# 1 Definitions

***Non-collapsing hash function.*** A hash function that is collision-resistant and not infinitely-often collapsing. (See [Zha17] for definitions of 'collision-resistant' and 'infinitely-often collapsing'. We can assume that there are adversaries who win the collapsing game for such functions with probability near 1. See [GYZ17], [Zha17] for examples of how one can boost the success probability of any bad adversary who breaks infinitely-often collapsing security so that it becomes a very good adversary.)

***Part-probabilistic non-collapsing hash function.*** A function $F(k, x, r)$ whose output takes the form $H(k, x) \mid\mid R(H(k, x), r)$, where $H$ is a (deterministic) collision-resistant hash function, and $R$ is another function whose output is permitted to depend upon the randomness $r$ and the output of $H$. We require the following security properties of $F$:

1. Collision resistance for $H$: For any quantum polynomial time adversary $A$,

$$\Pr[H(k, x_0) = H(k, x_1) \wedge x_0 \neq x_1 \ : \ (x_0, x_1) \leftarrow A(k), k \leftarrow \{0,1\}^\lambda] < \mathsf{negl}(\lambda)$$

2. (Infinitely-often) non-collapsing: There exists an adversary $A$ (consisting of two phases, $A_0$ and $A_1$) who can win the following game with probability $1 - \gamma$, where $\gamma$ is negligible.

   - The challenger has an input bit $b$.
   - The challenger chooses a random key $k$, which it gives to $A_0$.
   - $A_0$ creates a superposition $|\psi\rangle = \sum_x \alpha_x |x\rangle$ and submits this state to the challenger.
   - The challenger generates a random $r$. It evaluates $F(k, \cdot, r)$ in superposition on $|\psi\rangle$, to get the state $\sum_x \alpha_x |x, H(k, x), R(H(k, x), r)\rangle$.
   - The challenger does one of the following:
     - If $b = 0$, it measures the last two registers, and returns the state $\sum_{x:H(k,x)=y} \alpha_x |x, y, R(y, r)\rangle$ to $A$.
     - If $b = 1$, it measures the entire state, and returns the state $|x_0, y, R(y, r)\rangle$ (for some $x_0$) to $A$.
   - $A_1$ outputs a guess for $b$. If $A_1$ is correct, $A$ wins the game.

Note that, under this definition of 'non-collapsing' (which mimics [Zha17]'s definition), the distinguisher $A_1$ is only guaranteed to exist *if the challenger behaves honestly.* $A_1$'s success may depend upon $r$ being honestly generated, and upon $R$ being honestly run; it has no way of verifying that either is the case. In the generic hash function setting, we cannot guarantee that $R$ will be run honestly on a random $r$, and we cannot guarantee that the distinguisher $A_1$ which wins the game above will still be useful if this is not the case.

It is evident that the (deterministic) non-collapsing hash function is a special case of the part-probabilistic non-collapsing hash function, so that any NCH function is also a PP-NCH function.

***Chosen-y-secure hash function.*** A hash function $H(k, x)$ for which no quantum polynomial time adversary can win the following game with more than negligible probability:

- The challenger chooses a random key $k$, which it gives to $A$.
- The challenger creates a uniform superposition over all inputs $x$ in the input space of $H$, and evaluates $H(k, \cdot)$ upon this superposition to obtain the state $\sum_x \alpha_x |x, H(k, x)\rangle$. It then measures the output register to obtain a state $|\psi_y\rangle = \sum_{x:H(k,x)=y} \alpha_x |x, y\rangle$ for some random $y$.
- The challenger gives $|\psi_y\rangle$ to the adversary. The adversary wins the game if it can recover $x_0, x_1$ such that $H(k, x_0) = H(k, x_1) = y$.

The collapsing security game can be defined in the same way for CYS hash functions that it is for collision-resistant hash functions.

Note that it is easy to construct a collision-resistance adversary from a chosen-$y$ adversary, and that, therefore, any collision-resistant hash function is also a chosen-$y$-secure hash function. This notion of 'chosen $y$' security is close to that of second preimage resistance security; the former can be considered a strengthening of the latter to suit the quantum setting.

Note, in addition, that any PP-NCH function can be transformed into a CYS-NCH function. Chosen-$y$ security follows directly from the assumption of collision resistance for the deterministic part of the PP-NCH function. The non-collapsing property, meanwhile, follows from the fact that the chosen-$y$ setting already demands that $y$ is chosen randomly (presumably by some trusted party) if the CYS function's preimage security is to hold. Any construction using a chosen-$y$ hash function for its preimage security properties, therefore, must generate honest randomness when it generates $y$; and, as such, $y$ can be used as a source of randomness for the randomised part of the PP-NCH function.

# 2 NCH implies BZ-GYZ

**Claim.** Any non-collapsing hash function can be used to build a one-time signature scheme that is Boneh-Zhandry secure but not Garg-Yuen-Zhandry secure.

*Scheme.* Given a non-collapsing hash function $F$, and an arbitrary BZ-secure one-time signature scheme $(\mathsf{Gen}_{\mathsf{BZ}}, \mathsf{Sign}_{\mathsf{BZ}}, \mathsf{Ver}_{\mathsf{BZ}})$ (these are easy to produce; for example, the standard Lamport construction of a one-time signature scheme from a collision-resistant hash function is BZ-secure, according to [BZ13]), we construct a BZ-GYZ scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ as follows.

- $\mathsf{Gen}$ simply runs $\mathsf{Gen}_{\mathsf{BZ}}$ to generate a pair of keys $(\mathsf{pk}, \mathsf{sk})$ for the BZ signature scheme.
- $\mathsf{Sign}(\mathsf{sk}, m) = F(m) \parallel \mathsf{Sign}_{\mathsf{BZ}}(\mathsf{sk}, F(m)) = \sigma$. In other words, $\mathsf{Sign}$ applies the non-collapsing hash function $F$ to the message, signs the hashed message using the BZ scheme, and outputs the hashed message concatenated with the signature it obtains from the BZ signing oracle.
- $\mathsf{Ver}(\mathsf{pk}, m, \sigma)$ firstly hashes the message $m$ to obtain $F(m)$, and then verifies $F(m)$ using $\mathsf{Ver}_{\mathsf{BZ}}$ and $\mathsf{pk}$.

