mathematics

MDPI

Article

A Hash-Based Quantum-Resistant Designated Verifier Signature Scheme

P. Thanalakshmi [1], R. Anitha [1], N. Anbazhagan [2], Chulho Park [3], Gyanendra Prasad Joshi [4,*] and Changho Seo [3,*]

[1] Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore 641004, India; ptl.amcs@psgtech.ac.in (P.T.); ani.amcs@psgtech.ac.in (R.A.)
[2] Department of Mathematics, Alagappa University, Karaikudi 630004, India; anbazhagann@alagappauniversity.ac.in
[3] Department of Convergence Science, Kongju National University, Gongju 32588, Korea; pch022@naver.com
[4] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Korea
* Correspondence: joshi@sejong.ac.kr (G.P.J.); chseo@kongju.ac.kr (C.S.)

Abstract: Digital signatures are unsuitable for specific applications that are sensitive on a personal or commercial level because they are universally verifiable. Jakobsson et al. proposed the Designated Verifier Signature (DVS) system, which only allows the intended verifier to validate a message's signature. It prohibits the disclosure of a conviction to a third party. This functionality is useful in applications that require both authenticity and signer privacy, such as electronic voting and tender calls. The vast majority of current DVS schemes are based on difficult number theory problems such as integer factorization or discrete log problems over various groups. The development of a large-scale quantum computer would render these schemes unsafe. As a result, it is critical to develop quantum-resistant DVS methods. In both quantum and classical computers, signatures based on one-way functions are more efficient and secure. They have several advantages over digital signatures based on trapdoor functions. As a result, hash-based signatures are now considered viable alternatives to number-theoretic signatures. Existing hash-based signatures, on the other hand, are easily verifiable by anyone. As a result, they do not protect the signer's identity. In addition, they are one-time signatures. This paper presents a hash-based multi-time designated verifier signature scheme that ensures signer anonymity. The unforgeability of the signature scheme is also tested in the random oracle model under chosen message attack. The properties such as non-transferability and non-delegatability are investigated.

Keywords: digital signatures; hash-based cryptography; designated verifier signatures; homomorphic hash function; preimage resistance; random oracle model

MSC: 94A60; 94A62

Citation: Thanalakshmi, P.; Anitha, R.; Anbazhagan, N.; Park, C.; Joshi, G.P.; Seo, C. A Hash-Based Quantum-Resistant Designated Verifier Signature Scheme. *Mathematics* **2022**, *10*, 1642. https://doi.org/10.3390/math10101642

Academic Editors: Ioana Boureanu and Liqun Chen

Received: 13 April 2022
Accepted: 9 May 2022
Published: 11 May 2022

1. Introduction

Digital signatures play a vital role in the security of internet and IT infrastructures. Message integrity, authenticity, and non-repudiation are all provided by digital signatures. Traditional digital signatures can be verified by the public. However, in certain circumstances, the signer ought not uncover his signature to different gatherings, for example when signing individual well-being records, a financial statement, or a proposal of a vendor in an electronic auction. Henceforth, Chaum et al. [1] proposed an undeniable signature scheme in which the signature validation by a verifier requires an interactive confirmation protocol with the signer. Hence, the signer has complete full control over his signature. However, if the signer refuses to take part in the verification, then the signatures are considered to be invalid. To address this problem, Jakobsson et al. presented a new sort of digital signature called Designated Verifier Signature (DVS) [2], which replaces the interactive

verification of an undeniable signature scheme by a non-interactive verification and thereby reduces the burden on the signer. The DVS system enables the designated verifier to create signatures that are indistinguishable from the signer's. Subsequently, no outsider can identify the genuine signer of DVS. Although DVS achieves non-transferability, it does not prevent an outsider from checking the correctness of the scheme and being convinced that the signature is generated either by the signer or by the verifier. These significant features allow DVS to be more useful in scenarios such as electronic voting, tendering, and so on, where it is important for the signer to designate who has to be convinced by his/her signature.

1.1. Related Work

Saeednia et al. introduced the notion of Strong Designated Verifier Signature (SDVS) in [3] that requires the private key of the designated verifier to verify the signature. Hence, the signature can only be verified by the chosen verifier and not by an outsider. Followed by the author, Huang et al. [4] and Kang et al. [5] proposed identity-based strong designated verifier signature (IBSDVS) schemes and under the Bilinear Diffie–Hellman (BDH) assumption, the schemes are unforgeable. In [6], Laguillaumie and Vergnaud presented a multi-designated verifier signature technique that allows the signer to prove the authenticity of a statement to a chosen group of verifiers and allows the verifiers to create an identical signature by having cooperation among themselves. The authors also emphasized the concept of signer's identity privacy in SDVS and established that without knowing the secret keys, it is impossible to tell the difference between the signer's and the verifier's signatures. Li et al. [7] introduced the concept of non-delegatability to the DVS scheme, which ensures that the signer and verifier cannot entrust the generation of their signatures to a third party without disclosing their secret keys. Zhang and Mao proposed an IBSDVS scheme in [8] and claimed that their scheme is non-delegatable. De Almeida et al. [9] presented a protocol that uses DVS in a context of packet-switching networks. Attracted by the applications of DVS, many researchers proposed different DVS schemes. The majority of DVS schemes use certificate-based or identity-based cryptography, with only a few using certificateless cryptography [10–12].

