to prove undetectability. First replace the real carrier D by an independent uniform string U_N . By pseudorandomness of the ciphertexts, $(pk_E, m^{(1)}, D)$ is computationally indistinguishable from $(pk_E, m^{(1)}, U_N)$. Thus this replacement changes the observable transcript only computationally negligibly.

Now consider one later encryption request, and let $X = (pk, m)$ denote its public key and message. Let $C \leftarrow \text{Enc}(pk, m)$, write $p_b(X) = \Pr[\pi(C) = b]$, and let $\delta(X) = |p_0(X) - 1/2|$. Ciphertext pseudorandomness implies $\mathbb{E}[\delta(X)] \leq \text{negl}(\lambda)$, where the expectation is over the distribution of X . Otherwise, since Enc is public, a probabilistic polynomial-time distinguisher could use polynomially many fresh encryptions under (pk, m) to estimate the more likely value of $\pi(C)$ and distinguish an honest challenge ciphertext from a uniformly random string with non-negligible advantage.

By Markov's inequality, $\Pr[\delta(X) > 1/4] \leq 4\mathbb{E}[\delta(X)]$, which is negligible. Outside this event, both $p_0(X)$ and $p_1(X)$ are at least $1/4$. Conditioned on such an X , let B be a uniform bit. The uncapped rejection sampler satisfies $\Pr[\text{RS}(pk, m, B) = c] = \Pr[C = c] / (2p_{\pi(c)}(X))$, and its total variation distance from the honest ciphertext distribution is exactly $\delta(X)$. Averaging over X , and accounting for the negligible bad event, this distance is negligible. Moreover, outside the bad event the probability that the sampler reaches the cutoff after λ trials is at most $(3/4)^\lambda$. Hence the cutoff probability is negligible.

When the carrier is uniform, its bits are independent uniform bits. Hence each later ciphertext is negligibly close to an honest ciphertext. A sequential hybrid argument extends this conclusion to any polynomial number of executions. The argument also covers adaptively chosen admissible messages, since at each hybrid step the distinguisher can efficiently simulate the preceding honest transcript before generating the next message.

³In the sense of Definition 6.

Replacing the uniform carrier by the real carrier D changes the transcript only computationally negligibly. Therefore the complete observable behavior of the subverted device is computationally indistinguishable from that of an honest implementation.

Finally, the only Eve-specific value stored in the device is pk_E . Eve's secret key sk_E never enters the device. Pseudorandom ciphertexts imply confidentiality, so efficient recovery of sk_E from pk_E would contradict the security of the encryption scheme. Thus reverse engineering the device reveals no secret of Eve. The construction is therefore a public-key encryption SETUP with complete plaintext recovery. $\square$

The Second Law above treats the cleanest case in which ciphertexts are pseudorandom binary strings. The phenomenon is more general. A ciphertext need not be naturally viewed as a binary string. It may instead belong to an arbitrary ciphertext space, and its pseudorandomness may be defined relative to the uniform distribution on that space. When the ciphertext space is a polynomial-length product over a polynomial-size alphabet, the same argument applies symbol by symbol.

Definition 9 (Ciphertext-Space Pseudorandomness). *Let a public-key encryption scheme have ciphertext space $\mathcal{C}_\lambda$. The scheme has pseudorandom ciphertexts over $\mathcal{C}_\lambda$ if, for every probabilistic polynomial-time algorithm $\mathcal{A}$ that outputs an admissible message $m \leftarrow \mathcal{A}(pk)$, the distributions $(pk, m, \text{Enc}(pk, m))$ and $(pk, m, U_{\mathcal{C}_\lambda})$ are computationally indistinguishable, where $(sk, pk) \leftarrow \text{KeyGen}(1^\lambda)$ and $U_{\mathcal{C}_\lambda}$ is uniform over $\mathcal{C}_\lambda$.*

Theorem 4 (General Form of the Second Law). *Let a public-key encryption scheme have ciphertext space $\mathcal{C}_\lambda = \Sigma_\lambda^{N(\lambda)}$, where Σ_λ is a public efficiently enumerable alphabet and both $|\Sigma_\lambda|$ and $N(\lambda)$ are polynomially bounded. If its ciphertexts are pseudorandom over $\mathcal{C}_\lambda$, then the scheme is kleptographically insecure.*

Proof. Let $\mathcal{C}_\lambda = \Sigma_\lambda^{N(\lambda)}$ and let $q_\lambda = |\Sigma_\lambda|$. Eve generates an independent legitimate key pair $(sk_E, pk_E) \leftarrow \text{KeyGen}(1^\lambda)$ and embeds pk_E into the encryption device.

