Short Paper: Breaking X-VRF, a Post-Quantum Verifiable Random Function

Omid Bodaghi[1] and Reihaneh Safavi-Naini[1]

University of Calgary, Alberta, Canada
{omid.bodaghi,rei}@ucalgary.ca

Abstract. A Verifiable Random Function (VRF) is a public key cryptographic system with a public and private key pair (SK, PK), that for an input value x uses SK to generate a "pseudorandom" value v together with a proof π, that proves correctness of computation using PK. VRFs must satisfy *uniqueness of the output* which requires that for any public key PK and any input value x, a unique valid pair (v, π) (acceptable by the verification algorithm using PK), can be constructed.
In this paper, we present an explicit deterministic attack on the uniqueness property of X-VRF, a recently proposed post-quantum secure VRF that was designed for Algorand's sortition algorithm. The attack allows a malicious user to increase their chance of success in being selected to propose a block in a given round.

Keywords: Post-Quantum Verifiable Random Function · X-VRF · Hash-based Signatures · WOTS$^+$ one-time signature

1 Introduction

Verifiable Random Functions (VRFs) were first introduced in [9] to address the challenge of generating "pseudorandom" values without the need to trust the generation process. VRFs are public key cryptographic primitives with a pair of secret and public key (SK, PK), that provide public verifiability for the function output. For a secret key SK, the function value on an an input x can be efficiently calculated, resulting in a pair (v, π), where v is the function value, and π is a *proof* that can be used to prove correctness of v with respect to PK. An important property of VRF is the uniqueness of the output, ensuring that even with the knowledge of SK, it is not possible to produce two valid evaluations of the function for a single input. That is for any public key PK, and any input x, there is a unique (v, π) that satisfies the verification algorithm.

VRFs have found wide application in Proof of Stake (PoS) blockchains for efficiently achieving consensus. As an example, in each round of Algorand, a user uses a VRF to participate in a "lottery" that determines if they are part of the committee that forms the block in that round [4]. Other important applications of VRF are in key transparency [7] and DNS security [11].

Existing VRFs, including those used in PoS systems, rely on computational hardness of problems such as Integer Factorization and Discrete Logarithm,

which can be efficiently solved by quantum algorithms [12]. X-VRF [2] is the first post-quantum VRF that can be used for many evaluations[1].

X-VRF uses XMSS, a stateful post-quantum hash-based signature scheme that is recently standardized by NIST [10], to construct a Verifiable Unpredictable Function (VUF)(also known as *unique signature [Footnote 5, [9]]*), and then uses a random oracle (RO) to obtain a pseudorandom output in RO model. VUFs are the same as VRFs with the difference that their output is *unpredictable*, instead of being pseudorandom. The uniqueness of X-VRF employs the state of a blockchain as a counter for XMSS, to determine the *state (index)* of the signature to be used for computing the function output. In other words X-VRF requires a secure blockchain as a building block, and authors considered application of X-VRF in Algorand.

Our work. We show that X-VRF fails to satisfy uniqueness property of VRFs, by presenting a deterministic algorithm that produces two distinct valid (successful verification) VRF outputs for any given input. Thus X-VRF, when used as the VRF in sortition algorithm of Algorand, allows a malicious user to *double* their chance of becoming a committee member. Our main contribution is constructing an efficient (deterministic) algorithm for WOTS$^+$, the one-time-signature (OTS) scheme which is the main building block of XMSS, that generates a malicious public key for any secret key, that allows two valid WOTS$^+$ signatures to be produced for any message. The attack directly extends to an attack on the uniqueness of XMSS and so X-VRF. We note that the attack is not a forgery attack against WOTS$^+$; rather, it is an attack against *uniqueness of WOTS$^+$ when used as a (one-time) VUF*.

Our attack exploits collisions in the hash function that is used in WOTS$^+$. Our attack assumes there exist at least one function (from a family of functions) with a *known collision* to an adversary. In practice the function family is realized by using SHA-3 for which *no collision is known* and so the attack cannot be launched until a collision can be found in SHA-3. Thus the attack is primarily on the theoretical security proof of X-VRF (computational uniqueness) that does not exclude existence of such function. The attack points out possibility of maliciously generating keys for WOTS$^+$ and XMSS, that are important (standardised) signatures, in applications that require uniqueness property for signatures (see Definition 1).