*Proof.* We firstly prove that this scheme is BZ-secure, assuming that $(\mathsf{Gen}_{\mathsf{BZ}}, \mathsf{Sign}_{\mathsf{BZ}}, \mathsf{Ver}_{\mathsf{BZ}})$ is BZ-secure.

Suppose we have some adversary $A$ who is able to break the BZ-security of $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$. The adversary $B$ can then use $A$ to break the BZ-security of $(\mathsf{Gen}_{\mathsf{BZ}}, \mathsf{Sign}_{\mathsf{BZ}}, \mathsf{Ver}_{\mathsf{BZ}})$ as follows:

- $B$ receives $\mathsf{pk}$ from its challenger. It passes $\mathsf{pk}$ on to $A$.
- $A$ creates a superposition of messages $\sum_m \alpha_m |m\rangle$ and gives it to $B$ as a query. $B$ computes $F$ on it in superposition, and then passes $\sum_m \alpha_m |m, F(m)\rangle$ on to its challenger, who computes $\mathsf{Sign}_{\mathsf{BZ}}$ on it and returns the state $\sum_m \alpha_m |\, m, F(m), \mathsf{Sign}_{\mathsf{BZ}}(\mathsf{sk}, F(m))\,\rangle$ to $B$. $B$ gives this state to $A$.
- $A$ outputs its (classical) forgery for $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$. This forgery will take the form $((m_0, F(m_0), \sigma_0), (m_1, F(m_1), \sigma_1))$, where $m_0$ and $m_1$ are two distinct messages.
- If $F(m_0) = F(m_1)$, then we have found a collision for $F$, which ought to be impossible, because we assume that $F$, a non-collapsing hash function, is collision-resistant. If $F(m_0) \neq F(m_1)$, then $B$ outputs $((F(m_0), \sigma_0), (F(m_1), \sigma_1))$ as its forgery for the $(\mathsf{Gen}_{\mathsf{BZ}}, \mathsf{Sign}_{\mathsf{BZ}}, \mathsf{Ver}_{\mathsf{BZ}})$ scheme.
- If $A$'s success probability is non-negligible, then so is $B$'s.

Therefore, $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ is BZ-secure if $(\mathsf{Gen}_{\mathsf{BZ}}, \mathsf{Sign}_{\mathsf{BZ}}, \mathsf{Ver}_{\mathsf{BZ}})$ is BZ-secure.

We now prove that the proposed scheme is not GYZ-secure. To do this, we use the fact that $F$ is non-collapsing. Let $D$ be an adversary which can break collapsing security for $F$. Following the proof to Theorem 13 in [GYZ17], we construct an adversary $A$ who uses $D$ to break the GYZ-security of $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$. $A$ acts as follows:

- $A$ queries $D$ to get a superposition of 'messages' (preimages) $\rho = \sum_{m,m'} \alpha_m \alpha^*_{m'} |m\rangle\langle m'|$, and places this superposition $\rho$ in its message register. $A$ then flips a coin with outputs in $\{0,1\}$ and measures the message register iff the coin gives 0. This results in the following state:

$$\frac{1}{2}\Big[\ |0\rangle\langle 0| \otimes \sum_m |\alpha_m|^2 |m\rangle\langle m| + |1\rangle\langle 1| \otimes \rho\ \Big] \tag{1}$$

- $A$ sends this state to the GYZ signing oracle, which signs it and places the signature in a newly created pair of signature registers. The result is then

$$\tfrac{1}{2}\Big[\ \begin{array}{lllll} \sum_m |\alpha_m|^2 & |0\rangle\langle 0| \otimes & |m\rangle\langle m| \otimes & |F(m)\rangle\langle F(m)| \otimes & |\,\mathsf{Sign_{BZ}}(\mathsf{sk}, F(m))\,\rangle\langle\,\mathsf{Sign_{BZ}}(\mathsf{sk}, F(m))\,| \\ +\sum_{m,m'} \alpha_m \alpha^*_{m'} & |1\rangle\langle 1| \otimes & |m\rangle\langle m'| \otimes & |F(m)\rangle\langle F(m')| \otimes & |\,\mathsf{Sign_{BZ}}(\mathsf{sk}, F(m))\,\rangle\langle\,\mathsf{Sign_{BZ}}(\mathsf{sk}, F(m'))\,| \end{array}\ \Big] \tag{2}$$

  $A$ measures the '$F(m)$' register (the third register from the left), to obtain the state

$$\tfrac{1}{2}\sum_y \beta_y \Big[\ \begin{array}{lllll} \sum_{m:F(m)=y} |\alpha_m|^2 & |0\rangle\langle 0| \otimes & |m\rangle\langle m| \otimes & |y\rangle\langle y| \otimes & |\,\mathsf{Sign_{BZ}}(\mathsf{sk}, y)\,\rangle\langle\,\mathsf{Sign_{BZ}}(\mathsf{sk}, y)\,| \\ + & |1\rangle\langle 1| \otimes & |\psi_y\rangle\langle\psi_y| \otimes & |y\rangle\langle y| \otimes & |\,\mathsf{Sign_{BZ}}(\mathsf{sk}, y)\,\rangle\langle\,\mathsf{Sign_{BZ}}(\mathsf{sk}, y)\,| \end{array}\ \Big] \tag{3}$$

  where $|\psi_y\rangle = \sum_{m:F(m)=y} \alpha_m |m\rangle$.

- $A$ applies $D$ to the second and third registers from the left and saves $D$'s output in a newly created ancilla register. If $D$ is a very good distinguisher which gives the right answer with probability $1-\gamma$, then, by the gentle measurement lemma, the resulting state is $4\sqrt{2\gamma}$ close to

$$\tfrac{1}{2}\sum_y \beta_y \Big[\ \begin{array}{lllll} \sum_{m:F(m)=y} |\alpha_m|^2 & |0\rangle\langle 0| \otimes & |m\rangle\langle m| \otimes & |y\rangle\langle y| \otimes & |\,\mathsf{Sign_{BZ}}(\mathsf{sk}, y)\,\rangle\langle\,\mathsf{Sign_{BZ}}(\mathsf{sk}, y)\,| \otimes |0\rangle\langle 0| \\ + & |1\rangle\langle 1| \otimes & |\psi_y\rangle\langle\psi_y| \otimes & |y\rangle\langle y| \otimes & |\,\mathsf{Sign_{BZ}}(\mathsf{sk}, y)\,\rangle\langle\,\mathsf{Sign_{BZ}}(\mathsf{sk}, y)\,| \otimes |1\rangle\langle 1| \end{array}\ \Big] \tag{4}$$

  (It is possible to boost the success probability of any bad distinguisher so that it becomes a good distinguisher, as long as the bad distinguisher still breaks collapsing security for $F$. See Section 4.1 of [GYZ17].)

