ON THE IMPLEMENTATION OF A LATTICE-BASED DAA FOR VANET SYSTEM

Doryan Lesaignoux and Mikael Carmona
Univ. Grenoble Alpes, CEA, Leti, MINATEC Campus, F-38054 Grenoble, France
{doryan.lesaignoux, mikael.carmona}@cea.fr

Keywords: Signature scheme, Lattice-based cryptography, Direct Anonymous Attestation, Zero-knowledge proof, Software implementation, Performances.

Abstract: Direct Anonymous Attestation (DAA) is a cryptographic protocol that enables users with a Trusted Platform Module (TPM) to authenticate without revealing their identity. Thus, DAA emerged as a good privacy-enhancing solution. Current standards have security based on factorization and discrete logarithm problem making them vulnerable to quantum computer attacks. Recently, a number of lattice-based DAA has been propose in the literature to start transition to quantum-resistant cryptography. In addition to these, DAA has been adapted to Vehicle Ad-hoc NETwork system (VANETs) to offer secure vehicule-to-vehicule/infrastructure communication (V2V and V2I). In this paper, we provide an implementation of the most advanced post-quantum DAA for VANETs. We explore the cryptographic foundations, construction methodologies, and the performance of this scheme, offering insights into their suitability for various real-world use cases.

1 INTRODUCTION

Direct Anonymous Attestation (DAA) is an anonymous digital signature scheme made up of three entities: The issuer, a set of signers and a set the verifiers (Brickell and *al.*, 2004). Issuer oversees credentials distribution for each signer. A signer is a pair of TPM and host. She proves her membership and her trustworthy to the group with a DAA signature. This signature includes a zero-knowledge proof *i.e.* user can prove she has valid credentials without the verifier learning any information other than the validity. DAA can be seen as a group signature variant with stronger privacy: group signature has a property of traceability *i.e.* the group manager can remove anonymity of a signer. Instead, DAA has a property of user-controlled linkability. This property is steered using a basename *i.e.* if the signer uses a same basename for two signatures, the verifier can know that these two signatures come from the same device.

DAA has been developed by the Trusted Computing Group (TCG) with the aim of striking a balance between data security and user privacy. In 2004, the first RSA-based DAA (Brickell and *al.*, 2004) is standardized by the TCG for TPM 1.2 (TCG, 2004). In 2008, Brickel, Chen and Li published the first DAA based on Elliptic Curve Cryptography in (Brickel and *al.*, 2008 and 2009). After several enhancements (Chen, 2010), (Chen and *al.*, 2010), (Smyth and *al.*, 2012), a series of ECC-based DAA protocols is specified in ISO/IEC 20008-2 to be integrated in the new generation of TPM 2.0 (Trusted Computed Group, 2014). However, those schemes are not secure against quantum cryptanalysis.

Post-quantum DAA state-of-art show that latticed-based cryptography is widely used compared to other constructions such as code or hash. From today, few Lattice-based DAA (LDAA) have been developed. The pioneering scheme is proposed in (El Bansarkhani and *al.*, 2017) and is improved in (El Kassem and *al.*, 2019) in term of speed and size of instances. However, the scheme still requires huge storage and computing capacity. Then, a new framework of LDAA is introduced in (El Kassem and *al.*, 2019) which improves signature size by two orders of magnitude regarding previous post-quantum DAA.

Vehicular Ad-hoc Network (VANET) is a network enabling communication between vehicles (V2V) as well as between vehicles and road infrastructures (V2I). The main goal of VANETs is to enhance road safety and traffic efficiency by allowing vehicles to communicate with each other and the surrounding infrastructure. Research studies several type of authentication schemes such as Public Key

Infrastructure (VPKI) (Petit and *al.*, 2014), ring signature (Cui and *al.*, 2017) or DAA (Chen and *al.*, 2011). However, those architectures still have shortcomings. In VPKI, generation and management of pseudonyms by the pseudonym provider requires high computational and storage resources. Additionally, the processus of revocation requires a pseudonym resolution in order to retrieve the user's long-term ID. With ring signatures, the computational cost and the signature size depend on the number of users. By using DAA for VANETs, users themselves generate pseudonyms and revocation does not require recovering the long-term ID. Moreover, performance does not depend on the number of users in the network. Based on the same framework as (Chen and *al.*, 2019), the first post-quantum DAA applied to VANETs is described by (Chen and *al.*, 2021) with some optimizations of the underlying operations coming from the original scheme.

In the objective to implement such schemes on operational targets (such as TPM), studying their performance and identifying the bottleneck is a central task. Due to the recent nature of LDAA, it exits only one work (Chen and *al.*, 2021) providing deep insight on implementation concerns and performance. In particular, the generation of parameters to achieve a given security level is not straightforward. The purpose of this paper is to provide a software implementation of the most advanced post-quantum DAA scheme for VANETs named V-LDAA (Chen and *al.*, 2021). The implementation comes with a structured and comprehensive overview of the scheme, and, with a methodology for input parameters generation. A performance and bottleneck analysis is proposed and completed by a benchmark with pre-quantum DAA.