Organization. Section 2 provides background and reviews the construction of WOTS$^+$. Section 3 describes malicious key generation for WOTS$^+$ and shows that it does not provide uniqueness property, and extend the results to XMSS and finally breaking X-VRF uniqueness property. Section 4 concludes the paper.

2 Background

2.1 Hash-based Signatures

An important class of post-quantum signature schemes are *hash-based signature schemes* that do not use any hardness assumption. The Security of hash-

[1] The first post-quantum VRF in [3] can only be used few times for a key pair.

based signatures relies on the minimal assumption of the existence of One-Way-Functions (OWFs).

Hash-based signatures were introduced by Lamport [6] as a One-Time Signature (OTS) for signing a single message. The scheme was later extended to a digital signature scheme named MSS for signing 2^N messages by using a Merkle hash tree that combined the public keys of 2^N OTSs into a single hash value that serves as the public key of the scheme [8]. Security of hash-based signatures initially required collision-freeness of the hash function. This requirement was later reduced to second-preimage resistance for one-time signatures in WOTS$^+$ [5] and for many-time signatures in XMSS [1]. In contrast to MSS, XMSS employs WOTS$^+$ as an OTS scheme and introduces randomization elements to eliminate the need for assuming the presence of a collision-resistant hash function.

This section includes the construction of WOTS$^+$ as explained in [5], while the explanation of XMSS is omitted due to space constraints.

WOTS$^+$ Construction [5] Consider n as security parameter, m as message length, and w as Winternitz parameter. Set $l_1 = \lceil \frac{m}{logw} \rceil$, $l_2 = \lfloor \frac{log(l_1(w-1))}{logw} \rfloor$, and $l = l_1 + l_2$.

WOTS$^+$ uses a family of *one-way, second pre-image, and undetectable* functions $\mathcal{F}_n : \{f_k : \{0,1\}^n \to \{0,1\}^n | k \in \mathcal{K}_n\}$ where $\mathcal{K}_n$ denotes the key space.

Define the chaining function $c_k^i(x, \mathbf{r})$ as:

$$c_k^i(x, \mathbf{r}) = \begin{cases} f_k(c_k^{i-1}(x, \mathbf{r}) \oplus r_i) & \text{if } i > 0 \\ c_k^0(x, \mathbf{r}) = x & \text{if } i = 0 \end{cases}$$

Where $x \in \{0,1\}^n$ is the input, $\mathbf{r} = (r_1, ..., r_j) \in \{0,1\}^{n \times j}$ with $j \geq i$ are the randomization elements, $i \in N$ is the iteration number (N represents the set of natural numbers), and $k \in \mathcal{K}_n$ is the key. $c_k^{i-1}(x, \mathbf{r})$ represents the chaining function at the previous iteration, r_i is the i-th element of the randomization elements $\mathbf{r}$, and $\oplus$ denotes bitwise XOR.

Key Generation ($(PK, SK) \leftarrow Kg(1^n)$): For security parameter n:

- Choose l n-bit strings uniformly at random as sk_i and set $SK = (sk_1, sk_2, ..., sk_l)$.
- Choose $w-1$ n-bit strings uniformly at random as r_i and set $\mathbf{r} = (r_1, r_2, ..., r_{w-1})$.
- Choose k uniformly at random from $\mathcal{K}_n$.
- Set $PK = (pk_0, pk_1, ..., pk_l) = ((\mathbf{r}, k), c_k^{w-1}(sk_1, \mathbf{r}), ..., c_k^{w-1}(sk_l, \mathbf{r}))$
- Return (PK, SK)

Signature Algorithm ($\sigma \leftarrow Sign(M, SK, \mathbf{r})$): On input of a m-bit message M, secret key SK, and randomization elements $\mathbf{r}$:

- Compute base w representation of $M = (M_1, ..., M_{l_1})$.
- Compute checksum $C = \sum_{i=1}^{l_1} (w - 1 - M_i)$ and its base w representation $C = (C_1, ..., C_{l_2})$.

- Set $B = M||C = (b_1, ..., b_l)$.
- Return the signature $\sigma = (\sigma_1, ..., \sigma_l) = (c_k^{b_1}(sk_1, \mathbf{r}), ..., c_k^{b_l}(sk_l, \mathbf{r}))$.