Consider a target execution in which the device is asked to encrypt a message $m^{(1)}$ under an honestly generated public key $pk^{(1)}$. The device behaves honestly and outputs $c^{(1)} \leftarrow \text{Enc}(pk^{(1)}, m^{(1)})$. Internally, it also computes a carrier $D \leftarrow \text{Enc}(pk_E, m^{(1)})$. Write $D = (D_1, \dots, D_N)$, where $N = N(\lambda)$ and each $D_i \in \Sigma_\lambda$.

Now let $\pi : \mathcal{C}_\lambda \rightarrow \Sigma_\lambda$ return the first coordinate of a ciphertext. For $a \in \Sigma_\lambda$, let $\text{RS}(pk, m, a)$ repeatedly sample $c \leftarrow \text{Enc}(pk, m)$ until $\pi(c) = a$, and output the first accepted ciphertext. To obtain a strict polynomial-time algorithm, the sampler stops after λq_λ trials and outputs one final honest encryption if no ciphertext has been accepted.

During the next N encryption executions, suppose the i -th request is to encrypt an admissible message $m^{(i+1)}$ under an honestly generated public key $pk^{(i+1)}$. The device outputs $c^{(i+1)} \leftarrow \text{RS}(pk^{(i+1)}, m^{(i+1)}, D_i)$. Every accepted ciphertext is an output of the honest encryption algorithm on the requested message, so correctness is preserved.

Eve observes the ciphertexts and reads $D_i = \pi(c^{(i+1)})$ for every i , except when the corresponding rejection sampler reaches its cutoff. After recovering all N symbols, she reconstructs D and computes $m^{(1)} = \text{Dec}(sk_E, D)$. Thus Eve recovers the complete plaintext of the target encryption except with the probability of a cutoff.

It remains to prove undetectability and that the cutoff probability is negligible. First replace the real carrier D by an independent uniform element $U_{\mathcal{C}_\lambda}$. By ciphertext-space pseudorandomness, $(pk_E, m^{(1)}, D)$ and $(pk_E, m^{(1)}, U_{\mathcal{C}_\lambda})$ are computationally indistinguishable. Since the remaining behavior of the device is efficiently computed from the carrier, this replacement changes the observable transcript only computationally negligibly.

Now consider one later encryption request, and let $X = (pk, m)$ denote its public key and message. Let $C \leftarrow \text{Enc}(pk, m)$. For $a \in \Sigma_\lambda$, write $p_a(X) = \Pr[\pi(C) = a]$, and define $\delta(X) = \frac{1}{2} \sum_{a \in \Sigma_\lambda} |p_a(X) - 1/q_\lambda|$. Since Σ_λ is public, efficiently enumerable, and polynomial in size, ciphertext-space pseudorandomness implies $\mathbb{E}[\delta(X)] \leq \text{negl}(\lambda)$. Otherwise, a probabilistic polynomial-time distinguisher could use polynomially many fresh encryptions under (pk, m) to estimate the probabilities $p_a(X)$ to sufficient inverse-polynomial accuracy and distinguish the first coordinate of an honest challenge ciphertext from a uniform element of Σ_λ with non-negligible advantage.

By Markov's inequality, $\Pr[\delta(X) > 1/(4q_\lambda)] \leq 4q_\lambda \mathbb{E}[\delta(X)]$, which is negligible because q_λ is polynomially bounded. Outside this event, for every $a \in \Sigma_\lambda$ we have $|p_a(X) - 1/q_\lambda| \leq 2\delta(X) \leq 1/(2q_\lambda)$ and hence $p_a(X) \geq 1/(2q_\lambda)$. Conditioned on such an X , let B be uniform over Σ_λ . The uncapped rejection sampler satisfies $\Pr[\text{RS}(pk, m, B) = c] = \Pr[C = c]/(q_\lambda p_{\pi(c)}(X))$, and its total variation distance from the honest

ciphertext distribution is exactly $\delta(X)$. Averaging over X , and accounting for the negligible bad event, this distance is negligible. Moreover, outside the bad event the probability that the rejection sampler fails to accept within λq_λ trials is at most $(1 - 1/(2q_\lambda))^{\lambda q_\lambda} \leq e^{-\lambda/2}$. Since $N(\lambda)$ is polynomially bounded, the probability that any of the $N(\lambda)$ samplers reaches its cutoff is negligible.

