Backdooring Post-Quantum Cryptography: Kleptographic Attacks on Lattice-based KEMs

Prasanna Ravi[1,2], Shivam Bhasin[1], Anupam Chattopadhyay[1,2], Aikata[3], and Sujoy Sinha Roy[3]

[1]*Temasek Labs, Nanyang Technological University, Singapore*
[2]*School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[3]*IAIK, Graz University of Technology, Graz, Austria*

***Abstract*—Post-quantum Cryptography (PQC) has reached the verge of standardization competition, with Kyber as a winning candidate. In this work, we demonstrate practical backdoor insertion in Kyber through kleptrography. The backdoor can be inserted using classical techniques like ECDH or post-quantum Classic Mceliece. The inserted backdoor targets the key generation procedure where generated output public keys subliminally leak information about the secret key to the owner of the backdoor. We demonstrate first practical instantiations of such attack at the protocol level by validating it on TLS 1.3.**

***Index Terms*—Post Quantum Cryptography, Kyber, Kleptography, backdoor**

I. Introduction

The standardization of Post-Quantum Cryptography (PQC), driven by NIST, has hit a milestone with recent winner announcement [1]. This will trigger immediate efforts to develop in-house and third party IPs and products for PQC algorithms, targeting variety of platforms. While the winners have been thoroughly scrutinized from the standpoint of theoretical post-quantum security guarantees, implementation performance and susceptibility to side-channel attacks, the threat of *backdoors* has received very little attention.

In the early 1990s, Young and Yung published a series of works on new types of (backdoor) attacks referred to as *kleptographic attacks*, targeting implementations of public-key cryptography in a black-box setting [2], [3]. For instance, security products such as Hardware Security Modules (HSMs), Trusted Platform Modules (TPMs) and smart cards, claim to have almost perfect security of cryptographic keys against external access, while also obfuscating the type of cryptograhpic operations performed within the device. This leaves a certain element of uncertainty for the user, since there is no way to verify if the *black-box* module is doing exactly what it claims. In this scenario, it is possible that the designer of the black-box device can insert a backdoor to compromise its security, thereby impacting the user of the device and/or anyone who is communicating with the device.

The setup of a kleptographic attack involves a *backdoored* black-box implementation denoted as T. The backdoor is implemented using another public-key encryption scheme denoted as A in such a way that, the secret key of T (i.e.) $\mathsf{sk_T}$ is encrypted using the public key of A (i.e.) $\mathsf{pk_A}$. The encrypted result is subsequently hidden within the output of T (i.e.) $\mathsf{out_T}$. Only the backdoor owner who has the secret key $\mathsf{sk_A}$ should be able to recover $\mathsf{sk_T}$ from $\mathsf{out_T}$, and no one else. Reverse engineering the device may detect the backdoor, but this knowledge can neither be used to decrypt leaks of the past nor information that would leak in the future.

Though the topic of kleptography has been known since the 1990s, it did not receive much attention until the Snowden revelations in 2013, which provided very disturbing information about *subversion of standards* by NSA [4]. In particular, several news articles and reports have strongly suggested the possible backdoor-ing of the elliptic-curve random-number generator DualEC [5], which is very closely related to the "repeated DH setup" attack of Young and Yung. There exist several works which have reported kleptographic attacks on different types of classical public-key cryptographic schemes such as RSA, finite field DH, and elliptic-curve cryptography (ECC) [6], [7]. However, kleptographic attacks on post-quantum schemes has received lesser attention, and in particular we are not aware of any prior work over the leading candidates and winners of the NIST standardization process[1].

In this work, we propose the following novel contributions:

1) We present the first practical kleptographic attack on Kyber KEM. We propose novel approaches to construct backdoors within the key generation procedure of Kyber KEM, through exploitation of inherent properties of the LWE problem.
2) Unlike previous techniques to backdoor the key-generation procedure, our proposed technique allows use of both classical (pre-quantum) cryptograhpic schemes like ECC as well as post-quantum cryptograhpic schemes with much larger ciphertexts.
3) We experimentally validate our attacks by instantiating two types of backdoors for Kyber KEM: (1) Pre-Quantum based backdoor using Elliptic Curve Diffie Hellman (ECDH) scheme and (2) Post-Quantum based backdoor using Classic Mceliece KEM. We validated our attacks on the software implementation of Kyber KEM, running on the Intel x86-64 machine. We positively confirm that our attack works with a 100% success

[1]The work of Hemmert et al. [8] on backdoor-ing LWE cryptosystem is recently and independently developed, published online during preparation of our manuscript

rate with both the backdoors. Moreover, the generated backdoor-ed public keys remain valid for key exchange with an overwhelming probability.

4) We also integrate our proposed backdoor-ed Kyber KEM, within the open-source implementation of TLS 1.3 protocol, to present the first practical instantiation of a backdoor-ed PQC-based TLS 1.3 protocol, thereby demonstrating the capability of our proposed kleptographic attack at the protocol level.