The paper is organized in two sections. Section 2 introduces the studied V-LDAA and describes the methodology used to derive a set of parameters for a given security level. Section 3 describes our software implementation of the V-LDAA. The performance analysis allows the bottleneck identification and a comparison with the existing implementation from (Chen and *al.*, 2021) and the pre-quantum DAA.

2 Study of V-LDAA SCHEME

2.1 Notations

In this paper, we define the polynomial ring $\mathcal{R} = \mathbb{Z}[X]/\langle X^d + 1\rangle$ and quotient ring $\mathcal{R}_\mathrm{q} = \mathcal{R}/q\mathcal{R}$ with q a prime integer and d an integer power of 2. For clarity, we denote element of $\mathcal{R}$ in lowercase letter, vector over $\mathcal{R}$ in bold lowercase letter and matrix over $\mathcal{R}$ in capital bold letter. We consider the scalar product of two elements of $\mathcal{R}$ as $\langle a|b\rangle = \sum a_i b_i$ where a_i and b_i are the coefficients of polynomials a and b. We denote the Euclidean norm over $\mathcal{R}$ as $\|a\|_2 = \|a\| = \sqrt{\langle a|a\rangle}$. We further define the notation for vectors $\|\boldsymbol{a}\| = \sqrt{\sum \|a_i\|^2}$ and matrices $\|\boldsymbol{A}\| = \sqrt{\sum \|\boldsymbol{a_i}\|^2}$.

We denote by $D_{\Lambda,\sigma,\boldsymbol{c}}$ the discrete gaussian distribution over a lattice Λ with standard deviation $\sigma > 0$ and center $\boldsymbol{c} \in \mathbb{R}^n$. If $\boldsymbol{c} = 0_n$, we omit it.

2.2 V-LDAA: General framework

V-LDAA is a digital signature scheme secure under Random Oracle Model where each user composed of a host and a TPM can prove their trustworthiness to a party of the network *i.e.* a verifier. It is important to notice that even if a host and a TPM are embedded in the same platform, they have two distinct roles in the DAA. The scheme is described in (Chen and *al.*, 2021) and is based on the same framework proposed in (Chen and *al.*, 2019). V-LDAA is composed of five main primitives:

- **Setup** to initialize parameters of TPM, host and issuer.
- **Join** to enable a user to join the network and get credentials from the issuer.
- **Create** to build pseudonym and two signatures for revocation process and credentials verification.
- **Sign/Verify** for signing and verifying a signature of a plain text.
- **Revok** to revoke a user.

Notice that **Linkability** primitive is not specified (Chen and *al.*, 2021) but it is described in the original framework (Chen and *al.*, 2019). In the **Join** process, the creation of credentials is based on the same concept as the ABB lattice-based signature scheme (Agrawal and *al.*, 2010) with a slight modification. Following a Zero-Knowledge Proof (ZKP) proving that a user has a valid endorsement key, the issuer generates a pair $(i, \boldsymbol{s})$, named credentials, satisfying the relation:

$$\overbrace{[\overline{\boldsymbol{A}} \mid \boldsymbol{B} + i\boldsymbol{g}]}^{\boldsymbol{A}} \cdot \boldsymbol{s} = u + \boldsymbol{u_T} \cdot \boldsymbol{e_T} + \boldsymbol{u_H} \cdot \boldsymbol{e_H} \tag{1}$$

where $\overline{\boldsymbol{A}}, \boldsymbol{B}, u$ are the issuer public key, $\boldsymbol{g}$ is the gadget matrix and the two pairs $(\boldsymbol{u_T}, \boldsymbol{e_T})$, $(\boldsymbol{u_H}, \boldsymbol{e_H})$ are the public and private key of the TPM and the host. i is the long-term ID attributed to the new member and $\boldsymbol{s}$ is a secret bounded-norm vector of polynomials proving membership. It's important to notice that the matrix $\boldsymbol{A}$ is unique for each member due to the long-term ID i. Thus, it is essential that this matrix is known only to the issuer and the user concerned, otherwise one could break the anonymity of anyone by recovering i. Before signing anonymously a message, a member need first, to build a pseudonym by creating a Ring-LWE instance with the TPM's secret key and a basename. Two signatures will be linkable if a user decides to use the same basename for each of them. Second, they must generate a ZKP (named attestation signature) of their credentials to prove their legitimacy to the verifiers with the underlying primitive $\mathbf{Blindsign}$. This procedure is described in **Create** primitive. During **Sign** operation, the signer sends the signature of the message as well as the signature of the credentials. Finally, the verifier checks each signature.

For further details, we refer the reader to the references papers (Chen and *al.*, 2019) and (Chen and *al.*, 2021).

2.3 Basics and concepts on lattices for LDAA

In this section, we describe fundamental lattice-based notions used in V-LDAA scheme.

A lattice Λ of $\mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$. Any lattice Λ of $\mathbb{R}^n$ is spanned on $\mathbb{Z}$ by a set of m vectors $\boldsymbol{a_0}, \dots, \boldsymbol{a_{m-1}}$ of $\mathbb{R}^n$ where $0 < n \leq m$. This set is a (non-unique) base for Λ. Let $\boldsymbol{A} \in \mathbb{Z}_q^{n\times m}$, by definition, $\Lambda(\boldsymbol{A})$ is the lattice of $\mathbb{R}^n$ spanned by the columns of $\boldsymbol{A}$. Given a lattice Λ with basis $\boldsymbol{B}$ and $\widetilde{\boldsymbol{B}}$ the Gram-Schmidt orthogonalization of $\boldsymbol{B}$, we define a bound of the smoothing parameter η_ϵ of Λ as $\eta_\epsilon(\Lambda) < \|\widetilde{\boldsymbol{B}}\| \ln(2n(1+1/\epsilon)/\pi)$ for any $\epsilon > 0$ (Theorem 3.1, Gentry and *al.*, 2008).