- A sends this state to its challenger for verification. Because all the signatures in this state were legally obtained, this state passes verification with probability 1, and is not perturbed by verification. Therefore, the challenger's output $\texttt{GYZ-Exp}(A)$ is equal to the state above.

- By the same reasoning that is used in the proof of Theorem 13 in [GYZ17], there is no basis-respecting adversary that could produce such an output, as basis-respecting adversaries must commute with measurement in the computational basis. (Intuitively, it is clear that the distinguisher does not commute with measurement, because its very purpose is to determine whether or not a state has been measured.)

Therefore, if $F$ is non-collapsing, $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ is not GYZ-secure.

# 3 NCH implies quantum tokens

**Claim.** Any non-collapsing hash function can be used to build a testable tokenised signature scheme.

*Scheme.* Given an arbitrary non-collapsing hash function $F$, we construct a one-bit, one-time, testable tokenised signature scheme. (By Section 5 of [BS16], a one-bit, one-time tokenised signature scheme can be extended to a fully-fledged tokenised signature scheme via a series of reductions.)

Because $F$ is non-collapsing, there exists an adversary $A$ that can break collapsing security for $F$ with very high success probability $1-\gamma$. We use $A$, along with an arbitrary classical digital signature scheme $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$, to construct a tokens scheme.

- Let $(\mathsf{pk}, \mathsf{sk})$ denote the public and secret keys for the classical signature scheme, and let $(pk, sk)$ stand for the keys to our tokens scheme.
- $A$ consists of two phases: 1) the phase which outputs a superposition of messages for its challenger to hash, and 2) the phase which guesses whether the challenger measured its entire state or only the output registers. Following [Zha17], we let $A_0$ denote the first phase and $A_1$ the second.
- $\mathsf{key\text{-}gen}$ runs $\mathsf{Gen}$ to generate $(\mathsf{pk}, \mathsf{sk})$ for the digital signature scheme. It then outputs $sk = (\mathsf{sk}, A_0)$ and $pk = (\mathsf{pk}, A_1)$.
- $\mathsf{token\text{-}gen}(sk)$ firstly runs $A_0$, twice, to generate two superpositions of messages $\sum_{m_0} \alpha_{m_0} |m_0\rangle$ and $\sum_{m_1} \alpha_{m_1} |m_1\rangle$. It computes $F$ in superposition on both, and measures both output registers. The result is two states $|\psi_{y_0}\rangle, |\psi_{y_1}\rangle$, where $|\psi_{y_b}\rangle = \sum_{m:F(m)=y_b} |m, F(m)\rangle$. It then signs the tuple $(y_0, y_1)$ using $\mathsf{sk}$ and $\mathsf{Sign}$. $\mathsf{token\text{-}gen}$ outputs $\left(|\psi_{y_0}\rangle, |\psi_{y_1}\rangle, \mathsf{Sign}(\mathsf{sk}, (y_0, y_1))\right)$ as $|\text{🔏}\rangle$.
- $\mathsf{sign}(\alpha \in \{0,1\}, |\text{🔏}\rangle)$ measures the input register of $|\psi_{y_\alpha}\rangle$, and outputs, as a signature for $\alpha$, a preimage $m_\alpha$ for $y_\alpha$ under $F$, along with an unaltered $|\psi_{y_{1-\alpha}}\rangle$ and the signature $\mathsf{Sign}(\mathsf{sk}, (y_0, y_1))$.
- $\mathsf{verify}\Big(pk, \alpha, \sigma = \left(m_\alpha, |\psi_{y_{1-\alpha}}\rangle, \mathsf{Sign}(\mathsf{sk}, (y_0, y_1))\right)\Big)$ firstly verifies the signature on $(y_0, y_1)$ using $\mathsf{pk}$ and $\mathsf{Ver}$. Following this, it checks that $m_\alpha$ hashes to $y_\alpha$, and that the superposition of messages in the message register of $|\psi_{y_{1-\alpha}}\rangle$ hashes to $y_{1-\alpha}$. It then checks, using $A_1$, that the purported $|\psi_{y_{1-\alpha}}\rangle$ really does still have an unmeasured message register. In more precise terms, if $A_1$ outputs 'measured', $\mathsf{verify}$ outputs $F$; otherwise, $\mathsf{verify}$ outputs $T$.
- $\mathsf{verify\text{-}token}(|\text{🔏}\rangle)$ applies $\mathsf{Ver}_0$, from Section 4.2 of [Zha17], to both candidate states $|\psi_{y_0}\rangle$ and $|\psi_{y_1}\rangle$.

*Proof.*

*Testability.* Testability follows directly from Zhandry's work in Section 4.2 of [Zha17]. The correctness portion of testability is identical to the correctness requirement for quantum money, which Zhandry's construction satisfies. The security portion of testability also follows from Section 4.2. By Zhandry's proof of security, we can assume that two dishonest candidate states $|\phi_0\rangle$ and $|\phi_1\rangle$ pass $\mathsf{verify\text{-}token}$ with at most negligible probability. Therefore, except with negligible probability, any two states passing $\mathsf{verify\text{-}token}$ will be honest states. Because $\mathsf{verify}$ always accepts honest tokens, our scheme satisfies equation (8) of [BS16].

*Unforgeability.* Suppose that there is an adversary who can, after seeing a single token $|\text{🔏}\rangle = (|\psi_{y_0}\rangle, |\psi_{y_1}\rangle)$, produce two signatures $(x_0, |\phi_0\rangle)$ and $(x_1, |\phi_1\rangle)$ that both pass $\mathsf{verify}$ with non-negligible probability. (Assume, for the present, that the adversary did not attempt to forge a signature on $(y_0, y_1)$.)