Verification ($Accept/Reject \leftarrow Ver(1^n, M, \sigma, PK)$): On input of security parameter, message M, signature σ, and public key PK:

- Verifier first computes B as explained above.
- Verifier validates:
  $PK = (pk_0, ..., pk_l) \stackrel{?}{=} ((\mathbf{r}, k), c_k^{w-1-b_1}(\sigma_1, \mathbf{r}_{b_1+1,w-1}), ..., c_k^{w-1-b_l}(\sigma_l, \mathbf{r}_{b_l+1,w-1}))$
  $\mathbf{r}_{a,b}$ denotes the subset $r_a, ..., r_b$ of $\mathbf{r}$.
- If euqality holds, verifier returns Accept, else Reject.

f_k only requires to be *one-way, second pre-image, and undetectable* [theorem 1, [5]]. Security of WOTS$^+$ does not rely on collision-resistance of f_k.

2.2 VUF, VRF and Unique Signatures

Verifiable unpredictable functions, also known as *unique signatures* were introduced in [9] and used to construct verifiable random functions.

Definition 1 (Verifiable Unpredictable Function [9]). *A Verifiable Unpredictable Function (VUF) scheme F consists of three algorithms $(Gen, Eval, Ver)$ as follows:*

1. *$(SK, PK) \leftarrow Gen(1^\lambda)$: On input the security parameter λ, this probabilistic algorithm generates a public key and a secret key. The public key is known to other users, but the private key must be kept private.*
2. *$(v, \pi) \leftarrow Eval(SK, m)$: Given a secret key and a message $m \in \{0,1\}^{a(\lambda)}$, this algorithm calculates an output value $v \in \{0,1\}^{b(\lambda)}$ and a proof π that shows it is calculated correctly (a and b are polynomial functions).*
3. *$Yes/No \leftarrow Ver(PK, m, v, \pi)$: On input of the public key PK, input m, output v, and proof π, this algorithm uses the proof, the input, and the public key to verify whether the output is calculated correctly or not.*

A secure VUF scheme must satisfy the following requirements:

- *Provability*: If $(SK, PK) \leftarrow Gen(1^\lambda)$ and $(v, \pi) \leftarrow Eval(SK, m)$, then $Ver(PK, m, v, \pi) = Yes$.
- *Computational Uniqueness*: No probabilistic polynomial-time (PPT) adversary should be able to output any $v_1, v_2, \pi_1, \pi_2, m, pk$ such that $v_1 \neq v_2$ and $Ver(PK, m, v_1, \pi_1) = Ver(PK, m, v_2, \pi_2) = Yes$ with a probability higher than $negl(\lambda)$.
- *Unpredictability*: Consider a PPT adversary $\mathcal{A}$ playing the following game Exp_{UPR}:
  1. $(SK, PK) \leftarrow Gen(1^\lambda)$
  2. $(x, guess) \leftarrow \mathcal{A}_1^{O_{Eval\_VUF}(SK,.)}(PK)$
  3. $\mathcal{A}$ succeeds if $(guess, .) = Eval(SK, x)$ and x was not queried before.

$O_{Eval\_VUF}(SK, .)$ is an oracle where the adversary can send a polynomial number of inputs to the oracle and receive the VUF output and proof that correspond to each input. We say that a VUF scheme is unpredictable if the adversary wins Exp_{UPR} game with a probability at most $negl(\lambda)$.

We note that computational uniqueness property must be satisfied for *any well-formed* PK, including those that are maliciously generated.

Definition 2 (Verifiable Random Function [9]). *[Verifiable Random Function] A Verifiable Random Function (VRF) scheme F consists of three algorithms* $(Gen, Eval, Ver)$ *in the same way as VUF, except that unpredictability property is replaced with the following property:*

- *Pseudorandomness: Consider a PPT adversary* $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ *playing the following game* Exp_{RND}*:*
  1. $(SK, PK) \leftarrow Gen(1^\lambda)$
  2. $(x, st) \leftarrow \mathcal{A}_1^{O_{Eval\_VRF}(SK,.)}(PK)$
  3. $(v_0, .) \leftarrow Eval(SK, x)$ *and* $v_1 \leftarrow \{0,1\}^{b(\lambda)}$
  4. *Flip a coin* $b \leftarrow \{0,1\}$*, and send* v_b *to the adversary.*
  5. *Adversary outputs* $b' \leftarrow \mathcal{A}_2^{O_{Eval}(SK,.)}(v_b, st)$

  $O_{Eval\_VRF}(SK, .)$ *is an oracle where the adversary can send a polynomial number of inputs to the oracle and receive the VRF output and proof that correspond to each input. The adversary wins the game if* $b' = b$*. We say that a VRF scheme is pseudorandom if the adversary wins* Exp_{RND} *game with a probability at most* $\frac{1}{2} + negl(\lambda)$.