For lattice-based cryptography applications, we introduce two q-ary lattices $\Lambda_q^{\perp}(\boldsymbol{A}) := \{\boldsymbol{x} \in \mathbb{Z}^m \mid \boldsymbol{Ax} = \boldsymbol{0_n} \mod q\}$, the set of all vectors of $\mathbb{Z}^m$ orthogonal to $\boldsymbol{A}$ and $\Lambda_q^{u}(\boldsymbol{A}) := \{\boldsymbol{v} \in \mathbb{Z}^m \mid \boldsymbol{Av} = u \mod q\}$, the set of all pre-images $\boldsymbol{v}$ of $\boldsymbol{u}$ by $\boldsymbol{A}$. Two problems follow from these lattices:

- Short Integer Solution problem (SIS): Given a real $\beta > 0$, (Atjai, 1996) showed that finding a vector $\boldsymbol{x}$ in $\Lambda_q^{\perp}(\boldsymbol{A})$ such that $0 < \|\boldsymbol{x}\| \leq \beta$ is NP-hard.
- Learning With Error problem (LWE): Given a random secret vector $\boldsymbol{s} \in \mathbb{Z}_q^m$ and a gaussian error vector $\boldsymbol{e} \leftarrow \mathcal{D}_{\mathbb{Z}^n,\mu}$, (Regev, 2009) showed that, if $\mu q > 2\sqrt{n}$, finding $\boldsymbol{s}$ by knowing the pair $(\boldsymbol{A}, \boldsymbol{As}+\boldsymbol{e})$ is NP-hard.

These two cryptographic problems are adaptable to the polynomial case in $\mathcal{R}$ with Ring-SIS (Lyubashevsky and *al.*, 2006) and Ring-LWE (Lyubashevsky and *al.*, 2010). They guarantee the security of V-LDAA.

Let δ be an integer > 0, $\boldsymbol{g} = (1 \;\; \delta \;\; \dots \;\; \delta^{k-1})$ be the gadget matrix in $\mathbb{Z}^k$ for polynomials and i be an invertible polynomial in $\mathcal{S}_q = \{a_0 + a_1 X^{d/2} \mid a_0, a_1 \in \mathbb{Z}_q\}$. Notice that $k = \log_\delta(q)$. The construction of a trapdoor is based on a matrix $\boldsymbol{A} = [\overline{\boldsymbol{A}} | i\boldsymbol{g} - \overline{\boldsymbol{A}}\boldsymbol{R}] \in \mathcal{R}_q^{n\times m}$ with $\overline{\boldsymbol{A}}$ a random uniform matrix in $\mathcal{R}_q^{n\times nk}$. The matrix $\boldsymbol{R} \in \mathcal{R}^{(m-nk)\times nk}$ is called a $\boldsymbol{g}$-trapdoor for $\boldsymbol{A}$ under the tag i and with $\boldsymbol{A}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{I}\end{bmatrix} = i\boldsymbol{g}$ (Micciancio and Peikert, 2012). Applied to V-LDAA, we recover the same structure described in equation (1) where $\boldsymbol{R}$ is the issuer secret key and $\boldsymbol{B} = -\overline{\boldsymbol{A}}\boldsymbol{R}$. We denote this procedure by the primitive $(\boldsymbol{A}, \boldsymbol{R}) = \mathbf{TrapGen}(n, m, k, q, \beta)$. In the next section, we give concrete parameters of this construction for V-LDAA.

To generate credentials for each user in the **Join** process, issuer needs to generate a m-dimensional discrete gaussian pre-image $\boldsymbol{s}$ satisfying (1) from $D_{\mathcal{R}^m,\sigma,\boldsymbol{c}}$. This can be done using primitive $\mathbf{SampleGaussian}(\boldsymbol{A}, \boldsymbol{R}, i, u, \gamma, \alpha)$. In this work, we use sampling method from (Micciancio and Peikert, 2012).

As specified in (section 5.4, Micciancio and Peikert, 2012), we first need to sample perturbation vector $\boldsymbol{p} \leftarrow \boldsymbol{\mathcal{D}}_{\mathcal{R}^m,\sqrt{\Sigma_p}}$ with algorithm **SamplePerturbZ** and covariance $\Sigma_p = \gamma^2 \boldsymbol{I} - \alpha^2 \begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{I}\end{bmatrix}[\boldsymbol{R} \quad \boldsymbol{I}]$. Then, compute syndrome $v = i^{-1}(u - \boldsymbol{Ap})$, sample $\mathbf{z} \leftarrow \boldsymbol{\mathcal{D}}_{\Lambda_q^v(\boldsymbol{g}),\alpha}$ with algorithm **SampleG** and output $\boldsymbol{s} = \boldsymbol{p} + \begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{I}\end{bmatrix}\boldsymbol{z}$. We emphasize the perturbation sampling, which ensures that no information about the trapdoor $\boldsymbol{R}$ is leaked.