According to Shor's algorithm [13], the emergence of quantum computers would make integer factorization and discrete logarithm issues insecure, which offer a stable foundation for the above-mentioned schemes. As a result, alternate strategies that are resistant to quantum computer attacks must be devised. Despite the fact that quantum computers are still in their early stages of development [14,15], their theoretical ability to compromise present cryptographic techniques has motivated the development of post-quantum cryptographic schemes. Thanalakshmi et al. [16] proposed the DVS scheme and Assar et al. [17] and Ren et al. [18] Shooshtari [19] proposed SDVS schemes based on the hard problem bounded syndrome decoding in coding theory that are believed to be quantum resistant [20,21]. However, it is demonstrated in [16] that the systems proposed by Ren et al. and Shooshtari et al. fail to meet non-transferability. Wang et al. [22], Li et al. [7], Noh and Jeong [23], and Cai et al. [24] proposed SDVS schemes based on the hard problems in lattices and claimed the schemes are quantum secure. However, the above schemes are based on trapdoor one-way functions and are considered to be more complex than the schemes based on one-way functions.

As hash functions are one-way and sufficient to have an efficient and secure transmission of data, the hash-based cryptosystem is considered as another promising quantum immune cryptosystem. In [25], the authors introduce hash-based signature schemes in the IoT ecosystem. For convenient transmission, a large file can be broken into smaller blocks and transmitted. One can receive a subset of blocks sequentially and finally be able to reconstruct the original file. Until the original file is decoded, one cannot check whether an intermediate block is valid or not. By using a homomorphic hash function, one can solve such a situation as follows. Compute the hash value of individual blocks and send the list of hash values to the user. The user can use it to verify the incoming blocks as they

come and can compute the hash of the original file from the hashes of individual blocks. Chen presented a PDP protocol based on an algebraic signature and a hash function with homomorphic property [26]. The homomorphic hash function is proven to enable rapid and efficient content retrieval as well as proved data possession and data integrity protection in cloud storage. Thanalakshmi et al.[27] proposed a quantum-resistant chameleon signature scheme with homomorphic hash functions and homomorphic pseudorandom generators and without using complex algebraic computation. Hence, to build identification schemes and special signatures such as DVS in a quantum world, quantum-resistant homomorphic hash functions are essential.

Lamport's one-time signature scheme in [28] laid a foundation for introducing the hash-based signature scheme by Merkle in [29]. Various extensions and improvements were made in Lamport's original scheme either by iterating the application of one-way functions several times or revealing the intermediate values of one-way functions as signatures. All of these methods can be thought of as variations on the approach proposed by Bleichenbacher and Maurer in [30]. As the security proofs of their schemes are left open, Hevia and Micciancio in [31] proposed a modification in [30] and proved the proposed signature is provably hard to break since the underlying functions are a hash function and a pseudorandom generator. A novel one-time signature scheme NOTS, which offers minimum key and signature sizes from existing OTS/FTS schemes, is presented in [32]. However, the above signature schemes are universally verifiable. Hence, they are unsuitable for applications where the intended verifier alone has to verify the signature. In addition, they are one-time—that is, each time when a signer wants to sign a document, he has to generate a new key pair, which in turn increases the key generation time. This motivates us to design a multi-time privacy providing DVS scheme based on hash functions. Although SDVS protects the privacy of signers, there are specific situations in which SDVS is ineffective and only DVS is appropriate. DVS is beneficial in electronic commerce applications such as the sale of digital products, for example. If Cindy buys a digital product from vendor Bob over the internet, she needs a digital receipt that ensures the product's quality, validity, and legality. Cindy would be convinced that the product is genuinely manufactured by Alice and sold by Bob if the receipt is bound with the identities of Bob and the product's manufacturer, say Alice. In this situation, Alice can create a DVS as a digital receipt for Bob. When Cindy buys a goods and receives a receipt from Bob, she may verify the signature using Bob's and Alice's public keys and be convinced of the merchandise. Under certain conditions, such as when the product requires service during the warranty period, Bob can provide it and issue a DVS receipt to satisfy the customer and build goodwill with the customer. In addition, DVS is preferable to SDVS, since Cindy will be unable to validate the signature if she receives SDVS as a receipt because SDVS requires Bob's private key for verification. Hence, in this paper, a DVS scheme is designed using a homomorphic hash function such that the signer can sign many documents with one private key, and the intended verifier can alone verify the validity of the signature. The scheme is proven to be existentially unforgeable under the chosen message attack in the random oracle model under the assumption of the preimage resistance of hash functions. The proposed technique further retains the non-transferability property by allowing the verifier to generate signatures that are indistinguishable from those issued by the signer. The proposed scheme satisfies the non-delegatability property, which is a desirable property for many applications of DVS such as the hypothetical e-voting protocol and the online subscription system.

1.2. Paper Organization

The following is a breakdown of the paper's structure. Cryptographic primitives, the specification of designated verifier signature schemes, and the security model are all covered in Section 2. In Section 3, a novel hash-based designated verifier signature mechanism is presented. In Section 4, the scheme's security is demonstrated using the

provided security definitions. Section 5 discusses the suggested scheme's performance as well as a comparison study. Finally, Section 6 brings the paper to a close.

2. Preliminaries

This section recalls the standard definition of a cryptographic hash function. It also introduces the concept and security requirements of a designated verifier signature scheme.

2.1. Cryptographic Primitives

Hash function is a powerful tool which is a crucial ingredient for many applications. It is typically used to ensure the data integrity, reduce the quantity of the data to be processed, and develop safe signature methods in the random oracle model.