Without loss of generality, consider the $(x_0, |\phi_0\rangle)$ tuple. To pass the hash tests which $\mathsf{verify}$ executes, $x_0$ must be a valid preimage to $y_0$ under $F$. Note that the only states $|\phi_0\rangle$ which the adversary can hold without violating the collision resistance of $F$, given that he already has $x_0$, are states negligibly close to $|x_0, y_0\rangle$. However, these states will almost invariably fail $A_1$'s distinguishing test, because $A_1$ is—under the assumption that we have already boosted its success probability to some $1-\gamma$—able to tell, with probability $1-\gamma$, the difference between any $|x_0, y_0\rangle$ and the state $|\psi_{y_0}\rangle$. The adversary's only alternative is to forge a signature on $(y_0, y_1)$, but his succeeding would violate the security of $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$, which we assume is impossible. Therefore, there is no adversary which can produce two signatures that pass $\mathsf{verify}$, having seen only one token—except with negligible probability.

*Remarks.*

- The part of this scheme which is applicable to public-key quantum money is no different from Zhandry's construction of [Zha17]. We have essentially produced an extension of that construction which happens to be a tokens scheme.

- One curious property of this NCH-based construction for tokens is that the signatures $\sigma$ it produces are quantum, while [BS16]'s signatures were classical. In consequence, our construction loses some of the properties which [BS16] considered desirable, such as the ability to convert quantum money into 'classical cheques' which could be sent over classical channels. The sacrifice is not a fruitless one, however: since our scheme is based on non-collapsing hash functions, we also acquire the properties of 'collision-free quantum money' which [Zha17] considered desirable, such as the ability to ensure that even the bank cannot forge quantum money (or signing tokens).

# 4 NCH from iO

## 4.1 Definition of NCH function

Let $k, n, m$ be positive integers linear in $\lambda$, with $k < n < m$. Let $q$ be a positive even integer which satisfies the following two conditions.

- q is superpolynomial in $\lambda$.
- Let the bound constant $B_P$ be equal to $\frac{q}{4C_T\sqrt{mn\log q}}$ (where $C_T$ is a universal constant). Let $B_V$ be a bound such that
  - $B_V < B_P$;
  - the ratio $\frac{B_P}{B_V}$ is superpolynomial in $\lambda$.

  We require $q$ to be large enough so that $2\sqrt{n} < B_V < B_P$.

Let $L$ be a map from $\mathbb{F}_2$ to $\mathbb{Z}_q$ such that

$$L(b) = \begin{cases} 0 & b = 0 \\ \frac{q}{2} & b = 1 \end{cases}. \tag{5}$$

When we write $L(x)$, where $x$ is a vector with entries in $\mathbb{F}_2$, or $L(M)$, where $M$ is a matrix with entries in $\mathbb{F}_2$, we mean the entry-wise application of $L$ to $x$ or to $M$. Note that $L$, in all its forms, is injective.

The non-collapsing hash function is defined as follows:

- $\mathsf{key\text{-}gen}(1^\lambda)$: Generates the following constants which define the hash function.
  - An $m \times n$ LWE matrix, $A$, chosen uniformly at random, whose entries are in $\mathbb{Z}_q$;
  - $k$ $n$-bit binary strings $s_1, \ldots, s_k$, chosen uniformly at random, which represent $k$ secrets.

  The hash function key that $\mathsf{key\text{-}gen}$ outputs consists of the following:
  - The matrix $A$.
  - $k$ LWE samples $A\big(L(s_i)\big) + e_i$, for $i$ in $1, \ldots, k$. $e_i$ are randomly chosen error vectors in $\mathbb{Z}_q^m$ whose magnitudes satisfy $||e_i|| < \frac{\sqrt{m}}{k} B_V$.
  - A program $P$. Let $\mathcal{S}$ denote the $k \times n$ matrix over $\mathbb{F}_2$ whose rows are $s_1, \ldots, s_k$. $P(\cdot)$ is the obfuscation under $\mathsf{shO}$ of the subspace membership oracle for the subspace $\ker(I_k \mid \mathcal{S})$, where $I_k$ is the $k \times k$ identity matrix, and $I_k \mid \mathcal{S}$ is the $k \times (n + k)$ block matrix obtained by joining $I_k$ and $\mathcal{S}$ together, with $I_k$ on the left and $\mathcal{S}$ on the right. For convenience, we denote this subspace $(\ker(I_k \mid \mathcal{S}))$ by $S_0$.
- Evaluation: Let $s_b = \bigoplus_i b_i s_i$, where $b_i$ denotes the $i$th bit of $b$. Similarly, let $e_b = \sum_i b_i e_i$. (The sum in the latter uses addition modulo $q$.)

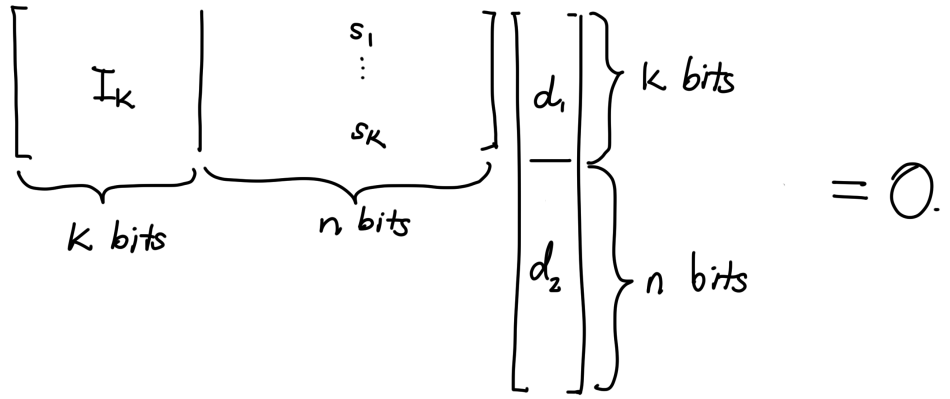
  Let $(b, x, e)$ be in $\mathbb{F}_2^k \times \mathbb{F}_2^n \times \mathcal{E}$, where $\mathcal{E} = \big\{e \in \mathbb{Z}_q^m \;\big|\; ||e|| \leq \sqrt{m} B_P\big\}$. Define $F : \mathbb{F}_2^k \times \mathbb{F}_2^n \times \mathcal{E} \to \mathbb{Z}_q^m$ by

$$F(b, x, e) = A\big(L(x)\big) + e + A\big(L(s_b)\big) + e_b. \tag{6}$$

  $F$ can be evaluated publicly using the hash function key: the user computes $A\big(L(x)\big) + e$ for himself, and then, for $i = 1, \ldots, k$, adds the sample $A\big(L(s_i)\big) + e_i$ into the output register (modulo $q$) iff the $i$th bit of $b$ is 1.