Availability of software: For scrutiny and reproducibility, we have made the backdoor-ed implementation of Kyber KEM available at https://github.com/PRASANNA-RAVI/Klepto_on_Kyber.

II. Background

A. Notations

We denote the ring of integers modulo a prime q as $\mathbb{Z}_q$. The polynomial ring $\mathbb{Z}_q(x)/\phi(x)$ is denoted as R_q. We denote $\mathbf{r} \in R_q^{k\times\ell}$ as a *module* of dimension $k \times \ell$. Polynomials in R_q and modules in $R_q^{k\times\ell}$ are denoted in bold lower case letters. The i^{th} coefficient of a polynomial $\mathbf{A} \in R_q$ is denoted as $\mathbf{A}[i]$. Product of polynomials $\mathbf{a}$ and $\mathbf{b}$ in the ring R_q is denoted as $\mathbf{c} = \mathbf{a} \times \mathbf{b}$, while coefficient-wise multiplication is denoted using the symbol $\circ$. Kyber utilizes the well-known Number Theoretic Transform (NTT) for polynomial multiplication. The output of NTT over a polynomial $\mathbf{a} \in R_q$ is denoted as $\hat{\mathbf{a}}$. The product $\mathbf{c} = \mathbf{a} \times \mathbf{b}$ using NTT is computed as $\mathbf{c} = \mathsf{INTT}(\mathsf{NTT}(\mathbf{a}) \circ \mathsf{NTT}(\mathbf{b}))$. Byte arrays of length n are denoted as $\mathcal{B}^n$. The i^{th} bit in an element $x \in \mathbb{Z}_q$ is denoted as x_i.

B. Kyber KEM

Kyber is a chosen-ciphertext secure (CCA-secure) KEM based on the Module-Learning With Errors (MLWE) problem, that has been selected for standardization of PQC based KEMs, owing to its strong theoretical security guarantees and implementation performance [9]. The scheme derives its security from the hardness of the MLWE problem. The search MLWE problem requires the attacker to solve for $(\mathbf{s}, \mathbf{e}) \in R_q^k$ given polynomially many LWE instances of the form $(\mathbf{a}, \mathbf{t} = \mathbf{a}\cdot\mathbf{s}+\mathbf{e}) \in (R_q^{k\times k} \times R_q^k)$, where coefficients of $\mathbf{a}$ are uniformly in random in the range $[0, q]$, while coefficients of $\mathbf{s}$ and $\mathbf{e}$ are sampled from a smaller range $[-\eta, \eta]$ based on a Centered Binomial Distribution (CBD) with $\eta << q$.

1) Algorithmic Description: Kyber offers three security levels: Kyber512 (NIST Security Level 1), Kyber768 (Level 3) and Kyber1024 (Level 5) with $k = 2, 3$ and 4 respectively. It operates over the anti-cyclic polynomial ring R_q with a prime modulus $q = 3329$ and degree $n = 256$. The CCA-secure Kyber contains in its core, a chosen-plaintext secure encryption scheme of Kyber (i.e.) IND-CPA secure Kyber PKE scheme. Refer to Algorithm 1 for a simplified description of the key-generation procedure of Kyber KEM, as it forms the main focus of our work. The function Sample_U denotes sampling from a uniform distribution; the function Expand inflates a small seed into a uniformly random matrix in $R_q^{k\times k}$; and the function Sample_B uses a short seed to sample coefficients from the Centered Binomial Distribution (CBD) in $[-\eta, \eta]$ respectively.

Algorithm 1 Key Generation Procedure of IND-CPA secure Kyber PKE scheme

1: **procedure** KeyGen
2: $seed_A \in \mathcal{B}^{32} \leftarrow \mathsf{Sample}_U()$
3: $seed_B \in \mathcal{B}^{32} \leftarrow \mathsf{Sample}_U()$
4: $\hat{\mathbf{a}} \in R_q^{k\times k} \leftarrow \mathsf{Expand}(seed_A)$ ▷ Sample $\hat{\mathbf{a}}$
5: $\mathbf{s} \in R_q^k \leftarrow \mathsf{Sample}_B(seed_B, coin_s)$ ▷ Sample $\mathbf{s}$
6: $\mathbf{e} \in R_q^k \leftarrow \mathsf{Sample}_B(seed_B, coin_e)$ ▷ Sample $\mathbf{e}$
7: $\hat{\mathbf{t}} = \hat{\mathbf{a}} \circ \mathsf{NTT}(\mathbf{s}) + \mathsf{NTT}(\mathbf{e})$ ▷ Compute $\mathsf{NTT}(\mathbf{t})$
8: Return $(pk = (seed_A, \hat{\mathbf{t}}), sk = (\mathsf{NTT}(\mathbf{s}))$
9: **end procedure**

Referring to Alg.1, the key-generation procedure (KeyGen) of Kyber PKE simply involves computation of an MLWE instance. It starts by sampling random seeds $seed_A \in \mathcal{B}^{32}$ and $seed_B \in \mathcal{B}^{32}$ (Lines 2-3). The module $\hat{\mathbf{a}} \in R_q^{k\times k}$ is sampled from $seed_A$, while the secret $\mathbf{s}$ and error $\mathbf{e}$ are sampled from $seed_B$ based on CBD distribution (Lines 5-6). The MLWE instance $\mathbf{t} \in R_q^k = (\mathbf{a} \cdot \mathbf{s} + \mathbf{e})$ is computed (Line 7), and the tuple $(seed_A, \mathbf{t})$ forms the public key, while $\mathsf{NTT}(\mathbf{s})$ (i.e.) $\hat{\mathbf{s}}$ forms the secret key (Line 8).