Construction of VRF from VUF in standard model was first proposed in [9] and uses hardcore bit construction that results in a single output bit. This is then extended to multibit VRF by repeated call to the hard core bit construction. Efficient construction of VRF from VUF is in random oracle (RO) model, and is obtained by applying an RO to the output of the VUF.

2.3 X-VRF: A Quantum-resistant VRF based on XMSS

X-VRF [2] is a quantum-resistant VRF scheme that uses XMSS and a blockchain to construct a VUF that has proved verifiability and computational uniqueness, and uses a random oracle to achieve pseudorandomness in the RO model.

XMSS does not satisfy the required uniqueness property of VUFs because every leaf of the Merkle tree (corresponding to a distinct WOTS$^+$ signature) can be used to sign a message and so more than one signature can be generated on a message. To achieve uniqueness in X-VRF, the state of a blockchain is used as a counter that determines the leaf of Merkle tree that must be used for the current block. Authors note that "In particular, the block number of a particular round in the blockchain consensus can serve as a global counter." [2]. X-VRF achieves uniqueness with this restriction, and noting that WOTS$^+$ is a unique signature (Section 1.4, [2]).

Theorem 1. *(X-VRF Security). X-VRF is correct and satisfies the properties of computational uniqueness and pseudorandomness in the random oracle model. In particular, the uniqueness holds in the sense that the same ctr (or leaf index) must be used in VRF_{Eval}.*

3 Breaking X-VRF

In this section, we show that one can maliciously generate the public key of the WOTS$^+$ (and so XMSS) such that two signatures can be generated on the same message, both verifiable under the same public key. Then we show that this breaks the uniqueness property of X-VRF, and consequently rejects Theorem 1. The algorithm exploits existence of a collision in f_k, which is allowable according to the WOTS$^+$ assumptions [Section 3, [5]].

Lemma 1. *Let the function family $\mathcal{F}_n$ in WOTS$^+$ include a function $f_k \in \mathcal{F}_n$ with a known collision. Then for any secret key SK, there is an efficient (deterministic) algorithm that constructs a public key PK that corresponds to SK such that for any message m, one can generate two valid signatures σ_1 and σ_2 that can be correctly verified under PK.*

This implies that WOTS$^+$ does not satisfy uniqueness property of a one-time VUF (i.e. is not a unique signature).

Proof. The proof is constructive. That is, in Algorithm 1, $PKeyGen$ constructs PK for any given SK, and $Sign$ procedure constructs two legitimate signatures, σ and σ', for any message m. σ is generated by WOTS$^+$ sign algorithm and the second signature, σ' only differs from σ in index i. □

We show that both generated signatures, σ and σ', can be successfully verified. σ is generated by WOTS$^+$ signing algorithm and will be verified with respect to the definition of the digital signature scheme [Section 2.1, [5]]. Since σ' differs from σ only at index i, and WOTS$^+$ verification algorithm is component-wise, we need to only consider the verification step that corresponds to index i. The following provides details of the steps of WOTS$^+$ verification algorithm as described in Section 2 (WOTS$^+$ construction). The verification steps are: Step (1): starts the verification process and states the equality that must be checked; Step (2): expands the right-hand side (RHS) of (1). Note that $\mathbf{r}_{b_i+1,w-1}$ is a subvector of $\mathbf{r}$ and includes elements in the range $[b_i+1, w-1]$. For instance, the first element of $\mathbf{r}_{b_i+1,w-1}$ corresponds to $r_{1+(b_i+1)-1} = r_{b_i+1}$; Step (3): expands $c_k^{w-1-b_i-1}(.,.)$, from Step (2). This process continues for $w-1-b_i$ steps. In the final recursion step, we have $c_k^0(\sigma'_i, \mathbf{r}_{b_i+1,w-1}) = \sigma'_i$, and then computation reverses and computes steps backward one after the other until it arrives at $c_k^{w-1-b_i}$.