Another important notion used in DAA protocols is zero-knowledge proof (ZKP) since they are used in **Join**, **Create**, **Sign** and **Revok** procedure. In lattice-based cryptography, a ZKP is to prove the knowledge of a low-norm secret vector $\boldsymbol{s}$ in $\Lambda_q^u(\boldsymbol{A})$ without revealing any other information. However, schemes such as (Ling and *al.*, 2013) proving the exact knowledge of $\boldsymbol{s}$ remain very expensive in proof size and computation. Over polynomial rings, a more efficient way is to use “Fiat-Shamir with Abort” ZKP technique from (Lyubashevsky, 2012). It relies on proving the knowledge of a slighty larger-norm vector $\overline{\boldsymbol{s}}$ such that $\boldsymbol{A}\overline{\boldsymbol{s}} = cu$ where c is a sparse polynomial challenge of $\mathcal{C} = \{c \in R_q \mid \|c\|_\infty = 1 \wedge \|c\|_1 = \kappa, 2^\kappa \binom{d}{\kappa} \geq 2^\zeta\}$. The secret key is “hidden” by creating a Ring-LWE instance. Then, the protocol uses a rejection sampling subroutine of standard deviation ξ to assert outputs are Gaussian distributed and independent of the secret key and the challenge. Finally, the verifier sets a bound of acceptance β_z proportional to ξ to check the proof. We call this primitive $\mathbf{ZKP}(\boldsymbol{A}, \boldsymbol{s}, u, \xi, \beta_z)$.

In **Create**, users generate a signature with the algorithm $\mathbf{BlindSign}$ to enable verifiers to check the authenticity of their credentials. Formally, the protocol is a non-interactive ZKP from (Del Pino and *al.*, 2018) proving that the user knows a pair $(\boldsymbol{s}, i)$ satisfying (1). However, this cannot be performed directly because verifiers need to know the value of $[\overline{\boldsymbol{A}} \mid \boldsymbol{B} + i\boldsymbol{g}]$ which would compromise the long-term ID i. Therefore, the proof is applied on a commitment value of each term of $i\boldsymbol{g}$. For simplification, we consider $\boldsymbol{g} = (1 \quad \delta)$, the same gadget matrix as in V-LDAA. We use the method described in (Baum and *al.*, 2018). The main idea is to sample two random uniforms matrix $\boldsymbol{A_1} = [\boldsymbol{I_n} \quad \boldsymbol{A'_1}]$ with $\boldsymbol{A'_1} \xleftarrow{\$} \mathcal{R}_q^{n\times(m-n)}$, $\boldsymbol{A_2} = [\boldsymbol{0_n} \quad 1 \quad \boldsymbol{A'_2}]$ with $\boldsymbol{A'_2} \xleftarrow{\$} \mathcal{R}_q^{1\times(m-n-1)}$ and a short random vector $\boldsymbol{r}$ in $\mathcal{R}_3^m$ s.t : $Com(x, \boldsymbol{r}) = [\boldsymbol{A_1} \quad \boldsymbol{A_2}]^{\mathrm{T}}\boldsymbol{r} + [0 \quad x]^T = [t_1 \quad t_2]^T = t$ where x is a value to commit. Given two commitments $t = Com(i, \boldsymbol{r})$ and $t' = Com(i\delta, \boldsymbol{r'})$, observe that:

$$\boldsymbol{v^T s'} = u \tag{2}$$

where : $\boldsymbol{v^T} = [\boldsymbol{A^T} | \boldsymbol{B^T} + [\boldsymbol{t_2} \ \boldsymbol{t'_2}]| - [\boldsymbol{u_T} \ \boldsymbol{u_h}]| - [\boldsymbol{a_T} \ \boldsymbol{a_H}]]$ and $\boldsymbol{s'} = [\boldsymbol{s_1} \quad \boldsymbol{s_2} \quad [\boldsymbol{e_T} \ \boldsymbol{e_H}] \ - [\boldsymbol{r} \ \boldsymbol{r'}]\boldsymbol{s_2}]^T$.

With this commitment scheme, the protocol **BlindSign** creates 3 proofs of knowledge (Del Pino and *al.*, 2018):

- Π_1 : commit values open to each committed values x: using "Fiat-Shamir with Abort" technique explained above, one can prove the knowledge of a slightly larger-norm polynomial vector $\bar{\boldsymbol{r}}$ s.t: $ct = [\boldsymbol{A_1} \quad \boldsymbol{A_2}]^{\mathrm{T}}\,\bar{\boldsymbol{r}} + c[0 \quad x]^T$.
- Π_2 : commit values open to long-term ID i such that $\sigma_5(i) = i$ where $\sigma_5 : X \to X^5$. The protocol uses automorphism stability to prove that $i \in S_q$ (Corollary 4.2 from (Del Pino and *al.*, 2018)).
- Π_3 : user knows s' satisfying (2).

2.4 Parametrization of the V-LDAA

We now introduce the set of parameters of the VLDAA and we provide a practical expression for each of them. Firstly, table 1 below summarizes all parameters discussed in previous section.