We do not discuss the encryption and decryption procedures of the Kyber PKE scheme, and refer the reader to [10] for more details. In the context of our work, it is important to note that Kyber PKE scheme works on noisy inputs and outputs, unlike symmetric encryption schemes such as AES which work on exact variables. Thus, the Kyber PKE scheme is not perfectly correct and features a certain non-negligible probability of decryption failure.

The size of the error $\mathbf{e}$ is a critical component that determines the correctness and security of Kyber. Increasing (resp. decreasing) the size of error (i.e.) η, increases (resp. decreases) the security of the MLWE problem, as well as the decryption failure rate. The parameters of Kyber are chosen so as to achieve the desired security levels as well as a close to negligible probability of decryption failure.

C. Kleptography

A kleptographic attack [2] typically contains in its core, a SETUP, which is an abbreviation of Secretly Embedded Trapdoor with Universal Protection. The SETUP is an efficient algorithm that can be integrated within the cryptosystem, to covertly leak secret information through the output of the cryptosystem.

Definition II.1 (SETUP) *Let* C *be a publicly known cryptosystem with known parameters. SETUP involve a series of steps that modifies* C *to* C' *that satisfies the following properties:*

1) *The input and output of* C' *agree with the public specifications of* C.

TABLE I
COMPARISON OF PRIOR KLEPTOGRAPHIC ATTACKS ON LATTICE-BASED SCHEMES.

Work	Type of Target Scheme	Target Procedure	Backdoor Scheme	Experimental Validation
Kwant *et al.* [11]	NTRU	Encrypt	ECDH (Pre-Quantum)	Primitive
Xiao and Yu [12]	LWE	Encrypt	Smaller LWE (Post-Quantum)	Primitive
Yang *et al.* [13]	LWE	Encrypt	NTRU (Post-Quantum)	Primitive
Yang *et al.* [14]	LWE	KeyGen	ECDH (Pre-Quantum)	✗
Hemmert [8]	LWE	KeyGen	ECDH (Pre-Quantum)	✗
This work	LWE	KeyGen	ECDH (Pre-Quantum) Classic-Mceliece KEM (Post-Quantum)	Primitve & TLS 1.3

2) C' *can efficiently compute the attacker's public encryption function* Enc_{adv}*, that is contained within* C'.
3) *The output of* C' *must contain information about the secret key, that can by easily recovered by the attacker.*
4) *The attacker's private decryption function* D_{adv} *is not contained within* C'*. Thus, even a complete reverse engineer of* C'*, should not be able to compromise past outputs or future outputs.*
5) *The outputs of* C' *and* C *must be polynomially distinguishable by anyone, except the attacker.*

The above describes a *regular* SETUP, however, there also exists the notion of a *weak* SETUP, which is very similar to the regular SETUP, except that the outputs of C' and C are polynomially indistinguishable to everyone except the attacker and the user of C', who knows its secret key. While weak SETUPs can be detected by the user in theory, it can be quite challenging in practice. This is because the user must first assume that C' contains a SETUP, and must also know how to test for its presence. Moreover, weak SETUPs are sufficient when the user wants to compromise its own secret key.

D. Kleptography v/s Hardware Trojans

One can argue that hardware trojans can also be used to compromise the key-generation procedure of Kyber KEM in a black-box setting, potentially in a much easier manner than kleptographic attacks. One trivial example would be to use a biased random number generator or a PRNG with a known seed to generate the secret keys. However, we identify some key differences and advantages that kleptographic attacks offer to an adversary, compared to hardware trojans.

1) Kleptographic backdoors can be seen as *cryptogrphic trojan horses*, since the backdoors are implemented using asymmetric/public-key cryptograhpic algorithms. These backdoors are robust against reverse engineering. It guarantees that even a complete reverse engineer of the target, cannot be used to exploit the inserted backdoor. Such exclusivity in compromising the cryptosystem, even after full reverse engineer, can be achieved through kleptographic attacks, but not by design level hardware trojans.
2) Once the backdoor is implemented, the attacker does not require access/communication with the target. The knowledge of its outputs is sufficient to compromise the target (i.e.) passive eavesdropping. On the other hand, hardware trojans typically require access/communication with the DUT to trigger the trojan's operation.