$$
\begin{aligned}
c_k^{w-1}(sk_i, \mathbf{r}) &\stackrel{?}{=} c_k^{w-1-b_i}(\sigma_i', \mathbf{r}_{b_i+1,w-1}) && (1)\\
c_k^{w-1-b_i}(\sigma_i', \mathbf{r}_{b_i+1,w-1}) &= f_k(c_k^{w-1-b_i-1}(\sigma_i', \mathbf{r}_{b_i+1,w-1}) \oplus r_{(w-1-b_i)+(b_i+1)-1}) \\
&= f_k(c_k^{w-1-b_i-1}(\sigma_i', \mathbf{r}_{b_i+1,w-1}) \oplus r_{w-1}) && (2)\\
c_k^{w-1-b_i-1}(\sigma_i', \mathbf{r}_{b_i+1,w-1}) &= f_k(c_k^{w-1-b_i-2}(\sigma_i', \mathbf{r}_{b_i+1,w-1}) \oplus r_{w-2}) && (3)\\
&\vdots \\
c_k^{w-1-b_i-(w-2-b_i)}(\sigma_i', \mathbf{r}_{b_i+1,w-1}) &= f_k(c_k^0(\sigma_i', \mathbf{r}_{b_i+1,w-1}) \oplus r_{b_i+1}) \\
&= f_k(\sigma_i' \oplus r_{b_i+1}) && (4)
\end{aligned}
$$

Note that $\sigma_i = c_k^{b_i}(sk_i, r)$ and $\sigma_i' = o' \oplus r_{b_i+1}$ ($Sign$ procedure - Algorithm 1). Also, $r_{b_i+1} = o \oplus c_k^{b_i}(sk_i, r)$ (line 5 of $PKeyGen$ procedure, Algorithm 1). Therefore, we have $f_k(\sigma_i \oplus r_{b_i+1}) = f_k(o)$ and $f_k(\sigma_i' \oplus r_{b_i+1}) = f_k(o')$. Since σ is generated using WOTS$^+$ signing algorithm and $f_k(o) = f_k(o')$, σ' will be verified successfully.

We also need to show that the two generated signatures are distinct. It suffices to show that there exists at least one index i such that $\sigma_i \neq \sigma_i'$. Select i as step 5 in $Sign$ procedure in Algorithm 1. Using the $Sign$ and $PKeyGen$ procedures of the algorithm, we have, $\sigma_i = c_k^{b_i}(sk_i, r)$ and $\sigma_i' = o' \oplus r_{b_i+1} = o' \oplus o \oplus c_k^{b_i}(sk_i, r)$. Since $c_k^{b_i}(sk_i, r)$ is common in o and o', we have $\sigma_i \neq \sigma_i'$ as long as $o \neq o'$, which hold due to the assumption. Therefore, σ and σ' differs in at least one index i.

We note that the WOTS$^+$ construction [5] does not require the function family to be collision resistant and so as long as the hash function domain is larger than its range, hash functions in $\mathcal{F}_n$ will have collisions, and the attack requirement is satisfied if there is a hash function in $\mathcal{F}_n$ with a *known collision*. The (computational) uniqueness property of VUF requires that an *efficient adversary* cannot generate a public and secret key pair that results in two correctly verifiable VUF outputs for the same input.

Corollary 1. *XMSS signature scheme does not satisfy uniqueness property as a many-time VUF (i.e. it is not a unique signature).*

Proof. An XMSS signature on a message m consists of a WOTS$^+$ signature, together with an authentication path (AP) that authenticates WOTS$^+$'s public key with respect to the public key of the XMSS. For any message m, a malicious user proposes two distinct XMSS signatures XMSS.σ = (WOTS$^+$.σ, i, AP) and XMSS.σ' = (WOTS$^+$.σ', i, AP). Since WOTS$^+$.σ and WOTS$^+$.σ' share the same public key and are legitimate OTS signatures (Lemma 1), the verification algorithm of XMSS succeeds for both XMSS.σ and XMSS.σ'. □

Proposition 1. *If XMSS signature scheme is not a unique signature, then the uniqueness property of X-VRF will not hold. This is true even if the same WOTS$^+$ (same leaf index) is used in VRF_{Eval}.*

Proof. The X-VRF output is given by $H(XMSS.\sigma, x)$ [Section 4.1, [2]]. Consider a malicious user $\mathcal{A}$ who can output two XMSS signatures, $XMSS.\sigma$ and $XMSS.\sigma'$, for an input x. Then $\mathcal{A}$ can construct two distinct and correctly verifiable VRF outputs, using the same leaf index (same WOTS$^+$ key pair). This breaks the uniqueness property of X-VRF. □