Definition 1. *(Cryptographic Hash Function) A cryptographic hash function* $h : \{0,1\}^* \rightarrow \{0,1\}^n$ *is a mathematical function that has the following properties:*

(i) Computability or One-way: For any given message, it is easy to compute the hash value but practically impossible to invert.
(ii) Preimage resistance: It is computationally infeasible to find a message M that is hashed to y for any hash value y.
(iii) Second-preimage resistance: For a given message M, it is computationally infeasible to find another message M'*, which hashes to the same value as the message M, i.e.,* $h(M) = h(M')$*.*
(iv) Collision resistance: It is computationally infeasible to find another message M' *that hashes to the same value as the message M, i.e.,* $h(M) = h(M')$ *for a given message M.*

Definition 2. *(Homomorphic hash function). Let* $(R, +)$ *be a group. A homomorphic hash function is a cryptographic hash function* $H : R^2 \rightarrow R$ *that satisfies homomorphic property. Hence,* $H(x + y) = H(x) + H(y)$ *for every* $x, y \in R^2$*.*

A post-quantum homomorphic hash function is a homomorphic hash function that is resistant against quantum attacks. It is proposed by Micciancio in [33]. Chen et al. in [26] have proved that the hash family $\mathcal{H}_{R,m} = \{H : R^m \rightarrow R \mid R \text{ is a ring and } H \text{ is homomorphic}\}$ is a post-quantum homomorphic hash family where the ring R is $Z_p[x]/ < f >$ for an irreducible monic polynomial $f \in Z_p[x]$ of degree n and for some prime p.

The formal definition of a designated verifier signature scheme and its security requirements as in [2] are given in the following subsection.

2.2. Designated Verifier Signature Schemes

A DVS scheme consists of the following four polynomial-time algorithms:

Key Generation (1^κ)**:** The key generation algorithm is a probabilistic polynomial-time algorithm which takes the security parameter κ as input and generates public/private key pairs (pk_i, sk_i), $i = S, V$ where S ,V stands for the signer and verifier, respectively.

Sign: The signature generation algorithm outputs a designated verifier signature σ based on the input of a message M from the message space, the signing secret key sk_S, and the verifying public keys pk_S and pk_V. Either a probabilistic or deterministic algorithm can be used.

Verify: The signature verification algorithm is a deterministic algorithm that takes as inputs and outputs a bit string σ, a message M, and the verifying public keys pk_S and pk_V. If σ is a valid designated verifier signature on M, accept it; otherwise, reject it.

Sim: Transcript simulation is an algorithm that creates an identically distributed transcript σ' that is indistinguishable from the signature issued by the signer when given the designated verifier's secret key sk_V, the verifying public keys pk_S and pk_V, and a message M.

In public key infrastructure, everyone shields his/her private key and advertises his/her public key with the help of digital certificates. Otherwise, an attacker might use a signer's or verifier's information to create a new key pair in the signer's or verifier's name

and place a copy of the public key on a public key server. Assume a signer generates a DVS using a verifier's public key that has been placed by an attacker. The DVS property then allows the verifier to generate the signature. The attacker in this case is the verifier, who can easily generate the signature. To overcome this issue, a digital certificate should be used. It is a method of associating public keys with their owner. Certificate Authorities (CAs) issue these to validate the owners of public keys. The CA accomplishes this by validating (via various processes) the identity of the public key's owner. After that, it will bind the public key to a digital certificate and sign it with its private key to ensure its authenticity. All parties who need to validate the CA's assertion of public key ownership have access to the CA's public key.

2.3. Designated Verifier Signature Schemes' Security Model

The following important properties of a designated verifier signature scheme should be met: correctness, unforgeability, non-transferability, and non-delegatability.

Definition 3. *(Correctness) For any pk_S, sk_S, pk_V, sk_V and any message $M \in \{0,1\}^*$, the correctness of the algorithm requires that*

$$Pr\left[Verify\begin{pmatrix} pk_S, pk_V, M, \\ \sigma = sign(sk_S, \\ pk_S, pk_V, M) \end{pmatrix} = Accept\right] = 1$$

and

$$Pr\left[Verify\begin{pmatrix} pk_S, pk_V, M, \\ \sigma^{'} = sim(sk_V, \\ pk_S, pk_V, M) \end{pmatrix} = Accept\right] = 1.$$

Unforgeability

A valid DVS cannot be produced with non-negligible probability by anyone other than the signer and the designated verifier. Formally, we define unforgeability as the following game between a challenger C and a Probabilistic Polynomial-Time (PPT) adversary A:

(a) C produces (pk_S, sk_S) and (pk_V, sk_V) key pairs for the signer S and the verifier V, respectively, and gives pk_S, pk_V to the adversary A.
(b) O_h: For the appropriate inputs, A can query the hash oracle O_h.
(c) O_{Sign}: A can ask the signing oracle O_{Sign} for a signature on a message M for the signer S and the chosen verifier V. The oracle responds by returning a signature σ on M, where σ is valid with regard to pk_S and pk_V.
(d) Finally, A outputs a forgery σ^* on a message M^* without querying O_{Sign}. A wins the game if the signature is valid for M^* in terms of pk_S and pk_V and it did not query O_{Sign} on input M^*.