### 4.2 Non-collapsing

$P$ implements a subspace membership program which takes in $d$ $(= d_1||d_2)$ as argument and checks whether the following matrix equation holds:




Note that this is equivalent to checking that, for all $i$, $d_2 \cdot s_i = d_{1,i}$ (where $d_{1,i}$ represents the $i$th bit of $d_1$).

We now prove that $F$ is non-collapsing, viz., there is an adversary $A$ (consisting of two phases, $A_0$ and $A_1$) who wins the collapsing game with non-negligible probability. When $A_0$ is called upon to provide a superposition over the message registers, it prepares the superposition

$$q^{-\frac{n}{2}} \sum_{\substack{b\in\{0,1\}^k \\ x\in\{0,1\}^n \\ e\in\mathbb{Z}_q^m}} \sqrt{D_{\mathbb{Z}_q^m,B_P}(e)}\,|b,x,e\rangle \tag{7}$$

(This superposition can be produced efficiently by creating the superposition $q^{-\frac{n}{2}} \sum_{e\in\mathbb{Z}_q^m} \sqrt{D_{\mathbb{Z}_q^m,B_P}(e)}\,|e\rangle$—which is possible by Lemma 3.12 of [Reg05]—and adding uniform superpositions over all $b$ and all $x$.)

The challenger evaluates $F$ on this superposition, and obtains the state

$$q^{-\frac{n}{2}} \sum_{\substack{b\in\{0,1\}^k \\ x\in\{0,1\}^n \\ e\in\mathbb{Z}_q^m}} \sqrt{D_{\mathbb{Z}_q^m,B_P}(e)}\,|b,x,e\rangle\,\big|A\big(L(x)\big) + e + A\big(L(s_b)\big) + e_b\big\rangle\,. \tag{8}$$

The challenger then measures either the entire state or else the output register only, and returns one of the two following states to $A_1$:

- $|\psi_{\text{meas}}\rangle = |b,x,e\rangle\,|F(b,x,e)\rangle$ for some $(b,x,e)$.
- $|\psi_y\rangle$ for some $y$ in the image of $F$. We claim that $|\psi_y\rangle$ is negligibly close to the state

$$\alpha \sum_b |b, x^* \oplus s_b, e^* - e_b\rangle\,|y\rangle \tag{9}$$

for some global normalisation factor $\alpha$, and for some $(x^* \in \mathbb{F}_2^n, e^* \in \mathbb{Z}_q^m)$. To show that $|\psi_y\rangle$ has this form, we prove lemmas 4.1 and 4.2.

**Lemma 4.1.** *Fix a $y$ in the image of $F$. For each $b \in \{0,1\}^k$, there is exactly one $(x', e')$ with $x' \in \mathbb{F}_2^n, e' \in \mathbb{Z}_q^m$ such that $F(b,x',e') = y$, and it has the form $(x' = x^* \oplus s_b, e' = e^* - e_b)$, where $s_b = \bigoplus_i b_i s_i$, $e_b = \sum_i b_i e_i$, and $(x^*, e^*)$ is the unique solution to the equation $A\big(L(x^*)\big) + e^* = y$.*

*Proof.* Suppose that a $y$ has been measured. Note that this $y$ must be in the form $A\big(L(x^*)\big) + e^*$ for some $x^* \in \mathbb{F}_2^n$, and some $e^* \in \mathbb{Z}_q^m$ with magnitude $||e^*|| \leq \frac{q}{2C_T\sqrt{n\log q}}$. We recall that $L$ is injective, apply the inversion theorem (Theorem 5.1 of [MP11]), and conclude that $(x^*, e^*)$ is unique. Given that this is so, the task of finding collisions becomes that of finding solutions $(b, x', e')$ to the following equation:

$$A\big(L(x^*)\big) + e^* = A\big(L(x')\big) + e' + A\big(L(s_b)\big) + e_b. \tag{10}$$

The equation can be rearranged to yield:

$$A\big(L(x')\big) + e' = A\big(L(x^*)\big) + e^* - \Big(A\big(L(s_b)\big) + e_b\Big). \tag{11}$$

$$\implies A\big(L(x')\big) + e' = A\big(L(x^* \oplus s_b)\big) + e^* - e_b. \tag{12}$$

Note that $||e^* - e_b|| \leq \frac{3}{4}\frac{q}{C_T\sqrt{n\log q}}$. Applying the inversion theorem once again, we conclude that

$$L(x') = L(x^* \oplus s_b) \tag{13}$$

$$\implies x' = x^* \oplus s_b; \tag{14}$$

$$e' = e^* - e_b. \tag{15}$$

For each $b \in \{0,1\}^k$, there will be exactly one $(x', e')$ such that the pair of equations above are satisfied, and $x'$ and $e'$ have the form we claimed they would have. The lemma is thus proved. □

**Lemma 4.2.** *$|\psi_y\rangle$ is negligibly close in trace distance to a state of the form*

$$\alpha \sum_b |b, x^* \oplus s_b, e^* - e_b\rangle\,|y\rangle\,, \tag{16}$$

*where $\alpha$ is a global normalisation factor.*

*Proof.* Given lemma 4.1, it is clear that (up to normalisation)

$$|\psi_y\rangle = \sum_b \sqrt{D_{\mathbb{Z}_q^m, B_P}(e^* - e_b)}\,|b, x^* \oplus s_b, e^* - e_b\rangle\,|y\rangle\,. \tag{17}$$