Table 1: Summary of all parameters used in V-LDAA

Parameter	Description
$\boldsymbol{\lambda}$	Level of security
n	Lattice rank
m	Lattice dimension
d	Dimension of $\mathcal{R}$
q	Arithmetic modulus
δ	Basis for gadget matrix
k	Size of q with basis δ
κ	Cardinal of $\mathcal{C}$
σ	Standard deviation for sampling of $\boldsymbol{R}$
ξ	Standard deviation for rejection sampling
β_z	Bound of acceptance
α	Standard deviation for $\boldsymbol{SampleG}$
γ	Upper bound on the spectral norm of α$\boldsymbol{R}$

V-LDAA scheme sets $n = 1, m = 4, k = 2$ and $\delta = \lceil\sqrt{q}\rceil$. That is, the gadget matrix $\boldsymbol{g}$ is a 2-dimension vector equal to $(1 \quad \sqrt{q})$. The dimension of the ring d is a power of two and the arithmetic modulus is $q \equiv 3 \ mod\ 4$.

About standard deviation σ, we use Theorem 4.1 from (Lyubashevsky and *al.*, 2010) which state that, for Ring-LWE instance, σ must be higher or equal to $2\omega(\sqrt{\log(d)})$.In V-LDAA, σ is not used to sample the trapdoor $\boldsymbol{R}$ since coefficient of $\boldsymbol{R}$ are uniformly chosen in $\{-1,0,1\}$ but parameters of gaussian pre-image sampling depend on this $\boldsymbol{\sigma}$.

According to Lemma 2.2 of (Del Pino and *al.*, 2018), we set the standard deviation of rejection sampling to $\xi = 12\kappa\|s\|$ where $\boldsymbol{s}$ is a bounded secret polynomial vector, and the bound of acceptance $\beta_z = \xi\sqrt{2dm}$ to ensure that no information is leaked during the ZKP. Parameter κ is chosen according to the number of elements required to built $\mathcal{C}$. In our implementation κ is such that $|\mathcal{C}| > 2^\lambda$ to ensure that our hash function has, at least, 2^λ elements.

Regarding Gaussian pre-image sampling, we define standard deviation α for **SampleG** and upper bound γ on the spectral norm of α$\boldsymbol{R}$ as in (Bert and *al.*, 2021) *i.e.* $\alpha \geq \sqrt{2\delta}(2\delta + 1)\sqrt{\eta_\epsilon(\mathbb{Z}^k)}$ and $\gamma^2 > (\alpha^2 + 1)s_1(\boldsymbol{R}) + \eta_\epsilon^2(\mathbb{Z}^{nm})$ where $s_1(\boldsymbol{R}) < 1.1\sigma(\sqrt{2d} + \sqrt{dk} + 4.7)$.

The level of security λ is related to parameters $n, \mu = \sigma/q$ and q. λ is the computational complexity to solve LWE problem parametrized by n, μ and q. This writes $\lambda =$ **LWE_Est**$(n, \alpha, \alpha q)$ where **LWE_Est** is the best-known algorithm for solving LWE. n, α and q must be to choose to achieve a level of security greater or equal to λ. We evaluated the security of our instances using LWE-estimator (Albrecht and *al.*, 2015, and dedicated website). We ran the function **estimate_lwe** with the following options:

- Secret distribution = (-1,1)
- Reduction cost model = BKZ.sieve
- Number of LWE-samples: $m = 2d$

Below, we give two practical parameters sets (*i.e.*, respecting conditions listed above) for two security levels λ :

<u>VLDAA-128</u>:

$\lambda \approx 138, d = 512, q = 7583, \delta = 88, \kappa = 120$
$\sigma = 6.0$
$\alpha = 12575.899863196026$
$\gamma = 5702164.533998151$

<u>VLDAA-256</u>:

$\lambda \approx 261, d = 1024, q = 14867, \delta = 122, \kappa = 300$
$\sigma = 6.324555320336759$
$\alpha = 28874.112529324666$
$\gamma = 19125481.447302323$

3 IMPLEM

3.3 Implementation

In this section, we provide a finer analysis of the performance of our software implementation of the V-LDAA and highlight implementation issues. We also compare the results with a pre-quantum ECC-based DAA. Similarly to the original paper, we implemented the scheme using Sagemath. Additionally, we use the library Hashlib for hash primitive (Hashlib, 2001).

Our implementation has 128 bits of precision by default for basic operations such that scalar product or Euclidean norm computation. Table 2 summarizes the main primitives used for the implementation.

Table 2: Summary of all main primitives.