Definition 4. *(Unforgeability). (T, q_h, q_s, ϵ) is a DVS scheme that is existentially unforgeable against adaptive selected message attack if there is no adversary A who runs in time at most T, sends at most q_h queries to O_h, q_s queries to O_{sign} and wins the game with a probability of at least ϵ, as specified in [34].*

Non-transferability

Any probabilistic polynomial-time algorithm cannot distinguish the signature σ on a message M generated by either the signer or the designated verifier given a valid message–signature pair (M, σ). As a result, the signer's signature cannot be conveyed to a third party by a designated verifier. Its formal definition is as follows:

Definition 5. *(Non-transferability). If the signature generated by the designated verifier is indistinguishable from the original signature generated by the signer on the same message, the DVS*

method is non-transferable. It holds for any PPT, distinguisher D, and message $M \in \{0,1\}^*$ *for any* $(pk_S, sk_S), (pk_V, sk_V)$

$$\left| Pr\left(\begin{array}{c} \sigma_0 \leftarrow Sign(sk_S, pk_S, pk_V, M), \\ \sigma_1 \leftarrow Sim(sk_V, pk_S, pk_V, M), \\ b \leftarrow \{0,1\}, \\ b' \leftarrow D(sk_S, pk_S, sk_V, pk_V, \sigma_b) \\ : b' = b \end{array} \right) - \frac{1}{2} \right| \leq \epsilon(\kappa)$$

where the probability is taken over the randomness used in Sign and Sim, as well as the random coins used by D and $\epsilon(\kappa)$ *are a negligible function in the security parameter kappa. If the likelihood is* $\frac{1}{2}$*, the DVS scheme achieves perfect non-transferability, as stated in [17].*

Non-delegatability

Non-delegatability means that in order to generate a valid signature on a message, one must 'know' the signer's or designated verifier's secret key. Its formal definition is as follows:

Definition 6. *(Non-delegatability)* $(t, \epsilon, t', \epsilon')$ *is a non-delegatable DVS scheme if a knowledge extractor K can extract the private key* sk_S *or* sk_V *in time t with a probability of* $Adv^{ND}_{DVS}(\kappa) \geq \epsilon$ *against a forger algorithm F, a valid signature can be constructed on M with probability* ϵ' *in time* t'*, where* $\epsilon > poly_1(\epsilon')$ *and* $t < poly_2(t')$ *for two polynomial functions* $poly_1$ *and* $poly_2$*, respectively.*

3. Proposed Hash-Based Quantum-Resistant Designated Verifier Signature Scheme

The proposed Hash-Based Designated Verifier Signature Scheme (HBDVS) scheme is constructed using homomorphic hash function $H : \mathbb{F}_2^{2k} \rightarrow \mathbb{F}_2^k$ and a hash function $h : \mathbb{F}_2^k \times \mathbb{F}_2^k \times \mathbb{F}_2^k \rightarrow \mathbb{F}_2^l$, which are modeled as random oracles.

Construction of a Quantum-Resistant Hash-Based Designated Verifier Signature Scheme (HBDVS)

The scheme is made up of the polynomial-time algorithms listed below.

Signer S and verifier V with the input matrices X and Z of size l run Algorithm 1 and obtain the output matrices $H(X)$ and $H(Z)$. The signer and the verifier publish their public/private key pairs (pk_S, sk_S) as $(H(X), X)$ and $(H(Z), Z)$, respectively.

Algorithm 1 Key Generation (1^κ)

Input : H and a matrix X of size l where $X_i \in \mathbb{F}_2^{2k}$ for $i = 1, ..., l$
Output : (pk, sk)
Initialize pk of size l values as 0
Initialize sk as X
For $i = 1$ *to* l : do
    $pk[i] = H(X[i])$
EndFor
return (pk, sk)

The signer S chooses y_1,$t_1 \in \mathbb{F}_2^{2k}$ $r_1 \in \mathbb{F}_2^l$ and message M and runs Algorithm 2 and obtains the signature σ for M as (r_1, t_1, r_2, t_2).

When a signature $\sigma = (r_1, t_1, r_2, t_2)$ is received with the message M, the verifier runs Algorithm 3 and checks the validity of the signature. If the output is *True*, the verifier *Accepts* the signature σ for M and if the output is *False*, he *Rejects* the signature.

The verifier V randomly chooses y_2,$t_2 \in \mathbb{F}_2^{2k}$ and $r_2 \in \mathbb{F}_2^l$ and runs Algorithm 4 to simulate a signature σ for M.

Algorithm 2 Signing

Input : y_1,$t_1 \in \mathbb{F}_2^{2k}$,$r_1 \in \mathbb{F}_2^l$, pk_V, sk_S, h, H and M
Output : (r_1, t_1, r_2, t_2)
$Y_1 = H(y_1)$
$T_1 = H(t_1)$
$Y_2 = pk_V * r_1 \oplus T_1$
$r = h(M, Y_1, Y_2)$
$r_2 = r \oplus r_1$
$t_2 = sk_S * r_2 \oplus y_1$
return (r_1, t_1, r_2, t_2)

Algorithm 3 Verify

Input : signature $\sigma = (r_1, t_1, r_2, t_2)$, pk_S, sk_V, M, h and H
Output : Boolean (*True* or *False*)
$T_1 = H(t_1)$
$T_2 = H(t_2)$
$r = r_1 \oplus r_2$
$Y_1 = pk_S * r_2 \oplus T_2$, $Y_2 = pk_V * r_1 \oplus T_1$.
If $r = h(M, Y_1, Y_2)$ do:
    return "True"
Endif
Else do
    return "False"
EndElse