E. Related Works

Kwant *et al.* [11] presented the first kleptographic attack on PQC schemes, targeting the encryption procedure of NTRU-based PKE schemes through ECC-based backdoor, which produces backdoor-ed ciphertexts to compromise the message. Xiao and Yu in [12] proposed kleptograhpic attacks, targeting the encryption procedure in LWE-based cryptosystems, based on post-quantum secure backdoors implemented using custom-made small (non-standarized) LWE-based cryptosystems. Yang *et al.* [13] proposed similar attacks, using post-quantum secure NTRU-based backdoors within the encryption procedure.

More recently, Yang *et al.* [14] and Tobias *et al.* [8] presented kleptographic attacks targeting the key-generation procedure for generic LWE-based cryptosystems. However, both works only demonstrate backdoors based on ECC curves (pre-quantum) with 32-byte ciphertexts and their approach does not scale towards utilizing post-quantum secure backdoor with much larger ciphertext sizes (few hundred to few thousand bytes). *Moreover, previous works are devoid of practical instantiations of their attack on real implementations.* In this work, we present the first practical instantiation of a kleptographic attack on Kyber KEM, and our proposed approach works with 100% success rate for key recovery, and is adaptable for both pre-quantum and post-quantum backdoor. While all the aforementioned works only carry out their attack on the primitive level, we also successfully our backdoor-ed Kyber instance into the TLS 1.3 protocol. Refer to Tab.I that summarizes the comparison of prior kleptographic attacks on lattice-based schemes.

III. THE LWE BACKDOOR

In this section, we describe our proposed SETUP for the key-generation procedure of Kyber KEM, which generates backdoor-ed public keys pk that leak the entire secret key sk exclusively to the adversary. Refering to the key-generation procedure in Alg.1, we observe the the secret $\mathbf{s} \in R_q^k$ is entirely generated from a 32-byte seed $seed_B$ (Alg.1, Line 5). Thus, the knowledge of $seed_B$ is sufficient for an adversary to recover $\mathbf{s}$. Our SETUP/backdoor is therefore designed to leak information about the secret $seed_B$ through the public key pk.

A. Threat Model

We assume that the attacker is the designer of a black-box cryptographic module (e.g.) HSM, TPM, with a backdoor in the key-generation procedure of Kyber. The user/victim has black-box access to HSM, and can observe its outputs. The attacker only requires access to the device's public outputs, and does not require active access/communication with the target device.

B. Backdoor/SETUP Methodology and Operation

The proposed backdoor/SETUP can be constructed using KEM or Key-Exchange (KEX) scheme, consisting of three procedures: (1) key-generation ($\mathsf{KeyGen}_{\mathsf{bd}}$), (2) public encryption ($\mathsf{Enc}_{\mathsf{bd}}$) and (3) private decryption ($\mathsf{Dec}_{\mathsf{bd}}$).

The adversary first runs $\mathsf{KeyGen}_{\mathsf{bd}}(\cdot)$ in an offline manner to generate the corresponding public key pk_{bd} and secret key sk_{bd}. Adversary's pk_{bd} is installed within the target cryptosystem, while sk_{bd} is retained offline by the adversary.

We now describe the adversary's public encryption function and private decryption function $\mathsf{Enc}_{\mathsf{bd}}(\cdot)$ and $\mathsf{Dec}_{\mathsf{bd}}(\cdot)$ respectively. $\mathsf{Enc}_{\mathsf{bd}}(\cdot)$ is probabilistic, takes pk_{bd} as input and uses some internal randomness to generate a ciphertext ct_{bd} and a message m (i.e.) $\mathsf{E}_{\mathsf{bd}}(pk_{\mathrm{bd}}) = (ct_{\mathrm{bd}}, m_{\mathrm{bd}})$. $\mathsf{Dec}_{\mathsf{bd}}(\cdot)$ decrypts ct_{bd} using the backdoor secret key sk_{bd} to generate the message m'_{bd} (i.e.) $\mathsf{Dec}_{\mathsf{bd}}(sk_{\mathrm{bd}}, ct_{\mathrm{bd}}) = (m'_{\mathrm{bd}})$.

Within the backdoor-ed key-generation procedure C' of Kyber, $\mathsf{Enc}_{\mathsf{bd}}(pk_{\mathrm{bd}})$ is executed for every run of C' and the first 32 bytes of the output message m_{bd} is simply chosen to be the secret seed $seed_B$. If the associated ciphertext ct_{bd} can be exfiltrated through the public key pk of Kyber KEM in a manner recoverable by the adversary, then the adversary can simply decrypt ct_{bd} using sk_{bd} to recover the same message $m_{\mathrm{bd}} = seed_B$. In the following, we describe our proposed method to exfiltrate the payload ct_{bd} through the public key of Kyber KEM.