Primitive	Implementation
Underlying schemes	
Zero-knowledge proof	Lyubashevsky, 2012
Commitment	Baum and *al.*, 2018
BlindSign/Verify	Del Pino and *al.*, 2018
Lattice-Cryptography	
TrapGen	Micciancio and Peikert, 2012
SampleZ	Sagemath
SampleGaussian	Guenise and Micciancio, 2019
SampleG	Guenise and Micciancio, 2019
SamplePerturbZ	Guenise and Micciancio, 2019
Rejection Sampling	Lyubashevsky, 2012
SampleInBall	Crystals-dilithium, Ducas and *al.,2018*
General-Cryptography	
Seed generation	TRNG
Random Oracle	SHAKE-128

3.3.1 SampleGaussian

Our Gaussian sampler is based on the MP framework from (Micciancio and Peikert, 2012) instead of Klein sampler (Klein, 2000 and Gentry and *al.*, 2008). The first one is known to be parallelizable and generally faster than Klein but outputs longer vectors. We considered that MP sampler could be more adapted for VANET-based applications. Additionally, it avoids possible precision problems that could be encountered when sampling with Klein sampler, especially with the Gram-Schmidt orthogonalization primitive (Carmona and *al.*, 2023) and (Giraud and *al.*, 2005). Our implementation of MP sampler is based on the work done in (Genise and Micciancio, *2018*) which propose an algorithm adapted to ring lattice *i.e.* which takes advantage of the algebraic structure. ring structure. It also calls the primitive $\mathbf{SampleZ}(\boldsymbol{\sigma}, \boldsymbol{c})$ which outputs an integer from $\mathcal{D}_{\mathbb{Z},\sigma,c}$ and is implemented by Sagemath.

3.3.2 Zero-knowledge proof

For the zero-knowledge proof in $\mathbf{Join}$ procedure, we implemented the "Fiat-Shamir with Abort" ZKP scheme from (Lyubashevsky, 2012). We admit that one could use the scheme described in (Ling and *al.*, 2013) proving the exact knowledge of a secret since, even if it is not very efficient, the proof is done one time. We justify this choice by the fact that the general framework of (Lyubashevsky, 2012) is also used on the others digital signature of the V-LDAA : $\mathbf{BlindSign}$ and $\mathbf{Sign}$ as well as the lattice-based digital signature Crystals-Dilithium which has been recently selected by the NIST (Ducas et *al.*, 2018 and NIST, 2024) to be standardized. To generate a polynomial challenge, we implemented the algorithm $\mathbf{SampleInBall}$ used in Crystals-Dilithium.

3.3.2 Commitment

For the commitment, we use the same method used by the authors of the V-LDAA *i.e.* the technique from in (Baum and *al.*, 2018) introduced in the previous section. We emphasize on a particularity met on the implementation of the commitment applied to the V-LDAA during the $\mathbf{Create}$. The framework implies to create two commits $t = Com(i, \boldsymbol{r})$ and $t' = Com(i\sqrt{q}, \boldsymbol{r}')$. Let $\boldsymbol{A_2}$ and $\boldsymbol{A_2'}$, the vectors in $\mathcal{R}_q^{1\times 4}$of the form $[0 \quad 1 \quad \mathcal{R}_q \quad \mathcal{R}_q]$ (as defined in section 3.2) and respectively associated to t and t'. Using equation (2) to recover equation (1), it's easy to see that $\boldsymbol{A_2}$ and $\boldsymbol{A_2'}$ need to be equal to $-[\boldsymbol{a_T} \ \ \boldsymbol{a_H}]$ that contradicts the fact that those two matrix are generated as $[0 \quad 1 \quad \mathcal{R}_q \quad \mathcal{R}_q]$. This issue is not mentioned in the original paper.

The $\mathbf{BlindSign/Verify}$ protocol is implemented as described in (Del Pino and *al.*, 2018).

3.4 Performances analysis

We give a performance analysis of the timing of each primitive of V-LDAA for two sets of parameters. First, we compare our implementation with the one provided in (Chen and *al.*, 2021) with the same instances: $d = 128, q = 114356107 \approx 2^{27}$. According to LWE-estimator, the security level is $\lambda \approx 42$. Secondly, we tested our implementation with a more practical (regarding security) set achieving 128 bits of security.

The implementation run on a Core i5-8265U CPU @1.60GHz. To get execution time, we took the average of the times recorded on 50 executions for each primitive.

Table 3 below provides the execution time of all primitives of the scheme. One can observe that our implementation speeds up the $\mathbf{Join}$ procedure by 85% and the $\mathbf{Create}$ procedure by almost 80% regarding previous work.

Table 3: Execution time of V-LDAA for two sets of parameters and performance comparison with implementation from (Chen and *al.*, 2021).

	Time (in seconds)		
	(Chen and *al.*, 2021)	Our implementation	
Parameters	$n = 128$ $q = 2^{27}$ $\lambda \approx 44$		**VLDAA-128**
Setup	-	0.022	0.043
Join	7.45	1.19	72.05
Create	5.41 (Blindsign)	1.13	5.11
Sign	0.030	0.030	0.087
Verify	0.047 (BlindVerify)	0.069	0.68
Revoke	-	1.41 + T_{enc}	2.77 + T_{enc}

Benchmark shows that the slower primitive is $\mathbf{Join}$. The bottleneck comes from the gaussian sampler and especially from the generation of perturbation. This is due to the number of operations on polynomials (multiplication and inverse) executed in the ring $\mathbb{R}[X]/\langle X^d + 1\rangle$ of real-coefficients polynomial. However, in practice, the procedure $\mathbf{Join}$ is called one time and can be executed offline. Additionally, the algorithm used for gaussian sampling is parallelizable.