Algorithm 1 (i) Malicious key generation algorithm; (ii) Generating two signatures on the same message

procedure PKeyGen(1^n, SK)
  Input: Security parameter n and WOTS+ secret key SK
  Output: Public key of the WOTS+ scheme
  Select k such that there exist at least values (collisions) o and o' such that $o \neq o'$ and $f_k(o) = f_k(o')$.
  Choose arbitrary i, and to compute $\mathbf{r} = (r_1, ..., r_{w-1})$, set $r_j = o \oplus c_k^{j-1}(sk_i, \mathbf{r})$ for $j \in [1, w-1]$. ▷ For r_i, we need $r_j, 1 \leq j \leq i-1$
  Set $PK = (pk_0, pk_1, ..., pk_l) = ((\mathbf{r}, k), c_k^{w-1}(sk_1, \mathbf{r}), ..., c_k^{w-1}(sk_l, \mathbf{r}))$
  Return PK
end procedure

procedure Sign(SK, M)
  Input: WOTS+ secret key SK and an m-bit message M
  Output: Two distinct signatures σ and σ'
  Calculate $B = (b_1, ..., b_l)$. ▷ B is concatenation of message and checksum.
  Calculate these two signatures with respect to the selected i. (only the i-th part of these two are different):
  - $\sigma = (\sigma_1, ..., \sigma_i, ..., \sigma_l) = (c_k^{b_1}(sk_1, \mathbf{r}), ..., c_k^{b_i}(sk_i, \mathbf{r}), ..., c_k^{b_l}(sk_l, \mathbf{r}))$.
  - $\sigma' = (\sigma_1, ..., \sigma'_i, ..., \sigma_l) = (c_k^{b_1}(sk_1, \mathbf{r}), ..., o' \oplus r_{b_i+1}, ..., c_k^{b_l}(sk_l, \mathbf{r}))$.
  Return (σ, σ')
end procedure

3.1 Consequence for Algorand

In the Algorand blockchain, users are chosen through a sortition algorithm (lottery) to participate in a committee. From this committee, a single user is selected to propose the next block containing transactions.

Each Algorand user commits to their VRF function by publishing their own public key, which is knwon to others. In the creation of each new block, a common random value (seed) is obtained from the previous block. In each block, all users obtain the same seed. Users then compute the VRF output on this random seed with their own secret keys (users commit to their VRFs by publishing the public key in the first place). Users output a tuple (v, π, j), where v represents the VRF output, π denotes the proof, and j is determined by the user's stake in Algorand.

For $0 \leq i \leq j-1$, each user calculates $H(v, i)$ and publishes (v, π, i) which resulted in the highest value. Then, the user with the greatest value among the received published values is selected as the proposer for the next block.

It is important to note:

- Users with more stakes have a higher j, allowing them to compute more hash functions and thereby increasing their chance of publishing the greatest value.
- The value of j for each user can be verified by others, preventing users from manipulating this value.

The uniqueness property ensures that a single input cannot yield more than one VRF output. However, our research reveals that X-VRF fails to satisfy this uniqueness property. This shows that X-VRF is unsuitable for use in Algorand, as malicious users could calculate multiple outputs for a given input, allowing them to selectively choose the output with a higher priority (hash value).

Although it is assumed that malicious users own less than $\frac{1}{3}$ of the stakes, our malicious algorithm (Algorithm 1) increases their chances of being block proposer, without requiring them to increase their stakes.

X-VRF's security is guaranteed based on XMSS security, assuming a second pre-image resistant hash function. This implies that an adversary must find a collision x' for a **random value** x, such that $f(x) = f(x')$. However, our attack demonstrates that even if an adversary discovers a collision for an **arbitrary value** x, it can compromise the security of X-VRF.

Since Algorand uses a hash function that there is not collision pair for that yet, even for arbitrary values, our attack is not applicable to Algorand readily. However, based on our findings, now adversary needs to find a colliding value x' for an arbitrary value x, instead of finding a colliding value x' for a random value x.

4 Concluding Remarks

We presented a deterministic algorithm that compromises the uniqueness property of X-VRF, by breaking the uniqueness property of the XMSS signature, a post-quantum hash-based signature scheme, which is used as a VUF in the construction of X-VRF.

Our attack algorithm generates malicious public keys for WOTS$^+$, the one-time signature scheme that is used in XMSS, resulting in two valid signatures for a single input. This indicates that WOTS$^+$ and XMSS do not meet the security requirements of VUFs (one-time and many-time, respectively).