C. Exfiltration of Payload within the Public Key

The payload ct_{bd} should be integrated into the public key pk, while satisfying the following two requirements:

1) ct_{bd} should be efficiently recoverable from pk.
2) The backdoor-ed public key pk of Kyber KEM, should still remain valid for KEM exchanges with an overwhelming probability, else the backdoor can be easily detected by the user.

We choose to exfiltrate ct_{bd} within the LWE component $\mathbf{t} \in R_q^k$ of the public key $pk = (seed_A \in \mathcal{B}^{32}, \hat{\mathbf{t}} \in R_q^k)$. We propose to *additively* integrate the payload ct_{bd} into the pseudo-random LWE component $\mathbf{t}$ with uniformly random coefficients in $[0, q]$ in the following manner.

The payload ct_{bd} can be considered to be a string of b bits (i.e.) $ct_{\mathrm{bd}} = (bt_0, bt_1, \ldots, bt_b)$. It is encoded into a module $\mathbf{p} \in R_q^k$ with $k \cdot n$ coefficients. Thus, each coefficient of $\mathbf{p}$ can be represented in $c = \lceil b/(k \cdot n) \rceil$ bits. Thus, $\mathbf{p} \in R_q^k$ contains $\lceil b/c \rceil$ non-zero coefficients in $[0, 2^c)$, where

$$\mathbf{p}[i] = \sum_{j=i \cdot c}^{j=(i+1)\cdot c} bt_j \cdot 2^{j-(i \cdot c)} \text{ for } i \in [0, \lceil b/c \rceil) \tag{1}$$

while all the other coefficients of $\mathbf{p}$ are 0. We cannot simply add $\mathbf{p}$ to the pseudo-random $\mathbf{t}$, as it would completely hide $\mathbf{p}$. We thus propose to add to $\mathbf{t} \in R_q^k$, a small compensation denoted as $\mathbf{h} \in R_q^k$ (i.e.) $\mathbf{t}' = \mathbf{t} + \mathbf{h}$ such that:

$$\mathbf{t}'[i] \equiv \mathbf{p}[i] \mod c \text{ for } i \in [0, \lceil b/c \rfloor) \tag{2}$$

This compensation component $\mathbf{h}$ can be trivially computed, one coefficient at a time, and the coefficients of $\mathbf{h}$ lie in the zero-centered range of $[\lceil -c/2 \rceil, \lceil c/2 \rceil]$. Given $\hat{\mathbf{t}'}$, one can easily recover the $\lceil b/c \rfloor$ non-zero coefficients of $\mathbf{p}$ by simply computing the remainder of corresponding coefficients of $\mathbf{t}'$ modulo c, as shown in Eqn.2. Once ct_{bd} is recovered, an adversary can decrypt ct_{bd} using sk_{bd} to recover the secret seed $seed_B$. Refer to Alg.2 for the entire kleptographic attack on Kyber KEM, that describes backdoor installation, backdoor-ed key generation of Kyber KEM, as well as secret key recovery.

1) Comparison with Prior Works on Payload Exfiltration: We consider the attacks of Yang *et al.* [14] and Hemmert [8] which also target the key-generation procedure of LWE-based schemes. They however choose to exfiltrate ct_{bd} by simply utilizing it as $seed_A$ within the key-generation procedure (Alg.1, Line 2). Since $seed_A$ only spans 32 bytes for Kyber KEM, it forces the adversary to utilize a scheme for backdoor whose ciphertext size is 32 bytes. This only applies to a very limited set of pre-quantum cryptosystems such as ECC, while none of the PQC-based schemes have such small ciphertext sizes. However, our approach of integrating the payload into $\mathbf{t}$ allows for much larger payloads, enabling use of a wide variety of pre-quantum and post-quantum schemes as backdoor.

D. Analyzing Decryption Failure Rate for Backdoor-ed Keys

By exfiltrating the payload ct_{bd} within $\mathbf{t}'$, we are adding an additional error $\mathbf{h}$ into $\mathbf{t}$, with the new error being $\mathbf{e}' = (\mathbf{e}+\mathbf{h})$. For recommended parameter sets of Kyber (Kyber768), the coefficients of the error $\mathbf{e}$ follow a Centered Binomial Distribution (CBD) in the range $[-2, 2]$, while the coefficients of the compensation $\mathbf{h}$ are uniformly in random in the range $[-c/2, c/2]$ (Refer Sec.III-C). This increases the error span to the range $[-(c/2+2), (c/2+2)]$, which also correspondingly increases the decryption failure rate of Kyber KEM.