If we require that the maximum amplitude of $e_b$ is small compared to the Gaussian parameter of the distribution $D_{\mathbb{Z}_q^m, B_P}$, $\sqrt{D_{\mathbb{Z}_q^m, B_P}(e^* - e_b)}$ should be approximately the same for all $b$. More precisely, Lemma 34 in [BCM+18] guarantees that, if we choose $e_i$ so that $||e_b|| \leq \sqrt{m}B_V$ for all $e_b$—and $B_V$ is a bound such that the ratio $\frac{B_P}{B_V}$ is superpolynomial in $\lambda$—we have

$$H^2\big(D_{\mathbb{Z}_q^m, B_P}(e^*), D_{\mathbb{Z}_q^m, B_P}(e^* - e_b)\big) \leq 1 - e^{-\frac{2\pi m B_V}{B_P}}, \tag{18}$$

where $H^2$ is the Hellinger distance. Note that the right-hand-side is negligible in $\lambda$. It follows that the trace distance between the state $|\psi_y\rangle$ and the state

$$\sum_b \sqrt{D_{\mathbb{Z}_q^m, B_P}(e^*)}\,|b, x^* \oplus s_b, e^* - e_b\rangle\,|y\rangle \tag{19}$$

is negligible in $\lambda$. In other words,

$$|\psi_y\rangle \approx \alpha \sum_b |b, x^* \oplus s_b, e^* - e_b\rangle\,|y\rangle\,. \tag{20}$$

□

In the following analysis, therefore, we assume $|\psi_y\rangle$ is of the above form.
In order to tell whether it has been given $|\psi_{\text{meas}}\rangle$ or $|\psi_y\rangle$, $A_1$ performs the following procedure.

- Uncompute the register containing the errors. This is possible because, for any $(b, x, e, y)$, $e = y - A\big(L(x)\big) - \big(A\big(L(s_b)\big) + e_b\big)$—so that, using the hash function key, the value of $e$ for any member of the superposition can be computed from its corresponding values of $y$, $x$ and $b$.

- Apply the Hadamard transform to the remaining registers. In the case where $A_1$ was given $|\psi_y\rangle$, this results (up to normalisation) in the following state:

$$\sum_d \left( \sum_b (-1)^{d\cdot(b \,||\, x^* \oplus s_b)} \right) |d\rangle \tag{21}$$

  Not all $d \in \{0,1\}^{n+k}$ will be supported in this superposition (and we will make precise which ones are). In the case where $A_1$ was given $|\psi_{\text{meas}}\rangle$, however, all $d$s in $\mathbb{F}_2^{n+k}$ will appear in the superposition, and the magnitude of each $d$'s amplitude will be the same.

  **Lemma 4.3.** *An $(n+k)$-bit string $d$ is supported in the superposition $H^{\otimes(n+k)} |\psi_y\rangle$ if and only if $P(d) = 1$.*

  *Proof.* Let $d_1$ be the first $k$ bits of $d$, and let $d_2$ be the last $n$ bits of $d$. Note that, in the $|\psi_y\rangle$ case, we can rewrite the amplitude of each $d$ in the following way:

$$\begin{aligned}
\sum_b (-1)^{d\cdot(b \,||\, x^* \oplus s_b)} &= \sum_b (-1)^{d_1 \cdot b} (-1)^{d_2 \cdot (x^* \oplus s_b)} && (22)\\
&= (-1)^{d_2 \cdot x^*} \sum_b (-1)^{d_1 \cdot b} (-1)^{d_2 \cdot \left((x^* \oplus s_b) \oplus x^*\right)} && (23)\\
&= (-1)^{d_2 \cdot x^*} \sum_b (-1)^{d_1 \cdot b} (-1)^{d_2 \cdot s_b} && (24)\\
&= (-1)^{d_2 \cdot x^*} \sum_b (-1)^{\sum_i b_i \cdot (d_{1,i} \oplus d_2 \cdot s_i)} && (25)\\
&= (-1)^{d_2 \cdot x^*} \prod_{i=1}^{n} \left(1 + (-1)^{d_{1,i} \oplus d_{2,i} \cdot s_i}\right) && (26)
\end{aligned}$$

  If, for any $i$, $d_2 \cdot s_i \oplus d_{1,i} = 1$, then one of the terms in the product in (26) will be zero, and the $d$ under consideration will vanish from the superposition $H^{\otimes(n+k)} |\psi_y\rangle$. In other words, the amplitude of a given string $d$ is *nonzero* if and only if $d_2 \cdot s_i = d_{1,i}$ for all $i$. Recall that $P$ tests exactly this condition for the $d$ that it is given as input. Therefore, the only $d$s that will be supported in the superposition $H^{\otimes(n+k)} |\psi_y\rangle$ are those such that $P(d) = 1$. □

- Apply $P$ to the superposition of $d$s and measure the result. If $A_1$ was given $|\psi_y\rangle$, this measurement will yield 1 with probability 1. If $A_0$ was given $|\psi_{\text{meas}}\rangle$, the probability that this measurement yields 1 is negligible, because the probability $Pr\left[d \in S_0 \;\middle|\; d \xleftarrow{r} \mathbb{F}_2^{n+k}\right]$ is exponentially small if $k$ is linear in $n$. This completes the proof that $A$ wins the collapsing game with non-negligible probability $1 - \mathsf{negl}(n)$.

## 4.3 Collision resistance

**Lemma 4.4.** *Let $\ell$ be a positive integer, and let $n$ and $k$ be the same as they were in section 4.1. Under the following conditions on $n, k, \ell$, $F$ is collision resistant.*

1. *$n, k, \ell$ are linear in $\lambda$,*
2. *$0 < \ell < k < n$,*
3. *$\frac{k}{\ell} \leq \frac{n}{k}$,*

4. $k\ell > 2$.

*Proof.* Note that, if an adversary can recover $\big((b,x,e),(b',x',e')\big)$ such that $F(b,x,e) = F(b',x',e')$, he can evaluate $x \oplus x'$ and recover $s_{\hat{b}}$ for some $\hat{b}$. Therefore, in order to prove that it is impossible to find collisions for $F$, we prove that it is impossible to recover any $s_{\hat{b}} = \bigoplus_i \hat{b}_i s_i$ given the hash function key. We do so through a sequence of hybrids.