In the uniform-carrier hybrid, $D_1, \dots, D_N$ are independent uniform elements of Σ_λ . Therefore every subsequent rejection-sampled ciphertext is negligibly close to an honest ciphertext. A sequential hybrid over the polynomial number of executions shows that the complete observable transcript is computationally indistinguishable from an honest transcript. As above, adaptive admissible messages are covered because each hybrid distinguisher can efficiently simulate the preceding honest transcript. Restoring the real carrier D changes the transcript only computationally negligibly. Hence the complete observable behavior of the subverted device is computationally indistinguishable from that of an honest implementation.

Finally, the only Eve-specific value stored in the device is pk_E . Eve's secret key sk_E never enters the device. Efficient recovery of sk_E from pk_E would contradict the confidentiality implied by ciphertext-space pseudorandomness. Thus reverse engineering the device reveals no secret of Eve. The construction is therefore a public-key encryption SETUP with complete plaintext recovery. $\square$

Instantiation for Regev encryption. We now instantiate the general form of the Second Law with the Regev public-key encryption scheme [15]. Scheme-specific subversions of Regev encryption have been studied previously [16]. In particular, the previous construction exploits the freedom in choosing encryption randomness to embed a backdoor into the specific LWE encryption scheme. Our construction is different in nature. It uses no LWE-specific structure and requires no additional property beyond the ciphertext pseudorandomness assumed by the general form of the Second Law. The subversion is therefore already guaranteed by the universal law; Regev encryption serves here only as a concrete post-quantum instantiation. This example shows that the law applies naturally to ciphertext spaces that are not binary.

Let n be the security parameter and let $p \leq \text{poly}(n)$ be prime. Let $M \geq (1 + \varepsilon)(n + 1) \log_2 p$ for a constant $\varepsilon > 0$, and let χ be an error distribution for which Regev decryption is correct with overwhelming probability. The secret key is a uniformly random vector $s \leftarrow \mathbb{Z}_p^n$. The public key consists of M independent LWE samples (a_i, b_i) , where $a_i \leftarrow \mathbb{Z}_p^n$, $e_i \leftarrow \chi$, and $b_i = \langle a_i, s \rangle + e_i \pmod{p}$.

To encrypt a bit $\mu \in \{0, 1\}$, choose a uniformly random subset $S \subseteq \{1, \dots, M\}$ and output the ciphertext $c = (a, b)$, where $a = \sum_{i \in S} a_i$ and $b = \sum_{i \in S} b_i + \mu \lfloor p/2 \rfloor \pmod{p}$. Decryption computes $b - \langle a, s \rangle \pmod{p}$ and decides whether the result is closer to 0 or to $\lfloor p/2 \rfloor$. Thus the natural ciphertext space is $\mathcal{C}_n = \mathbb{Z}_p^n \times \mathbb{Z}_p = \mathbb{Z}_p^{n+1}$.

We claim that Regev ciphertexts are pseudorandom over $\mathcal{C}_n$ under the decisional LWE assumption. First replace the LWE public key $(a_i, b_i)_{i=1}^M$ by M independent uniform elements of $\mathbb{Z}_p^{n+1}$. By decisional LWE, this changes the joint distribution of the public key and ciphertext only computationally negligibly.

In this hybrid, an encryption of 0 is the sum of a uniformly random subset of M independent uniform elements of the group $\mathbb{Z}_p^{n+1}$. By the leftover hash lemma [17], its statistical distance from the uniform distribution on $\mathbb{Z}_p^{n+1}$ is negligible when $M \geq (1 + \varepsilon)(n + 1) \log_2 p$. An encryption of 1 differs only by adding the fixed element $(0, \dots, 0, \lfloor p/2 \rfloor)$, which is a permutation of $\mathbb{Z}_p^{n+1}$ and therefore preserves the uniform distribution. Hence, for either message, the ciphertext in this hybrid is statistically close to uniform over $\mathcal{C}_n$.

By decisional LWE once more, we may restore the real LWE public key while keeping the ciphertext uniform, changing the joint distribution only computationally negligibly. Therefore, for every probabilistic polynomial-time algorithm $\mathcal{A}$ that outputs a message $\mu \leftarrow \mathcal{A}(pk)$, the distributions $(pk, \mu, \text{Enc}(pk, \mu))$ and $(pk, \mu, U_{\mathcal{C}_n})$ are computationally indistinguishable. Regev encryption therefore has pseudorandom ciphertexts over its natural ciphertext space.