For Kyber768, the decryption failure rate is estimated to be $\approx 2^{-165.2}$, which is very conservative for practical applications. Refer to Fig.1, which shows the trend of decryption failure probability for Kyber768 with increasing size of the error span. These numbers were estimated using the open-source tool provided by the authors of Kyber KEM[2]. As we show in the following discussion, our practical backdoor instantiations only increase the error span to $[-4, 4]$. This

[2]Available at https://github.com/pq-crystals/security-estimates

Algorithm 2 Kleptographic Attack on Kyber KEM

procedure BACKDOOR INSTALLATION (OFFLINE)
  $(pk_{\text{bd}}, sk_{\text{bd}}) = \mathsf{KeyGen}_{\text{bd}}()$
  Install pk_{bd} on target, retain sk_{bd} offline
end procedure

---

procedure BACKDOOR-ED KEY GENERATION: $\mathsf{C}'(pk_{\text{bd}})$
  $\mathsf{Enc}_{\text{bd}}(pk_{\text{bd}}) = (ct_{\text{bd}}, m)$ ▷ Backdoor Encryption
  $seed_A \in \mathcal{B}^{32} \leftarrow \mathsf{Sample}_U()$
  $seed_B \in \mathcal{B}^{32} = m_{\text{bd}}$ ▷ m_{bd} used as $seed_B$
  $\hat{\mathbf{a}} \in R_q^{k \times k} \leftarrow \mathsf{Expand}(seed_A)$ ▷ Sample $\hat{\mathbf{a}}$
  $\mathbf{s} \in R_q^k \leftarrow \mathsf{Sample}_B(seed_B, coin_s)$ ▷ Sample $\mathbf{s}$
  $\mathbf{e} \in R_q^k \leftarrow \mathsf{Sample}_B(seed_B, coin_e)$ ▷ Sample $\mathbf{e}$
  $\mathbf{t} = \mathsf{INTT}(\hat{\mathbf{a}} \circ \mathsf{NTT}(\mathbf{s})) + \mathbf{e}$
  $ct_{\text{bd}} = (bt_0, bt_1, \ldots, bt_b)$; $c = \lceil b/(k \cdot n) \rceil$
  $\mathbf{p} = 0$ ▷ Compute Encoded Payload $\mathbf{p}$
  for i from 0 to $\lceil b/c \rceil - 1$ **do**
    $\mathbf{p}[i] = \sum_{j=i \cdot c}^{j=(i+1) \cdot c} bt_j \cdot 2^{j-(i \cdot c)}$
  end for
  $\mathbf{h} = 0$ ▷ Compute Compensation $\mathbf{h}$
  for i from 0 to $\lceil b/c \rceil - 1$ **do**
    $\mathbf{h}[i] = \mathsf{Compute\_Compensation}(\mathbf{t}[i], \mathbf{p}[i])$
  end for
  $\mathbf{t}' = \mathbf{t} + \mathbf{h}$ ▷ Adding Compensation $\mathbf{h}$ to $\mathbf{t}$
  Return $pk = (seed_A, \hat{\mathbf{t}'} = \mathsf{NTT}(\mathbf{t}'))$
end procedure

---

procedure SECRET KEY RECOVERY($pk = (seed_A, \hat{\mathbf{t}'})$)
  $\mathbf{t}' = \mathsf{INTT}(\hat{\mathbf{t}'})$
  $\mathbf{p} = 0$ ▷ Computing encoded payload $\mathbf{p}$ from $\mathbf{t}'$
  for i from 0 to $\lceil b/c \rceil - 1$ **do**
    $\mathbf{p}[i] = \mathbf{t}'[i] \mod c$
  end for
  $ct_{\text{bd}} = \mathsf{Reconstruct}(\mathbf{p})$ ▷ Recovering payload ct_{bd}
  $m'_{\text{bd}} = \mathsf{Dec}_{\text{bd}}(sk_{\text{bd}}, ct_{\text{bd}})$ ▷ Backdoor Decryption
  $\mathbf{s} \in R_q^k \leftarrow \mathsf{Sample}_B(m'_{\text{bd}}, coin_s)$ ▷ Recover secret module $\mathbf{s}$
  Return $\mathbf{s}$
end procedure

corresponds to an estimated failure probability of $\approx 2^{-124.5}$, that is still considered to be negligible for practical applications. The trend shows that one can exfiltrate larger payloads within the public key, if larger decryption failure rates can be tolerated for practical applications (i.e.) error span of $[-11, 11]$ and $[-25, 25]$ for decryption failure rate of 2^{-64} and 2^{-32} respectively.

E. Concrete Choices for Backdoor

We demonstrate our proposed kleptographic attack using two types of backdoors: 1) Classical (or Pre-Quantum) backdoor and 2) Post-Quantum backdoor.

1) Pre-Quantum Backdoor: We use the well-known Elliptic Curve Diffie Hellman (ECDH) to instantiate the pre-quantum backdoor. Our choice was motivated by the smallest ciphertext sizes offered by ECC based schemes. While use of PQC will see rise in adoption, ECDH is expected to stay for years to come in the form of the classical-postquantum hybrid.

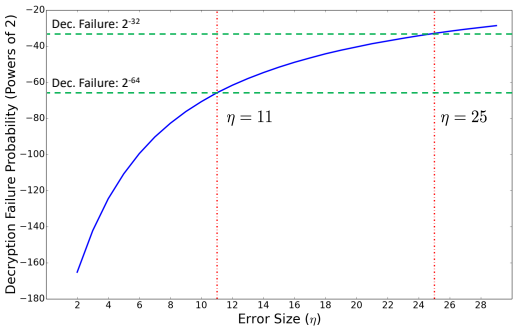

Fig. 1. Decryption Failure Probability for the recommended parameters of Kyber KEM (Kyber768) against increasing size of error.