- $H_0$: The challenger outputs the key as it is defined in section 4.1.
- $H_1$: Relying upon the security of $\mathsf{shO}$, the challenger swaps $P$—which checks membership in the $n$-dimensional subspace $S_0$—for another subspace membership program that instead checks membership in a random, $(n+\ell)$-dimensional subspace $S_1 \supset S_0$ (with $0 < \ell < k$). $H_0$ and $H_1$ are computationally indistinguishable by the security of $\mathsf{shO}$.
- $H_2$: The challenger gives out a full description of $S_1$, instead of a membership program. Any adversary who can break collision resistance in $H_1$ can also do so in $H_2$.
- $H_3$: The challenger outputs $k$ uniformly random strings instead of the samples $A\big(L(s_i)\big) + e_i$. It also outputs $S_1$ and $A$ as before.

  We show that this hybrid and the last are indistinguishable by reducing the problem of telling them apart to the problem of decisional LWE with secrets drawn from any distribution $\mathcal{D}$ that has sufficient entropy ([GKPV10]). Following [GKPV10], we denote this problem by $\mathsf{DLWE}(\mathcal{D})$.

  Let $C$ be the uniform distribution over the set $\big\{\mathcal{S}' \;\big|\; \ker(I_k \,|\, \mathcal{S}') \in S_1, \mathcal{S}' \in \mathbb{F}_2^{nk}\big\}$. We prove lemma 4.5 in order to show that $C$ has sufficient min-entropy to serve as the distribution $\mathcal{D}$ in the $\mathsf{DLWE}(\mathcal{D})$ problem.

  **Lemma 4.5.** *Let $S_1$ be $(n+\ell)$ dimensional, where $0 < \ell < k$. Under the following conditions on $n$, $k$ and $\ell$, there are at least exponentially many items in the set $\big\{\mathcal{S}' \;\big|\; \ker(I_k \,|\, \mathcal{S}') \in S_1, \mathcal{S}' \in \mathbb{F}_2^{nk}\big\}$.*

  1. *$n, k, \ell$ are linear in $\lambda$,*
  2. *$0 < \ell < k < n$,*
  3. *$\frac{k}{\ell} \leq \frac{n}{k}$,*
  4. *$k\ell > 2$.*

  *Proof.* Let $\mathcal{M} = \{M\}$ denote the set of all $k \times (n+k)$ matrices whose

  1. first $k$ columns are linearly independent;
  2. kernels are in $S_1$.

  (Notice that the first condition is equivalent to requiring that all $M$s in $\mathcal{M}$ row reduce to $I_k \,|\, \mathcal{S}'$, for some $\mathcal{S}'$.)

  Recall how we generated $S_1$: we set $S_0$ to be the kernel of a matrix $I_k \,|\, \mathcal{S}$, and chose $S_1$ to be a random subspace which contained $S_0$. As such, $S_1^\perp \subset \text{rowspan}(I_k \,|\, \mathcal{S})$. In other words, some linear combination of the rows of $(I_k \,|\, \mathcal{S})$ constitutes a basis for $S_1^\perp$.

  The rows of $(I_k \,|\, \mathcal{S})$ are of the form $(e_i, s_i)$ for $i$ in $1, \ldots, k$, where each $e_i$ is a standard basis vector that is $k$ bits long, and each $s_i$ is a random secret in $\mathbb{F}_2^n$. Suppose that $B = \big\{(l_j, r_j) \;\big|\; l_j \in \mathbb{F}_2^k, r_j \in \mathbb{F}_2^n; j \in 1, \ldots, k-\ell\big\}$ is a basis for $S_1^\perp$, where each basis vector $(l_j, r_j)$ is a linear combination of rows of $(I_k \,|\, \mathcal{S})$.

  **Lemma 4.6.** *If the set $B$ is linearly independent, then the set of all $l_j, j \in 1, \ldots, k-\ell$, is linearly independent.*

*Proof.* We prove the contrapositive: if there is some $l_{j^*}$ that is a linear combination of $l_{j_1}, \ldots, l_{j_p}$, then $(l_{j^*}, r_{j^*}) = (l_{j_1}, r_{j_1}) \oplus \cdots \oplus (l_{j_p}, r_{j_p})$.

Suppose such an $l_{j^*}$ exists. Then $r_{j^*}$ will have the following form:

$$r_{j^*} = (l_{j_1} \oplus \cdots \oplus l_{j_p})_1 \cdot s_1 \oplus \cdots \oplus (l_{j_1} \oplus \cdots \oplus l_{j_p})_k \cdot s_k. \tag{27}$$

(Here, $(l_{j_1} \oplus \cdots \oplus l_{j_p})_1$ denotes the first bit of $l_{j_1} \oplus \cdots \oplus l_{j_p}$.)

Expanding, we have that

$$\begin{aligned} r_{j^*} &= \big((l_{j_1})_1 \oplus \cdots \oplus (l_{j_p})_1\big) \cdot s_1 \oplus \cdots \oplus \big((l_{j_1})_k \oplus \cdots \oplus (l_{j_p})_k\big) \cdot s_k && (28) \\ &= (l_{j_1})_1 \cdot s_1 \oplus (l_{j_1})_2 \cdot s_2 \oplus \cdots \oplus (l_{j_1})_k \cdot s_k \\ &\quad \oplus \\ &\quad \vdots && (29) \\ &\quad \oplus \\ &\quad (l_{j_p})_1 \cdot s_1 \oplus (l_{j_p})_2 \cdot s_2 \oplus \cdots \oplus (l_{j_p})_k \cdot s_k \\ &= r_{j_1} \oplus \cdots \oplus r_{j_p}. && (30) \end{aligned}$$

As such, $(l_{j^*}, r_{j^*}) = (l_{j_1}, r_{j_1}) \oplus \cdots \oplus (l_{j_p}, r_{j_p})$. The lemma is thus proved. □

Recall that $B$ is the basis which we choose for $S_1^\perp$ from the rows of $(I_k \mid \mathcal{S})$. Suppose we write $B$ as a matrix $M'$ by using the elements of $B$ as the rows of $M'$. If we write $M'$ as a block matrix $(L' \mid R')$, where $L$ is a $(k-\ell) \times k$ matrix and $R$ is a $(k-\ell) \times n$ matrix, then we know by lemma 4.6 that $L'$ will be full-rank.