Algorithm 4 Sim

Input : y_2,$t_2 \in \mathbb{F}_2^{2k}$,$r_2 \in \mathbb{F}_2^l$, pk_S, sk_V, h, H and M
Output : (r_1, t_1, r_2, t_2)
$Y_2 = H(y_2)$
$T_2 = H(t_2)$
$Y_1 = pk_S * r_2 \oplus T_2$
$r = h(M, Y_1, Y_2)$
$r_1 = r \oplus r_2$
$t_1 = sk_V * r_1 \oplus y_2$.
return (r_1, t_1, r_2, t_2)

Correctness

A valid signature's correctness is determined as follows:

$$\begin{aligned} pk_S * r_2 \oplus T_2 &= H(sk_S) * r_2 \oplus H(t_2) \\ &= H(sk_S * r_2) \oplus H(t_2) \\ &= H(sk_S * r_2 \oplus t_2) \\ &= H(y_1) = Y_1 \end{aligned}$$

Similarly, one can verify $pk_V * r_1 \oplus T_1 = Y_2$. In addition, $r_1 \oplus r_2 = r$, according to the definition of r. As a result, $h(M, Y_1, Y_2) = r$. The correctness for a simulated signature follows the similar steps of correctness for a legitimate signature.

4. Security Analysis

The unforgeability of the proposed signature scheme is analyzed as follows:

EUF-CMA security: Pointcheval and Stern's [35] generic signatures are of the format $\sigma = (\sigma_0, h_1, \sigma_1)$, where σ_0 is randomly sampled from a huge set; $h_1 = h(M, \sigma_0)$ with a hash function h that is characterized as a random oracle, and M is the message to be signed; σ_1 depends merely on σ_0 and h_1. The signature on the message M that results is indicated by

$(\sigma_0, h_1, \sigma_1)$. Pointcheval and Stern showed that their schemes were existentially unforgeable in the random oracle model by the novel forking lemma.

The General Forking Lemma by Pointcheval and Stern [35] for a generic digital signature scheme is as follows:

The General Forking Lemma: Let κ be a security parameter for a generic digital signature system called Key Gen, Sign, Verify. Let A be a probabilistic polynomial-time Turing machine that takes public data as input and uses q_h and q_s queries to query the random oracle and the signer, respectively. Assume that within a time bound T, A creates a valid signature $(\sigma_1, h_1, \sigma_2)$ on M with the probability $\epsilon \geq \frac{10(q_s+1)(q_s+q_h)}{2^\kappa}$. When the triples $(\sigma_1, h_1, \sigma_2)$ can be simulated with an indistinguishable distribution probability without using the secret key, there exists another machine that has control over the machine A, replaces the interaction with the signer with simulation, and produces two valid signatures $(\sigma_1, h_1, \sigma_2)$ and $(\sigma_1, h_1^{'}, \sigma_2^{'})$ on M such that $h_1 \neq h_1^{'}$, in expected time $T^{'} \leq \frac{120686 q_h T}{\epsilon}$.

It is interesting to note that the proposed signature (r_1, t_1, r_2, t_2) on a message M meets the definition of Pointcheval and Stern's generic signatures. If r_1, t_1 and y_1 are sampled from big sets, r_2 can be easily derived from the hash output, and t_2 depends only on r_2 and the r_1, t_1 and y_1 inputs. The signature is supplied as (r_1, t_1, r_2, t_2), ignoring y_1, because the DVS scheme must have anonymity and the value of y_1 can be readily obtained from (r_2, t_2) when required. As a result, the Forking Lemma is used to assess the proposed scheme's security.

Theorem 1. *The signature (r_1, t_1, r_2, t_2) of the proposed scheme can be simulated using the hash oracle without knowing the private keys sk_S and sk_V, making it indistinguishable from the original signature unless the adversary can solve the hash function's preimage.*

Proof. A valid four-tuple (r_1, t_1, r_2, t_2), either original or simulated signature, must satisfy $pk_S * r_2 \oplus T_2 = Y_1$ and $pk_V * r_1 \oplus T_1 = Y_2$. Hence, simulation of the signature without the private keys sk_S and sk_V can be made as follows: choose randomly t_1, t_2 in F_2^{2k}, r_1, r_2 in F_2^l and compute $T_1 = H(t_1)$ and $T_2 = H(t_2)$, $Y_1 = pk_S * r_2 \oplus T_2$ and $Y_2 = pk_V * r_1 \oplus T_1$ and set $h(M, Y_1, Y_2) = r_1 \oplus r_2$. Therefore, the simulated signature satisfies either $Y_1 = H(y_1)$ or $Y_2 = H(y_2)$ where $y_1 = sk_S * r_2 \oplus t_2$ and $y_2 = sk_V * r_1 \oplus t_1$. However, all original signatures (r_1, t_1, r_2, t_2) satisfy both $Y_1 = H(y_1)$ and $Y_2 = H(y_2)$ with $h(M, Y_1, Y_2) = r_1 \oplus r_2$ and $y_1 = sk_S * r_2 \oplus t_2$ and $y_2 = sk_V * r_1 \oplus t_1$. Only with the knowledge of the secret keys sk_S and sk_V one can differentiate the simulated signature from a legitimate one. Therefore, telling a simulated signature from an original signature is the same as determining the secret keys from the public keys. It is the same as solving the hash function's preimage. □

Theorem 2. *Let A be an adversary who queries the hash and sign oracles with at most q_h and q_s queries, respectively, and performs an existential forgery under chosen message attack against the proposed HBDVS scheme with probability $\epsilon \geq \frac{10(q_s+1)(q_s+q_h)}{2^\kappa}$, within time bound T; then, the preimage of the hash function H can be computed within the expected time $T^{'} \leq \frac{120686 q_h T}{\epsilon}$. As a result, in the random oracle model, the HBDVS scheme is (T, q_h, q_s, ϵ)—existentially unforgeable.*

Proof. If an adversary A forges an HBDVS scheme with probability ϵ, an algorithm C exists that computes a preimage of the hash function H as follows: To construct a homomorphic hash function H, the challenger C executes the setup process on a security parameter κ. Then, C chooses randomly pk_S and pk_V as $k \times l$ matrices over $\mathbb{F}_2$ and sets them as the signer's and verifier's public key, respectively. C invokes A on parameters, pk_S and pk_V.