We use the specific instance of curve $K-409$ from [15] for our experiments. It provides 192-bit security against a classical adversary, matching the security strength of the recommended parameter sets of Kyber (i.e.) Kyber768.

- $\mathsf{KeyGen}_{\text{bd}}$ (Offline): Let the base point of ECC curve be P. A random scalar A is sampled and $P_A = A \cdot P$ is the public-key of the ECDH backdoor (pk_{bd}).
- Enc_{bd} (On Device): A random scalar B is sampled and the ciphertext ct_{bd} is $P_B = B \cdot P$ and the shared message $m = B \cdot P_A = B \cdot A \cdot P$.
- Dec_{bd} (Offline): The product of the ciphertext ct_{bd} and secret scalar A yields the same shared message $m' = P_B \cdot A = B \cdot A \cdot P$.

The size of the ciphertext and the public key for curve $K-409$ is only 104 bytes (832 bits). Given $\mathbf{t} \in R_q^k$ for Kyber768 has 768 coefficients, we can hide $\lceil 1.08 \rceil = 2$ bits in about 416 coefficients of $\mathbf{t}$. This amounts to an additional error in the range $[-2, 2]$, thereby increasing the effective error span to $[-4, 4]$, whose decryption failure probability is $\approx 2^{-124.5}$ (Fig.1).

2) Post-Quantum Backdoor: We use the Classic Mceliece KEM to instantiate the post-quantum backdoor for Kyber KEM. Classic Mceliece KEM offers the smallest ciphertext sizes among all the PQC based KEMs in the NIST standardization process. We instantiate our backdoor using the mceliece460896 variant (NIST Security level 3), whose ciphertext size is 156 bytes (1248 bits). This amounts to exfiltrating $\lceil 1.625 \rceil = 2$ bits in about 624 coefficients of $\mathbf{t}$. Since the internal working of the scheme is not relevant for our work, we refer the reader to [16] for internal working of the different procedures of the Classic Mceliece scheme.

F. Analyzing Properties of Proposed SETUP

We now verify the security properties of our proposed backdoor, compared to the properties of a SETUP listed in Section II-C.

1) We can easily observe that both the input and output interface of C (Alg.1) and C' (Alg.2) are identical.
2) As shown in Alg.2, C' is efficient to compute in polynonmial time, as the only additional operations are: 1) adversary's public encryption $\mathsf{Enc}_{\text{bd}}(\cdot)$ and 2) payload exfiltration, both of which can be computed efficiently.
3) The output of C' only contains information about ct_{bd}. Thus, the attacker with knowledge of sk_{bd} can exclusively exploit the backdoor. This guarantee is obtained from the security properties of the cryptosystem used to implement the backdoor (i.e.) ECDH or Classic Mceliece KEM.
4) C' only contains the backdoor's public key pk_{bd}, and thus even if one is able to fully reverse engineer C', it cannot exploit the past/future outputs of C', without the knowledge of sk_{bd}.

We now analyze the last property (i.e.) ability to polynomially distinguish between output of C and C'. An eavesdropper can only observe the uniformly random public key $pk = (seed_A, \hat{\mathbf{t}}')$. The coefficients of $\hat{\mathbf{t}}'$ in the NTT domain are uniformly in random in $[0, q]$, similar to the valid procedure C. However, the user who has access to the secret key $sk = \hat{\mathbf{s}}$, can recompute the error to detect the backdoor:

$$\mathbf{e} = \mathsf{INTT}(\hat{\mathbf{t}}' - \hat{\mathbf{a}} \circ \hat{\mathbf{s}}) \tag{3}$$

The increased error span of $[-(c/2+2), (c/2+2)]$ can be identified by the user. We propose propose this technique as a concrete countermeasure, that can be implemented by the user of the device. Since the user can detect the backdoor, our attack belongs to the category of *weak* SETUP. However, a user in practice, is typically unaware of the presence of the SETUP, and even otherwise, does not know how to test for its presence.

IV. Experimental Results

A. Validating Efficacy of Backdoor

We experimentally validated our attack on the key-generation procedure of Kyber, using both pre-quantum (ECDH based on ECC curve K-409) and post-quantum backdoor (Classic Mcelice KEM). We implemented our attack on the open-source reference C implementation of Kyber KEM from the NIST submission package [10], running on an Intel x86-64 machine. We used the C implementation from tiny-ECDH-c project[3]. For Classic Mceliece KEM, we used the reference C implementation from the NIST submission package [16]. We executed our attack on the backdoor-ed key generation procedure of Kyber768, for over a million times (10^6) for each backdoor. It recovers the secret with a 100% success rate, while not observing even a single decryption failure, which also positively confirms our theoretical analysis presented in Section III-D. Our attack is adaptable in a straightforward manner to all parameters of Kyber KEM.

