This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

Sparse Matrix Masking-based Non-Interactive Verifiable (Outsourced) Computation, Revisited

Liang Zhao, *Member, IEEE,* Liqun Chen, *Member, IEEE*

Abstract—A Privacy-preserving Verifiable (outsourced) Computation (PVC) protocol enables a resource-constrained client to outsource expensive and sensitive workloads to computationally powerful but possibly untrusted service providers (called workers) and to verify the correctness of the results. In a PVC protocol, the inputs and outputs of the computation are hidden, so that the worker is unable to determine them. This is referred to as the privacy property. A Non-interactive PVC (NPVC) protocol is a PVC protocol without communications between a client and worker, apart from distributing the workloads and results. In the literature, Sparse Matrices (SMs) have been used in NPVC protocols to efficiently hide the inputs and outputs from the worker. However, to the best of our knowledge, how the low density of an SM affects privacy in such NPVC protocols has not been formally analyzed. In this work, we first propose a formal definition of the privacy property of an NPVC protocol with respect to matrix density. We use this definition to demonstrate that all of the SM masking-based NPVC protocols that we know of do not hold this privacy property under the ciphertext-only attack model. We then propose an SM masking construction to modify two of those protocols, chosen because they are state-of-the-art, and prove that the modified protocols hold the privacy property under the chosen-plaintext attack model. Our modifications do not require the client operating matrix inversion, and they are able to keep the same level of high performance and all other properties as the originals.

Index Terms—Privacy-preserving verifiable (outsourced) computation, matrix-related computation, privacy with respect to density.

1 Introduction

1.1 Background

In recent years, with the broad applications of mobile and resource-constrained devices, outsourcing computation, which allows a computationally weak client to delegate some expensive computation to a more computationally powerful worker, has become increasingly fascinating. Many real-world scenarios make use of this computation paradigm, e.g., the delegation of computations for weak mobile devices, volunteer computing and cloud computing (examples can be found in [8], [9], [16]).

Although outsourcing computation is economically sensible for the computationally weak client, some intrinsic concerns have emerged and should be solved carefully. A significant and natural problem is whether the returned result can be trusted. On the one hand, hardware faults and software bugs on the worker side may cause the computation incorrect. On the other hand, an adversarial or selfish worker may inject errors into the computation on purpose or reduce the usage of computational resources for achieving the correct result due to financial considerations and simply send back a plausible result that needs less computation. This anxiety by the client is based on the fact that the process of the outsourced computation is not under the client's control and can be eliminated by checking the correctness of the returned result to guarantee the integrity of the whole computation. In [16], Gennaro et al. defined this concern as the *correctness and security* of outsourcing computation.

The next crucial problem is whether the sensitive and valuable input and output can be learned by a worker who is curious or premeditated. Because the client's data may be extremely confidential, e.g., the business secret of a company or the observation data of a research institute, the worker may sell this information to a competitor or opponent based on financial incentives or store it for the future usage. The solution to this problem is hiding the actual data of the input and output from the worker. According to [16], this concern is called the *privacy* of outsourcing computation. While correctness and security is the most significant requirement for outsourcing computation, privacy is the crucial issue for individuals and some business companies.

In order to solve the above problems, the client must spend some time doing local computation. However, this time should not be larger than the time needed for conducting the outsourced computation locally. This property is essential for the outsourcing computation and is defined as the *efficiency* in [16].

1.2 Related Work

Given the above concerns related to outsourcing computation, a feasible technique called Privacy-preserving Verifiable (outsourced) Computation (PVC), which can be either *interactive* or *non-interactive*, has drawn many researchers' attention. A large number of constructions for solving various functions have been proposed by two main communities: the cryptography community (including the theory community) and the information security community.

- *L. Zhao is with the College of Cybersecurity, Sichuan University, Chengdu, China, 610065, and with the HIFIVE Lennon Laboratory, Chengdu HiFive Technology Co., Ltd., Chengdu, China, 610094. L. Zhao is the corresponding author.*
E-mails: zhaoliangjapan@scu.edu.cn; liang.zhao@surrey.ac.uk.
- *L. Chen is with the Department of Computer Science, University of Surrey, Guildford, Surrey, U.K., GU2 7XH.*
E-mail: liqun.chen@surrey.ac.uk.