Let A query the hash oracle and sign oracle with at most q_h and q_s queries, respectively, and it wins the unforgeability game with probability ϵ. In Theorem 1, it is shown that the four-tuple (r_1, t_1, r_2, t_2) can be simulated without using the secret key with an indistinguishable distribution probability. According to General Forking Lemma, C that controls A replaces the interaction with the signer by simulation and produces two valid signatures

in expected time $T^{'} \leq \frac{120686 q_h T}{\epsilon}$. The two valid signatures (r_1, t_1, r_2, t_2) and $(r_1, t_1, r_2^{'}, t_2^{'})$ on M^* gives $y_1 = sk_S * r_2 \oplus t_2 = sk_S * r_2^{'} \oplus t_2^{'}$
$\Rightarrow sk_S = \frac{t_2^{'} \oplus t_2}{r_2 \oplus r_2^{'}}$. Thus, C determines the secret key sk_S, which is the preimage of the public key pk_S. In a similar manner, C computes the secret key sk_V for the public key pk_V. Thus, C succeeds in finding the preimages of hash function for two instances pk_S and pk_V within the expected time $T^{'} \leq \frac{120686 q_h T}{\epsilon}$. □

The following theorem proves non-transferability, which is an important property of the proposed DVS.

Theorem 3. *The distributions of the transcripts simulated by the verifier using the Sim algorithm are the same as the distributions of the transcripts received from the signer using the Sign algorithm. Hence, the proposed scheme is non-transferable.*

Proof. It is sufficient to establish that any valid designated verifier signature (r_1, t_1, r_2, t_2) on a message M created by a simulation algorithm is indistinguishable from one produced by a signing method to prove non-transferability. That is, the likelihood of the signing algorithm producing a signature is the same as the probability of the simulation algorithm producing a signature. For randomly selected $y_1, t_1 \in \mathbb{F}_2^{2k}$ and $r_1 \in \mathbb{F}_2^l$, the signature (r_1, t_1, r_2, t_2) produced by a signing algorithm is

$$\begin{pmatrix} r_1 \xleftarrow{R} \mathbb{F}_2^l \\ t_1 \xleftarrow{R} \mathbb{F}_2^{2k} \\ h(M, H(y_1), pk_V * r_1 \oplus H(t_1)) \oplus r_1 : y_1 \xleftarrow{R} \mathbb{F}_2^{2k} \\ sk_S * r_2 \oplus y_1 \end{pmatrix}$$

For randomly selected $y_2, t_2 \in \mathbb{F}_2^{2k}$ and $r_2 \in \mathbb{F}_2^l$, the signature (r_1, t_1, r_2, t_2) produced by a simulation algorithm is

$$\begin{pmatrix} h(M, pk_S * r_2 \oplus H(t_2), H(y_2)) \oplus r_2, : y_2 \xleftarrow{R} \mathbb{F}_2^{2k} \\ sk_V * r_1 \oplus y_2 \\ r_2 \xleftarrow{R} \mathbb{F}_2^l \\ t_2 \xleftarrow{R} \mathbb{F}_2^{2k} \end{pmatrix}$$

Let θ be a valid signature picked at random from among all the designated verifier signatures. The probability of a signature being formed by a signing algorithm is $Pr[\overline{\theta} = \theta] = \frac{1}{2^{4k+l}}$, while the probability of a signature $\theta^{'}$ being produced by a simulation algorithm is $Pr[\overline{\theta} = \theta^{'}] = \frac{1}{2^{4k+l}}$. When the random vectors and probabilities are compared, it is clear that the signatures produced by the signing and simulation algorithms have the same probability. As a result, the suggested signature system satisfies the property of non-transferability. □

Theorem 4. *The proposed HBDVS scheme cannot be delegated.*

Proof. We use the "General Forking Lemma" to show that the proposed HBDVS is non-delegatable. Assume that $\epsilon > \kappa = 1/2^l$ with $1/2^l$ representing the chance that F correctly predicts the hash value without consulting the random oracle h. We must prove that a knowledge extractor K exists that extracts the secret key of either the signer sk_S or the designated verifier sk_V with probability 1 using input σ and black-box oracle access to algorithm F. Let K choose randomly pk_S and pk_V as $k \times l$ matrices over $\mathbb{F}_2$ and establish them as the public keys of the signer and the verifier, respectively, and send to F. K supplies sign and hash oracles in the same manner as Theorem 2 does. K selects a message M^*

and sends it to F, which produces a signature on M^*. By General Forking Lemma, two signatures (r_1, t_1, r_2, t_2) and $(r_1, t_1, r_2^{'}, t_2^{'})$ on M^* are obtained.