The attack exploits a *known collision* for a function in the function family that is used in WOTS$^+$, and breaks the theoretical guarantee of WOTS$^+$ and consequently XMSS, as VUFs, as well as X-VRF as a VRF. We note that in practice WOTS$^+$ uses SHA-3 as the chaining function, for which no collision is currently known.

The attack raises interesting research questions. First, the effect of the attack on Algorand sortition algorithm if X-VRF is used as the VRF, requires further analysis. Algorand is shown to be secure as long as at least 2/3 of the stakes are

owned by honest holders, and each node has a chance proportional to its stake to suggest a block. The attack enables malicious users to double their chances of becoming a committee member (or block proposer) without increasing their investment (stake). Analysing the implications of such a doubling on the security of Algorand is an interesting research question.

A second question is construction of a secure many-time VRF using many-time hash-based signature schemes that satisfy the (computational uniqueness) property of VUFs.

References

1. Buchmann, J., Dahmen, E., Hülsing, A.: XMSS - A practical forward secure signature scheme based on minimal security assumptions. In: Yang, B. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7071, pp. 117–129. Springer (2011). https://doi.org/10.1007/978-3-642-25405-5“˙8, https://doi.org/10.1007/978-3-642-25405-5_8
2. Buser, M., Dowsley, R., Esgin, M.F., Kermanshahi, S.K., Kuchta, V., Liu, J.K., Phan, R.C., Zhang, Z.: Post-quantum verifiable random function from symmetric primitives in pos blockchain **13554**, 25–45 (2022). https://doi.org/10.1007/978-3-031-17140-6“˙2, https://doi.org/10.1007/978-3-031-17140-6_2
3. Esgin, M.F., Kuchta, V., Sakzad, A., Steinfeld, R., Zhang, Z., Sun, S., Chu, S.: Practical post-quantum few-time verifiable random function with applications to algorand. In: Borisov, N., Díaz, C. (eds.) Financial Cryptography and Data Security - 25th International Conference, FC 2021, Virtual Event, March 1-5, 2021, Revised Selected Papers, Part II. Lecture Notes in Computer Science, vol. 12675, pp. 560–578. Springer (2021). https://doi.org/10.1007/978-3-662-64331-0“˙29, https://doi.org/10.1007/978-3-662-64331-0_29
4. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Algorand: Scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017. pp. 51–68. ACM (2017). https://doi.org/10.1145/3132747.3132757, https://doi.org/10.1145/3132747.3132757
5. Hülsing, A.: W-OTS+ - shorter signatures for hash-based signature schemes. In: Youssef, A.M., Nitaj, A., Hassanien, A.E. (eds.) Progress in Cryptology - AFRICACRYPT 2013, 6th International Conference on Cryptology in Africa, Cairo, Egypt, June 22-24, 2013. Proceedings. Lecture Notes in Computer Science, vol. 7918, pp. 173–188. Springer (2013). https://doi.org/10.1007/978-3-642-38553-7“˙10, https://doi.org/10.1007/978-3-642-38553-7_10
6. Lamport, L.: Constructing digital signatures from a one way function (1979)
7. Melara, M.S., Blankstein, A., Bonneau, J., Felten, E.W., Freedman, M.J.: CONIKS: bringing key transparency to end users. In: Jung, J., Holz, T. (eds.) 24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015. pp. 383–398. USENIX Association (2015), https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/melara
8. Merkle, R.C.: Secrecy, authentication, and public key systems. Stanford university (1979)

9. Micali, S., Rabin, M.O., Vadhan, S.P.: Verifiable random functions. In: 40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA. pp. 120–130. IEEE Computer Society (1999). https://doi.org/10.1109/SFFCS.1999.814584, https://doi.org/10.1109/SFFCS.1999.814584
10. National Institute of Standards and Technology (NIST): Stateful hash-based signature schemes (sp 800-208) (2020), https://csrc.nist.gov/News/2020/stateful-hash-based-signature-schemes-sp-800-208
11. Papadopoulos, D., Wessels, D., Huque, S., Naor, M., Včelák, J., Reyzin, L., Goldberg, S.: Making nsec5 practical for dnssec. Cryptology ePrint Archive (2017)
12. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999). https://doi.org/10.1137/S0036144598347011, https://doi.org/10.1137/S0036144598347011