Manuscript received January 10, 2018; revised April 26, 2018.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

The cryptography community has long been researching securely and privately outsourcing the computations of expensive operations to untrusted/semi-trusted workers. In [13], Chaum and Pedersen proposed the concept of wallets with observers, which is secure hardware. Later, some of the work in this community began focusing on verifying specific functions [5], [6], [10], [11], [12], [18], [28], [29], [30]. Meanwhile, theory researchers in the community have been exploring PVC protocols for *any* function [8], [9], [16], [17]. In general, the theoretical work depends on an elegant mechanism, i.e., Fully Homomorphic Encryption (FHE). Although such theoretical results are tempting, until now, they have not been regarded as acceptable protocols for use in practice. This is because hundreds to trillions of years need to be spent for verification even for small cases [23]. Of course, other researchers have been developing a solution by avoiding the above heavier mechanism, e.g., Ananth et al.′s proposals [1], which are based on one-way functions or the decisional Diffie-Hellman assumption.

For the information security community, the proposals on the PVC protocol emphasize the efficiency and convenience of the implementation for specific purposes, which implies that such solutions can be deployed in practice immediately. Moreover, they generally employ the one-time pad mode in their constructions to enhance privacy [14]. Earlier work was conducted by Atallah et al. [2], who introduced a privacy-preserving framework for the computationally weak client to outsource numerical and scientific computations to an external agent. Achieving the privacy property is based on the application of the lightweight disguise technique. Later, a large body of work [3], [4], [7], [14], [15], [20], [21], [22], [24], [25], [26], [27], [31] focused on securely and privately outsourcing computations of various specific functions to untrusted/semi-trusted workers. For example, efficiently verifying solutions of large-scale systems of Linear Equations (LE) [7], [15], [24], [26], [27], the Linear Regression (LR) computation [14], the Matrix Multiplication (MM) computation [20], [31], the Matrix Determinant (MD) computation [21] and the Matrix Inversion (MI) computation [22] have been investigated recently. Note that, all of these computations can be collectively referred to as the matrix-related computation that is a basic computational task rooted in scientific and engineering fields, e.g., mathematical physics, statistics, and image processing.

According to the recent work from the information security community, the Non-interactive PVC (NPVC) protocol employing the Sparse Matrix Masking (SMM) technique [7], [14], [15], [20], [21], [22] has been a main research topic of the PVC and used for the outsourcing of the matrix-related computation (e.g., the LE solving and MD computation). The key point of the SMM technique is that the client′s data are masked by some *randomly* chosen sparse matrices (see Definition 6). Using the SMM technique, a client can avoid running expensive cryptographic operations and protect the outsourced data efficiently. Thus, the SMM-based NPVC protocol is practical for real-world computations, e.g., the widely applicable matrix-related computation.

Based on the different properties of sparse matrices used in the SMM technique, the SMM-based NPVC protocols can be divided into two types. For the *first* type, the sparse matrix employed by a protocol holds the property that there exists *only* one nonzero element in a row (or a column). Specifically, the merit of such a sparse matrix is that its inverse matrix satisfying the above property can be directly obtained. The SMM-based NPVC protocols [14], [15], [20], [22] belong to this type. In these protocols, two sparse matrices are used to efficiently mask a plaintext matrix. This implies that achieving privacy in these protocols is based on the application of the SMM technique. Recently, Lei et al. [21] combined the SMM technique with the block matrix technique and Lower-Upper (LU) decomposition to design an NPVC protocol. Most notably, Lei et al. believe that the combination of the SMM technique and the block matrix technique guarantees the privacy of the protocol. Until our work, to the best of our knowledge, the above combination is the state-of-the-art data protection mechanism used in the first type of protocol. For the *second* type, the sparse matrix used in a protocol holds the property that there exists *at least* one nonzero element in a row (or a column). Specifically, the product of two such sparse matrices can be a dense matrix. Chen et al.′s SMM-based NPVC protocol [7] belongs to this type. In this interesting protocol, two random sparse matrices are used to efficiently mask a nonsingular plaintext matrix. Then, achieving privacy in this protocol is based on the application of the SMM technique. Moreover, neither of participants of this protocol needs to run the matrix inversion, which guarantees the efficiency of the protocol. The used verification algorithm performs a checking process that can succeed with probability 1. For more details on Chen et al.′s protocol, please see Subsection 4.2.