This $\Rightarrow y_1 = sk_S * r_2 \oplus t_2 = sk_S * r_2^{'} \oplus t_2^{'} \Rightarrow sk_S = \frac{t_2^{'} \oplus t_2}{r_2 \oplus r_2^{'}}$. Thus, K computes the secret key sk_S for the public key pk_S. Similarly, K computes the secret key sk_V for the public key pk_V. □

5. Results and Discussion

In the proposed scheme, the signature generation requires three time computations of the hash function. Hence, the signature generation has the time complexity $3(O(1)) = O(1)$. The time complexity of verification is $O(1)$, since the verification process also requires three time computations of hash evaluation.

Table 1 shows the comparison results of the proposed method with a few current post-quantum DVS techniques for 256-bit security. We consider hash functions given by Ajtai's construction of hash functions based on regular lattices for the proposed HBDVS. As a result, the HBDVS parameters, $k = nlog, q$ and $l = log, q$, select the values of n and q, as cited in [33]. The other schemes compared in Table 1 are lattice-based schemes in ideal lattices. As a result, the parameters of those schemes are regarded as in [24]. Table 1 indicates that the proposed scheme's signature length is shorter than that of previous quantum secure techniques.

Table 1. Comparison of HBDVS scheme with existing DVS schemes.

Scheme	System	Hard Problem	Signature Size in Bits
Wang et al. (2012) [22]	Lattice-based	LWE-SIS	3.3×10^6
Noh and Jeong (2016) [23]	Lattice-based	LWE-SIS	9.7×10^4
Cai et al. (2019) [24]	Lattice-based	R-SIS	1.6×10^4
Proposed HBDVS	Hash-based	PR	7.1×10^3

SIS: short integer solution; LWE: learning with errors; R-SIS: ring-based analogue of SIS problem; PR: preimage resistance.

6. Conclusions

A DVS scheme is designed to preserve the signer's privacy, which is a requirement for many applications such as electronic voting, tender calls, and so on. As the attacks of the quantum computer threatens most of the existing designated verifier signature schemes, in this paper, a provably secure hash-based designated verifier signature scheme is proposed. The scheme is constructed with a minimal resource of homomorphic hash function. It is a many-time scheme without using the Merkle tree structure. It is a feasible replacement to number-theoretic techniques. The scheme's EUF-CMA security is demonstrated in the random oracle. It also meets the security requirements for a designated verifier signature technique, such as non-transferability and non-delegatability. This work will enhance the potential of one-way functions in public key cryptography. This study's scope could be expanded to include various types of hash-based designated verifier signatures, such as strong designated verifier signatures, proxy signatures, designated confirmer signatures, and so on.

Author Contributions: Conceptualization, P.T.; Data curation, P.T.; Formal analysis, P.T. and R.A.; Funding acquisition, G.P.J. and C.S.; Investigation, N.A. and C.P.; Methodology, R.A.; Project administration, G.P.J.; Resources, G.P.J. and C.S.; Software, N.A. and C.P.; Supervision, C.S.; Validation, R.A.; Visualization, N.A. and C.P.; Writing—original draft, P.T.; Writing—review and editing, G.P.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through the MSIT (Development of

Highly Efficient PQC Security and Performance Verification for Constrained Devices) under Grant 2021-0-00400.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Anbazhagan would like to thank RUSA Phase 2.0 (F 24-51/2014-U), DST-FIST (SR/FIST/MS-I/2018/17), DST-PURSE 2nd Phase programme (SR/PURSE Phase 2/38) and UGC-SAP(DRS-I) (F.510/8/DRS-I/2016(SAP-I)), Govt. of India.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chaum, D.; Antwerpen, H.V. Undeniable signatures. In *Proceedings of the Conference on the Theory and Application of Cryptology, LNCS*; Springer, New York, NY, USA, 1989; Volume 435, pp. 212–216.
2. Jakobsson, M.; Sako, K.; Impagliazzo, R. Designated verifier proofs and their applications. In *Advances in Cryptology EUROCRYPT96*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 143–154.
3. Saeednia, S.; Kremer, S.; Markowitch, O. An efficient strong designated verifier signature scheme. In Proceedings of the International Conference on Information Security and Cryptology, Seoul, Korea, 27–28 November 2003; pp. 40–54.
4. Huang, X.; Susilo, W.; Mu, Y.; Zhang, F. Short (identity-based) strong designated verifier signature schemes. In Proceedings of the International Conference on Information Security Practice and Experience, Hangzhou, China, 11–14 April 2006; pp. 214–225.
5. Kang, B.; Boyd, C.; Dawson, E. A novel identity-based strong designated verifier signature scheme. *J. Syst. Softw.*, **2009**, *82*, 270–273. [CrossRef]
6. Laguillaumie, F.; Vergnaud, D. Designated verifier signatures: Anonymity and efficient construction from any bilinear map. In Proceedings of the International Conference on Security in Communication Networks, Amalfi, Italy, 8–10 September 2004; pp. 105–119.
7. Li, Y.; Lipmaa, H.; Pei, D. On delegatability of four designated verifiersignatures. In Proceedings of the International Conference on Information and Communications Security, Beijing, China, 10–13 December 2005; pp. 61–71.
8. Zhang, J.; Mao, J. A novel id-based designated verifier signature scheme. *Inf. Sci.* **2008**, *178*, 766–773. [CrossRef]
9. De Almeida, M.P.; de Sousa, Júnior, R.T.; García, Villalba, L.J.; Kim, T.H. New DoS defense method based on strong designated verifier signatures. *Sensors* **2008**, *18*, 2813. [CrossRef] [PubMed]
10. Chen, Y.; Zhao, Y.; Xiong, H.; Yue, F. A certificateless strong designated verifier signature scheme with non-delegatability. *IJ Netw. Secur.* **2017**, *19*, 573–582.
11. Lin, H.Y. A new Certificateless strong designated verifier signature scheme: Non delegetable and SSA-KCA Secure. *IEEE Access* **2018**, *6*, 50765–50775. [CrossRef]
12. Han, S.; Xie, M.; Yang, B.; Lu, R.; Bao, H.; Lin, J.; Han, S. A certificateless verifiable strong designated verifier signature scheme. *IEEE Access* **2019**, *7*, 126391–126408. [CrossRef]
13. Shor, P.W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* **1997**, *26*, 1484–1509. 1997. [CrossRef]
14. Chang, W.L.; Vasilakos, A.V. *Fundamentals of Quantum Programming in IBM's Quantum Computers*; Springer: Berlin/Heidelberg, Germany, 2021.
15. Chang, W.L.; Chen, J.C.; Chung, W.Y.; Hsiao, C.Y.; Wong, R.; Vasilakos, A.V. Quantum Speedup and Mathematical Solutions from Implementing Bio-molecular Solutions for the Independent Set Problem on IBM's Quantum Computers. *IEEE Trans. NanoBiosci.* **2021**, *20*, 354–376. [CrossRef]
16. Thanalakshmi, P.; Anitha, R. A new code-based designated verifier signature scheme. *International J. Commun. Syst.* **2018**, *31*, e3803. [CrossRef]
17. Asaar, M.R.; Salmasizadeh, M.; Aref, M.R. Code-based Strong Designated Verifier Signatures: Security Analysis and a New Construction. *IACR Cryptol. ePrint Arch.* **2016**, *779*, 1–15
18. Ren, Y.; Wang, H.; Du, J.; Ma, L. Code-based authentication with designated verifier. *Int. J. Grid Util. Comput.* **2016**, *7*, 61–67. [CrossRef]
19. Shooshtari, M.K.; Ahmadian-Attari, M.; Aref, M.R. Provably secure strong designated verifier signature scheme based on coding theory. *Int. J. Commun. Syst.* **2016**, *30*, e3162. [CrossRef]
20. Daniel; A; Lejla; B.; Bernstein, D.J.; Bos, J.; Buchmann, J.; Castryck, W.; Dunkelman, O.; Guneysu, T.; Gueron, S.; Hulsing, A.; et al. Initial Recommendations of Long-Term Secure Post-Quantum Systems. PQCRYPTO. EU. Horizon 2020 2015. Available online: https://pqcrypto.eu.org/docs/initial-recommendations.pdf (accessed on 1 February 2020).
21. Process, S.P. *Third Round Candidate Announcement*; NIST Computer Security Resource Center (CSRC): Gaithersburg, MD, USA, 2020.