Finally, $\mathcal{C}_n = \Sigma_n^{N(n)}$ with $\Sigma_n = \mathbb{Z}_p$ and $N(n) = n + 1$. Since p and n are polynomially bounded, Regev encryption satisfies the hypotheses of the general form of the Second Law. It is therefore kleptographically insecure.

REFERENCES

- [1] W. Diffie and M. E. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, 22(6), 1976, pp. 644–654.
- [2] A. Young and M. Yung, *The Dark Side of “Black-Box” Cryptography, or: Should We Trust Capstone?*, Advances in Cryptology—CRYPTO ’96, LNCS 1109, Springer, 1996, pp. 89–103.
- [3] A. Young and M. Yung, *Kleptography: Using Cryptography Against Cryptography*, Advances in Cryptology—EUROCRYPT ’97, LNCS 1233, Springer, 1997, pp. 62–74.

- [4] M. Bellare, K. G. Paterson, and P. Rogaway, *Security of Symmetric Encryption against Mass Surveillance*, Advances in Cryptology—CRYPTO 2014, Part I, LNCS 8616, Springer, 2014, pp. 1–19.
- [5] M. Bellare, J. Jaeger, and D. Kane, *Mass-Surveillance without the State: Strongly Undetectable Algorithm-Substitution Attacks*, Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security—CCS 2015, ACM, 2015, pp. 1431–1440.
- [6] H. Jiang, J. Han, Z. Zhang, Z. Ma, and H. Wang, *Practical Algorithm Substitution Attacks on Real-World Public-Key Cryptosystems*, IEEE Transactions on Information Forensics and Security, 18, 2023, pp. 5069–5081.
- [7] L. von Ahn and N. J. Hopper, *Public-Key Steganography*, Advances in Cryptology—EUROCRYPT 2004, LNCS 3027, Springer, 2004, pp. 323–341.
- [8] G. Persiano, D. H. Phan, and M. Yung, *Anamorphic Encryption: Private Communication Against a Dictator*, Advances in Cryptology—EUROCRYPT 2022, Part II, LNCS 13276, Springer, 2022, pp. 34–63.
- [9] M. Kutyłowski, G. Persiano, D. H. Phan, M. Yung, and M. Zawada, *The Self-Anti-Censorship Nature of Encryption: On the Prevalence of Anamorphic Cryptography*, Proceedings on Privacy Enhancing Technologies, 2023(4), 2023, pp. 170–183.
- [10] Y. Wang, R. Chen, X. Huang, and M. Yung, *Sender-Anamorphic Encryption Reformulated: Achieving Robust and Generic Constructions*, Advances in Cryptology—ASIACRYPT 2023, Part VI, LNCS 14443, Springer, 2023, pp. 135–167.
- [11] M. Ringerud, *Note on Subversion-Resilient Key Exchange*, IACR Cryptology ePrint Archive, Report 2023/749, 2023.
- [12] Y. Dodis and E. Goldin, *Anamorphic-Resistant Encryption; Or Why the Encryption Debate is Still Alive*, Advances in Cryptology—CRYPTO 2025, Part III, LNCS 16002, Springer, 2025, pp. 440–471.
- [13] E. S. V. Freire, D. Hofheinz, E. Kiltz, and K. G. Paterson, *Non-Interactive Key Exchange*, Public-Key Cryptography—PKC 2013, LNCS 7778, Springer, 2013, pp. 254–271.
- [14] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes, *CSIDH: An Efficient Post-Quantum Commutative Group Action*, Advances in Cryptology—ASIACRYPT 2018, Part III, LNCS 11274, Springer, 2018, pp. 395–427.
- [15] O. Regev, *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*, Journal of the ACM, 56(6), 2009, Article 34, pp. 34:1–34:40.
- [16] Z. Yang, R. Chen, C. Li, L. Qu, and G. Yang, *On the Security of LWE Cryptosystem against Subversion Attacks*, The Computer Journal, 63(4), 2020, pp. 495–507.
- [17] R. Impagliazzo, L. A. Levin, and M. Luby, *Pseudo-Random Generation from One-Way Functions*, Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, ACM, 1989, pp. 12–24.

THE UNIVERSITY OF MANCHESTER, MANCHESTER, UK
 Email address: `trey.li@manchester.ac.uk`