[3]Available at https://github.com/kokke/tiny-ECDH-c

B. Integration of Backdoor-ed Kyber into TLS

While we verified our attack on Kyber KEM as a standalone primitive, these primitives are typically employed within cryptographic protocols such as TLS 1.3 in practical applications. We therefore verified our attack at the protocol level, by instantiating our backdoor-ed Kyber KEM within the TLS 1.3 protocol. We successfully integrated our backdoor-ed Kyber KEM into the open-source implementation of PQC enabled OpenSSL 1.1.1, provided by the Open-Quantum Safe (OQS) project [17][4]. We performed $10,000$ TLS 1.3 handshakes using backdoor-ed Kyber KEM, and were able to recover the secret key with 100% success rate for all handshakes, while observing no decryption failures. We therefore present the first practical instantiation of a backdoor-ed PQC-based TLS 1.3 protocol, demonstrating the capability of our proposed kleptographic attack at the protocol level.

[4]Available at https://github.com/open-quantum-safe/openssl

V. Conclusion

In this work, we present the first practical instantiation of a kleptographic attack on Kyber KEM. We present novel techniques to insert backdoors in its key-generation procedure, that allows an attacker to utilize both pre-quantum as well as post-quantum schemes to instantiate the backdoor. Our backdoor-ed key generation procedure is able to leak the complete secret key through the public keys with 100% success rate, while remaining valid with an overwhelming probability. We also demonstrate the first practical instantiations of such attack at the protocol level by validating it on TLS 1.3.

References

[1] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, C. Miller, D. Moody, R. Peralta *et al.*, "Status report on the third round of the nist post-quantum cryptography standardization process," *National Institute of Standards and Technology, Gaithersburg*, 2022.

[2] A. L. Young and M. Yung, "Cryptovirology: Extortion-based security threats and countermeasures," in *1996 IEEE Symposium on Security and Privacy, May 6-8, 1996, Oakland, CA, USA*. IEEE Computer Society, 1996, pp. 129–140. [Online]. Available: https://doi.org/10.1109/SECPRI.1996.502676

[3] Adam L. Young and Moti Yung, "Kleptography: Using cryptography against cryptography," in *Advances in Cryptology - EUROCRYPT '97*, ser. Lecture Notes in Computer Science, W. Fumy, Ed., vol. 1233. Springer, 1997, pp. 62–74. [Online]. Available: https://doi.org/10.1007/3-540-69053-0_6

[4] N. Perlroth, J. Larson, and S. Shane, "Nsa able to foil basic safeguards of privacy on web," *The New York Times*, vol. 5, pp. 1–8, 2013.

[5] S. NIST, "800-90," *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, 2007.

[6] A. Young and M. Yung, "Cryptography as an attack technology: Proving the rsa/factoring kleptographic attack," in *The new codebreakers*. Springer, 2016, pp. 243–255.

[7] Young, Adam and Yung, Moti, "The prevalence of kleptographic attacks on discrete-log based cryptosystems," in *Annual International Cryptology Conference*. Springer, 1997, pp. 264–276.

[8] T. Hemmert, "How to backdoor lwe-like cryptosystems," *Cryptology ePrint Archive*, vol. 2022, p. 1381, 2022.

[9] R. Avanzi, J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Kyber (version 3.0): Algorithm specifications and supporting documentation (October 1, 2020)," 2020.

[10] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-Kyber (version 3.02) - Algorithm Specifications And Supporting Documentation (August 4, 2021)."
[11] R. Kwant, T. Lange, and K. Thissen, "Lattice klepto," in *International Conference on Selected Areas in Cryptography*. Springer, 2017, pp. 336–354.
[12] D. Xiao and Y. Yu, "Klepto for ring-lwe encryption," *The Computer Journal*, vol. 61, no. 8, pp. 1228–1239, 2018.
[13] Z. Yang, R. Chen, C. Li, L. Qu, and G. Yang, "On the security of lwe cryptosystem against subversion attacks," *The Computer Journal*, vol. 63, no. 4, pp. 495–507, 2020.
[14] Z. Yang, T. Xie, and Y. Pan, "Lattice klepto revisited," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020, pp. 867–873.
[15] M. Brown, D. Hankerson, J. López, and A. Menezes, "Software implementation of the nist elliptic curves over prime fields," in *Cryptographers' Track at the RSA Conference*. Springer, 2001, pp. 250–265.
[16] D. J. Bernstein, T. Chou, T. Lange, I. von Maurich, R. Misoczki, R. Niederhagen, E. Persichetti, C. Peters, P. Schwabe, N. Sendrier *et al.*, "Classic mceliece: conservative code-based cryptography," *NIST submissions*, 2021.
[17] M. Mosca and D. Stebila, "Open Quantum Safe," 2017.