22. Wang, F.; Hu, Y.; Wang, B. Lattice-based strong designate verifier signature and its applications. *Malays. J. Comput. Sci.* **2012**, *25*, 11–22.
23. Noh, G.; Jeong, I.R. Strong designated verifier signature scheme from lattices in the standard model. *Secur. Commun. Netw.* **2016**, *9*, 6202–6214. [CrossRef]
24. Cai, J.; Jiang, H.; Zhang, P.; Zheng, Z.; Lyu, G.; Xu, Q. An Efficient Strong Designated Verifier Signature Based on R—SIS Assumption. *IEEE Access* **2019**, *7*, 3938–3947. [CrossRef]
25. Suhail, S.; Hussain, R.; Khan, A.; Hong, C.S. On the role of hash-based signatures in quantum-safe internet of things: Current solutions and future directions. *IEEE Internet Things J.* **2020**, *8*, 1–17. [CrossRef]
26. Chen, L.; Han, L.; Jing, J.; Hu, D. A post quantum provable data possession protocol in cloud. *Secur. Commun. Netw.* **2013**, *6*, 658–667. [CrossRef]
27. Thanalakshmi, P.; Anitha, R.; Anbazhagan, N.; Cho, W.; Joshi, G.P.; Yang, E. A Hash-Based Quantum-Resistant Chameleon Signature Scheme. *Sensors* **2021**, *21*, 8417. [CrossRef]
28. Lamport, L. *Constructing Digital Signatures from a One-Way Function*; Technical Report CSL-98; SRI International: Menlo Park, CA, USA, 1979; Volume 238.
29. Merkle, R.C. A digital signature based on a conventional encryption function. In Proceedings of the Conference on the Theory and Application of Cryptographic Techniques, Davos, Switzerland, 25–27 May 1988; pp. 369–378.
30. Bleichenbacher, D.; Maurer, U.M. Directed acyclic graphs, one-way functions and digital signatures. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 21–25 August 1994; pp. 75–82.
31. Hevia, A.; Micciancio, D. The provable security of graph-based one-time signatures and extensions to algebraic signature schemes. In Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, 1–5 December 2002; pp. 379–396.
32. Shahid, F.; Ahmad, I.; Imran, M.; Shoaib, M. Novel one time Signatures (NOTS): A compact post-quantum digital signature scheme. *IEEE Access* **2020**, *8*, 15895–15906. [CrossRef]
33. Micciancio, D.; Regev, O. Lattice-based cryptography. In *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 147–191.
34. Feng, D.; Xu, J.; Chen, W. Generic Constructions for Strong Designated Verifier Signature. *J. Inf. Process. Syst.* **2011**, *7*, 159–172. [CrossRef]
35. Pointcheval, D.; Stern, J. Security arguments for digital signatures and blind signatures. *J. Cryptol.* **2000**, *13*, 361–396. [CrossRef]