1.3 Open Problem and Our Contributions

Regarding privacy research on SMM-based NPVC protocols, two natural questions to ask are *whether a suitable formal definition related to the privacy property of NPVC has been given in the previous work?* and *whether all such solutions hold the privacy property if its formal definition is given?* In this work, we address these two questions.

To our best knowledge, there is no a suitable formal definition of the privacy property of the SMM-based NPVC protocol in the literature. Such a definition is essential for a proper privacy analysis. We first define the concept of *privacy with respect to (matrix) density* (see Definition 8 and 9), which can be regarded as a *basic* privacy property of the NPVC protocol for a matrix-related computation. Then, based on the definition of privacy with respect to density against passive eavesdropping (i.e., a Ciphertext-Only Attack (COA) [1]) (see Definition 8), we show that all of the SMM-based NPVC protocols that we know of [7], [14], [15], [20], [21], [22] do not hold this privacy property. Specifically, as presented before, since the protocols [7], [21] are state-of-the-art among the targeted solutions, we give the detailed analyses on the privacy property of these two protocols (see Section 5). Our analysis also solves an open problem concerning privacy remained in [7]: *Is the distribution* $\{A'_0{=}M_1{\cdot}A_0{\cdot}M_2\}$ *computationally indistinguishable from the distribution* $\{A'_1{=}M'_1{\cdot}A_1{\cdot}M'_2\}$*, where* A_0 *is a nonsingular sparse matrix,* A_1 *is a nonsingular dense matrix, and* M_1, M_2, M'_1 *and*

[1] According to Katti et al.′s work [19], privacy against passive eavesdropping is equivalent to IND-COA privacy.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

M_2' are random nonsingular sparse matrices? [2] *How about the case that A_0 is a nonsingular dense matrix?*

We *continue* the line of research on the second type of the SMM-based NPVC protocol and propose an SMM construction to modify the protocols for the outsourcing of MD computation [21] and LE solving [7]. To build our construction, *two interesting primitives* are adopted: the χ-sparse matrix pair and the adaptive adjustment function. In particular, a χ-sparse matrix only has *at most three* nonzero elements in a row (or a column), and it may be not a row diagonally dominant matrix. The adaptive adjustment function is used to control the masking process according to a nonsingular plaintext matrix. By using our SMM construction, the modified NPVC protocols (see Subsection 6.3) not only retain *all merits* of the originals (i.e., correctness, security and efficiency) but also hold privacy with respect to density against a Chosen-Plaintext Attack (CPA) (see Definition 9). This is very important for our modifications. Note that, in the modified NPVC protocols, the client does *not* need to operate the matrix inversion.

Furthermore, to confirm our contributions, the experimental results are listed in Table 1, 2, 4, and 5 and Figure 1 and 2. Specifically, Figure 1 shows the client's cost comparisons between the outsourced computations (i.e., the MD computation and LE solving), the original NPVC protocols and the modified NPVC protocols, visually.

1.4 Outline of this Paper

The remainder of this paper is organized as follows: After recalling the standard definitions of the NPVC protocol with its properties, and presenting the definition of the SMM technique in Section 2, we introduce in Section 3 our formal definitions about the NPVC protocol for a matrix-related computation and its basic privacy property. We then briefly review the constructions of the SMM-based NPVC protocols [7], [14], [15], [20], [21], [22] in Section 4. In Section 5, we give detailed privacy analyses of the protocols [7], [21] and the simulation results. Then, we present an SMM construction to modify the protocols [7], [21] and also provide the property analyses of the modified protocols in Section 6. Finally, the concluding remarks are given in Section 7.

2 Preliminaries

In this section, we present the classical concepts for an NPVC protocol and its properties. We also introduce the basic definition of the SMM technique.