To extend $M'$ to a matrix $M \in \mathcal{M}$, we fill the remaining $\ell$ rows uniformly at random. Notice that, if we write $M$ as a block matrix $(L \mid R)$, where $L$ is $k \times k$ and $R$ is $k \times n$, $M$ will be in $\mathcal{M}$ if and only if $L$ is full-rank. The chances that the last $\ell$ random vectors in $L$ will be linearly independent of the first $k-\ell$ which we chose earlier (in the form of $L'$) can be expressed in the following manner:

$$\begin{aligned} &\frac{1}{2^{k\ell}} \prod_{i=k-\ell+1}^{k} (2^k - 2^{i-1}) && (31) \\ &= \prod_{i=k-\ell+1}^{k} (1 - (\tfrac{1}{2})^{k+1-i}) && (32) \\ &= \prod_{i=1}^{\ell} (1 - (\tfrac{1}{2})^{2k-\ell+1-i}) && (33) \\ &\geq \prod_{i=1}^{\ell} (1 - (\tfrac{1}{2})^{\ell+1-i}) && (34) \end{aligned}$$

It is known that eq. (34) converges to a constant $> \frac{1}{4}$ as $\ell \to \infty$; as such, eq. (34) is bounded below by $\frac{1}{4}$ (and above by 1). There are $2^{\ell(n+k)}$ different ways to fill the last $\ell$ rows of $M$, and $> \frac{1}{4}$ of these will result in an $M$ whose first $k$ columns are linearly independent. Therefore, the set $\mathcal{M}$ (of $M$s whose first $k$ columns are linearly independent, and whose kernels are in $S_1$) must contain at least $2^{\ell(n+k)-2}$ elements.

Denote the reduced row echelon form of $M \in \mathcal{M}$ by $\hat{M}$. Note that there is a bijection between the set of all $\hat{M}$s and the set $\left\{\mathcal{S}' \,\middle|\, \ker(I_k \,|\, \mathcal{S}') \in S_1, \mathcal{S}' \in \mathbb{F}_2^{nk}\right\}$. The map from $M$ to $\mathrm{RREF}(M)$ is many-to-one, but, for any given $\hat{M}$, the maximum number of matrices $M$ which row reduce to $\hat{M}$ is given by

$$(2^k - 1) \cdots (2^k - 2^{k-1}) \leq 2^{k^2}. \tag{35}$$

The left-hand-side of eq. (35) is an expression for the number of ways one can choose a basis for a $k$-dimensional space over $\mathbb{F}_2$. Since matrices which have the same RREF always have the same kernel, their rows can be interpreted as different choices of basis for the perp space of that kernel.

Given eq. (35), and that there are at least $2^{\ell(n+k)-2}$ $M$s, the number of $\hat{M}$s must be at least $2^{\ell(n+k)-k^2-2}$. For any choice of $n, k, \ell$ such that

1. $n, k, \ell$ are linear in $\lambda$,
2. $0 < \ell < k < n$,
3. $\frac{k}{\ell} \leq \frac{n}{k}$,
4. $k\ell > 2$.

the set of all $\hat{M}$s will contain exponentially many elements. Since there is a bijection between the set of all $\hat{M}$s and the set $\left\{\mathcal{S}' \;\middle|\; \ker(I_k \mid \mathcal{S}') \in S_1, \mathcal{S}' \in \mathbb{F}_2^{nk}\right\}$, lemma 4.5 is true. □

By the $\mathsf{DLWE}(\mathcal{D})$ assumption, where $\mathcal{D}$ is the uniform distribution over the set $\{L(\mathcal{S}'^T) \mid \mathcal{S}' \leftarrow C\}$, no adversary can tell the difference between the samples $A\big(L(s_i)\big) + e_i$ and $k$ random strings. Suppose there were an adversary $A$ who could. We enlist $A$ to build an adversary $B$ who breaks $\mathsf{DLWE}(\mathcal{D})$. $B$ receives a random LWE sample $\Big(A, A\big(L(\mathcal{S}'^T)\big) + E\Big)$, where $A$ is an $m \times n$ matrix, $L(\mathcal{S}'^T)$ is an $n \times k$ matrix, and $E$ is an $m \times k$ matrix; $B$ also knows the distribution $\mathcal{D}$, which means that it knows $S_1$ and can provide $A$ with a description of it. Notice that the columns of $B$'s sample constitute $k$ vectors of the form $A\big(L(s_i)\big) + e_i$. Therefore, $B$ gives its sample to $A$ column-wise, and outputs whatever $A$ outputs. If $A$ can distinguish between $H_3$ and $H_2$, then $B$ can break $\mathsf{DLWE}(\mathcal{D})$. We conclude that $H_3$ and $H_2$ are indistinguishable.

- $H_4$: The challenger simply outputs $S_1$. It is impossible to recover $\mathrm{span}(s_1, \ldots, s_k)$—or any member of that subspace—with non-negligible probability from only $S_1$, provided that the dimension gap $d_1 - d_0$ (where $d_1$ is the dimension of $S_1$, and $d_0$ is the dimension of $S_0$) is linear in $n$: there are exponentially many equally probable possibilities, and the adversary has no way of distinguishing between them. This completes our proof that $F$ is collision resistant.

□

# References


[BCM+18] Z. Brakerski, P. Christiano, U. Mahadev, U. Vazirani, and T. Vidick. Certifiable Randomness from a Single Quantum Device. *ArXiv e-prints*, April 2018.

[BS16] S. Ben-David and O. Sattath. Quantum Tokens for Digital Signatures. *ArXiv e-prints*, September 2016.

[BZ13] Dan Boneh and Mark Zhandry. Secure signatures and chosen ciphertext security in a quantum computing world. In *Proc. of Crypto*, volume 8043 of *LNCS*, pages 361–379, 2013.

[GKPV10] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Robustness of the learning with errors assumption. In *ICS*, 2010.

[GYZ17] Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 342–371, Cham, 2017. Springer International Publishing.

[MP11] Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. Cryptology ePrint Archive, Report 2011/501, 2011. https://eprint.iacr.org/2011/501.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.

[Zha17] M. Zhandry. Quantum Lightning Never Strikes the Same State Twice. *ArXiv e-prints*, November 2017.