Execution time of primitive $\mathbf{Create}$ is versatile because of the number of polynomial multiplications and rejection sampling in $\mathbf{BlindSign}$. Using parameters from **VLDAA-128**, we registered a standard deviation of 5.05 seconds for this primitive.

$\mathbf{Sign}$ and $\mathbf{Verify}$ are the two primitives that have to be implemented in an embedded (*i.e.* resources-constrained) environment and executed several times. For **VLDAA-128,** in signing operation, polynomial error sampling takes 64.81% of the total time, rejection sampling takes 20.09%, hashing takes 12.09% and multiplications takes 3.01%. $\mathbf{Verify}$ is slower than $\mathbf{Sign}$ since two verifications are required: on the message signature and on the signature of credentials generated by $\mathbf{Blindverify}$. The latter takes most of the times in the verification due to the sixteen multiplications that takes 85.6% of the total time.

Regarding the structure of these two primitives it is natural to find synergies with primitives Sign and Verify of DILITHIUM (Ducas et *al.*, 2018). Both schemes sample a vector of polynomial to mask the secret vector multiplied by a public matrix. However, due to the specific structure of the ring and particular choice of $q \equiv 3 \, mod \, 4$ makes the V-LDAA not compliant with the NTT for accelerating multiplication operations (NTT requires $q \equiv 1 \, mod \, 2d$). Other acceleration strategies about DILITHIUM such as seed extension, sampling in a ball, hash function are possible. This will have a strong impact for implementations using hardware.

The execution time of $\mathbf{Revok}$ depends on T_{enc}, the time used to encrypt the message to send to all users for the revocation. The reference papers did not specify the algorithm used.

3.5 Benchmark with pre quantum DAA

We compare our implementation of the V-LDAA with a pre-quantum ECC-based DAA proposed in (Yang and *al.*, 2021) and compliant with the TPM 2.0 specifications (Trusted Computing Group, 2014). Benchmark of this ECC-DAA has been evaluated on a 1.80GHz Intel Core i7-8550U CPU (host) paired with an Infineon TPM 2.0. The implementation is in C and uses the library ACML. TPM 2.0 specifications include two pairing EC (Trusted Computing Group, 2014): BN_P256 and BN_P638. The first provides ≈ 100 bits of security but cryptanalysis shows that BN_P256 curve is no longer secure (Barbulescu and *al.*, 2018). The second guarantees 128 bits of security but is not implemented on Infineon TPM 2.0.

Table 4: Comparison of the execution time of sign/verify operation and size of credentials/signature between our implementation of V-LDAA and ECC-based DAA of (Yang and *al.*, 2021).

	VLDAA-128	**BN_P256**	**BN_P636**
	Size (in bytes)		
Credentials	44,142	193	479
Sign. size	431,254	705 + $o(1)$	1,800 + $o(1)$
	Time (in ms)		
Sign time	87	137	-
Verif. Time	680	81	-

The size of credentials and signature remain very large compared to those of the EC-DAA. The signature size includes the signature of the message ($\approx 25\ kB$) and the attestation signature (≈ 406 kB). This confirms a general impact of post-quantum cryptography on the instance size increasing regarding pre-quantum cryptography The sign operation is 37% faster in V-LDAA than in EC-DAA with BN_P256 but the verify much slower in the lattice-based case.

4 CONCLUSIONS

We proposed a software implementation of the most recent V-LDAA and a complete parametrization of the scheme. The main bottleneck remains the non-compatibility of the NTT with the algebraic structure of the V-LDAA to improve polynomial multiplication. Size of the instances also represents a huge challenge especially regarding to the size of attestation signature.

This work is a first step toward the optimization of V-LDAA implementation to achieve TPM requirements in terms of resources. The synergy with DILITHIUM regarding the Sign and Verify could bring strong enabler to accelerate the LDAA in hardware.

REFERENCES

Agrawal, S., Boneh, D., & Boyen, X. (2010). Efficient lattice (H) IBE in the standard model. In Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29 (pp. 553-572). Springer Berlin Heidelberg.

Albrecht, R.M., Player, R., Scott, S. (2015). On the concrete hardness of Learning with Errors, In *Journal of Mathematical Cryptology 2015*. Related website: https://lwe-estimator.readthedocs.io/en/latest/

Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., ... & Vaikuntanathan, V. (2021). Homomorphic encryption standard. *Protecting privacy through homomorphic encryption*, 31-62.

Bansarkhani, R. E., and Kaafarani, A. E. (2017) Direct anonymous attestation from lattices. In Cryptology ePrint Archive, Report 2017/1022.

Barbulescu, R., & Duquesne, S. (2019). Updating key size estimations for pairings. Journal of cryptology, 32, 1298-1336.

Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., & Peikert, C. (2018, August). More efficient commitments from structured lattice assumptions. In *International Conference on Security and Cryptography for Networks* (pp. 368-385). Cham: Springer International Publishing.

Bert, P., Eberhart, G., Prabel, L., Roux-Langlois, A., & Sabt, M. (2021). Implementation of lattice trapdoors on modules and applications. In *Post-Quantum Cryptography: 12th International Workshop, PQCrypto 2021, Daejeon, South Korea, July 20–22, 2021, Proceedings 12* (pp. 195-214). Springer International Publishing.

Brickell, E., Camenisch, J., and Chen, L. (2004). Direct anonymous attestation. In ACM Conference on Computer and Communications Security, CCS 2004, ACM.

Brickell, E., Chen, L., and Li, J. (2008). A new direct anonymous attestation scheme from bilinear maps. In Trusted Computing - Challenges and Applications, pp. 166–178.

Brickell, E., Chen, L., and Li, J. (2009). Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. Int. J. Inf. Sec. 8, 5, 315–330.

Chen, L. (2010). A DAA scheme requiring less TPM resources. In Information Security and Cryptology: 5th International Conference, Inscrypt 2009, Beijing, China, December 12-15, 2009. Revised Selected Papers 5 (pp. 350-365). Springer Berlin Heidelberg.

Chen, L., Ng, S. L., and Wang, G. (2011). Threshold anonymous announcement in VANETs. *IEEE Journal on selected areas in communications*, *29*(3), 605-615.

Chen, L., Page, D., & Smart, N. P. (2010, April). On the design and implementation of an efficient DAA scheme. In International Conference on Smart Card Research and Advanced Applications (pp. 223-237). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, L., Tu, T., Yu, K., Zhao, M., and Wang, Y. (2021). V-ldaa: A new lattice-based direct anonymous attestation scheme for vanets system. *Security and Communication Networks*, *2021*, 1-13.

Del Pino, R., Lyubashevsky, V., & Seiler, G. (2018, October). Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *Proceedings of the 2018 ACM SIGSAC conference* *on computer and communications security* (pp. 574-591).

Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., & Stehlé, D. (2018). Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 238-268.

El Kassem, N., Chen, L., Bansarkhani, R. E., Ali El Kaafarani, J. C., Hough, P., Martins, P. S. A., , and Sous, L. (2019) More efficient, provably-secure direct anonymous attestation from lattices. In Future Generation Computer Systems.

El Kassem, N., Chen, L., El Bansarkhani, R., El Kaafarani, A., Camenisch, J., Hough, P., ... & Sousa, L. (2019). More efficient, provably-secure direct anonymous attestation from lattices. Future Generation Computer Systems, 99, 425-458.

Genise, N., & Micciancio, D. (2018). Faster Gaussian sampling for trapdoor lattices with arbitrary modulus. In *Advances in Cryptology–EUROCRYPT 2018: 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29-May 3, 2018 Proceedings, Part I 37* (pp. 174-203). Springer International Publishing.

Gentry, C., Peikert, C., & Vaikuntanathan, V. (2008, May). Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing* (pp. 197-206).

Giraud, L., Langou, J., Rozloznik, M. (2005). The Loss of Orthogonality in the Gram-Schmidt Orthogonalization Process, In Computers & Mathematics with Applications.

He, D., Zeadally, S., Xu, B., & Huang, X. (2015). An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *IEEE Transactions on Information Forensics and Security*, *10*(12), 2681-2691.

International Organization for Standardization. ISO/IEC 20008-2: Information technology - Security techniques - Anonymous digital signatures – Part 2: Mechanisms using a group public key, 2013.

International Organization for Standardization. ISO/IEC 11889: Information technology - Trusted platform module library, 2015.

Klein, P. (2000, February). Finding the closest lattice vector when it's unusually close. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms* (pp. 937-941).

Ling, S., Nguyen, K., Stehlé, D., & Wang, H. (2013, February). Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *International workshop on public key cryptography* (pp. 107-124). Berlin, Heidelberg: Springer Berlin Heidelberg.

Micciancio, D., & Peikert, C. (2012, April). Trapdoors for lattices: Simpler, tighter, faster, smaller. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 700-718). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lyubashevsky, V., & Micciancio, D. (2006, July). Generalized compact knapsacks are collision resistant. In International Colloquium on Automata, Languages, and Programming (pp. 144-155). Berlin, Heidelberg: Springer Berlin Heidelberg.

Lyubashevsky, V., Peikert, C., & Regev, O. (2010). On ideal lattices and learning with errors over rings. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29* (pp. 1-23). Springer Berlin Heidelberg.

Lyubashevsky, V. (2012, April). Lattice signatures without trapdoors. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 738-755). Berlin, Heidelberg: Springer Berlin Heidelberg.

National Institute of Standards and Technology (NIST). Module-Lattice-Based Digital Signature Standard, 2024.

Petit, J., Schaub, F., Feiri, M., & Kargl, F. (2014). Pseudonym schemes in vehicular networks: A survey. *IEEE communications surveys & tutorials*, *17*(1), 228-255.

Smyth, B., Ryan, M., & Chen, L. (2012). Formal analysis of anonymity in ECC-based Direct Anonymous Attestation schemes. In Formal Aspects of Security and Trust: 8th International Workshop, FAST 2011, Leuven, Belgium, September 12-14, 2011. Revised Selected Papers 8 (pp. 245-262). Springer Berlin Heidelberg.

Trusted Computing Group. TPM main specification version 1.2, 2004.

Trusted Computing Group. Trusted platform module library specification, family "2.0", 2014.

Yang, K., Chen, L., Zhang, Z., Newton, C. J., Yang, B., & Xi, L. (2021). Direct anonymous attestation with optimal tpm signing efficiency. IEEE transactions on information forensics and security, 16, 2260-2275.