Notations. Let the client and worker be denoted by $\mathcal{C}$ and $\mathcal{W}$, respectively. Let lower-case letters ($\vec{x}$, $\vec{y}$, ...) be vectors and upper-case letters (X, Y, ...) be matrices. Let $\lambda \in \mathbb{N}_+$ be a security parameter, $\mathsf{poly}(\lambda)$ be the class of polynomial functions in λ, $\mathsf{negl}(\lambda)$ be some unspecified negligible function in λ and $\mathsf{posi}(\lambda)$ be the class of positive functions in λ (i.e., for any λ, $\mathsf{posi}(\lambda) > 0$). For an $n_1 \times n_2$ matrix X, we denote the number of zero-valued elements in X by N_0^X, the number of nonzero elements of the i^{th} row in X by $N_{nz}^{X(i,*)}$, the number of nonzero elements of the j^{th} column in X by $N_{nz}^{X(*,j)}$, and the maximum number of $N_{nz}^{X(i,*)}/N_{nz}^{X(*,j)}$ in X by $N_{ma\text{-}nz}^X$. We denote the transpose of x by x^t, the determinant of a matrix X by $\det(X)$ and the inverse matrix of X by X^{-1}. Finally ($x \overset{\$}{\leftarrow} \Psi$) indicates choosing x uniformly at random in a set Ψ.

2.1 Classical Definitions of Previous Work

We first recall Gennaro et al.'s formal definition of an NPVC protocol between $\mathcal{C}$ and $\mathcal{W}$ and its properties of correctness, security, privacy and efficiency in [16] as follows:

2.1.1 Syntax

Definition 1 (NPVC Protocol [16]). Given a security parameter λ, an outsourced function *f* with input x and output y where y=*f*(x), an NPVC protocol NPVC_f consists of the algorithms (Setup, KeyGen, $\mathsf{ProbGen}$, $\mathsf{Compute}$, Verify):

- $(pk, sk) \leftarrow \mathsf{Setup}(\lambda, f)$. Given λ and a function *f*, $\mathcal{C}$ produces a key pair (pk, sk). In particular, this pair (pk, sk) corresponds to *f*, where the public key *pk* encodes *f*, and the matching secret key *sk* is kept private by $\mathcal{C}$.
- $k \leftarrow \mathsf{KeyGen}(1^\lambda)$. Given λ, $\mathcal{C}$ generates a new value k corresponds to a function input x [3]. If $k=(\tau_x^{pv}, \tau_x^{sv})$, τ_x^{pv} and τ_x^{sv} are a public value and a secret value held private by $\mathcal{C}$.
- $(\sigma_x, \tau_x^{pv}) \leftarrow \mathsf{ProbGen}_{sk}(x, k)$. Upon input of a function input x and the value $k=(\tau_x^{pv}, \tau_x^{sv})$, $\mathcal{C}$ employs *sk* to encode x as a public value σ_x with the assistance of some primitive using τ_x^{pv}. Then, (σ_x, τ_x^{pv}) is sent to $\mathcal{W}$ for the computation.
- $\sigma_y \leftarrow \mathsf{Compute}_{pk}(\sigma_x, \tau_x^{pv})$. Given the public key *pk*, the public value τ_x^{pv} and the encoded input σ_x, $\mathcal{W}$ computes the encoded output σ_y, which is associated with the real output y=*f*(x).
- $y \cup \perp \leftarrow \mathsf{Verify}_{sk}(\tau_x^{sv}, \sigma_y)$. Upon input of the secret key *sk*, the secret value τ_x^{sv} and the encoded output σ_y, $\mathcal{C}$ invokes this verification algorithm to recover a real output of the function, i.e., y=*f*(x), or produce an error $\perp$, which indicates that σ_y is a false and invalid result.

Remark 1. Setup belongs to the pre-processing phase that is an expensive one-time step. The key pair (pk, sk) is fixed for a function *f* and can be reused numerous times. Since the generation of (pk, sk) does not save on $\mathcal{C}$'s cost, Setup is seen as an amortized phase that is meaningful only in the situation that $\mathcal{C}$ plans to use the identical function *f* many times [16]. Actually, for many specific-purpose-oriented solutions (e.g., [7], [14], [15], [20], [21], [22]), Setup is not employed. Such solutions are efficient for $\mathcal{C}$ and can be deployed in practice immediately. So, we treat it to be optional for constructing an NPVC protocol.

An NPVC protocol should hold the following properties.

2.1.2 Correctness and Security

Intuitively, an NPVC protocol is correct iff the public value generated by $\mathsf{ProbGen}$ allows the honest $\mathcal{W}$ to produce the encoded value that passes Verify and corresponds to the real output of the target function *f* on the input successfully.

Definition 2 (Correctness [16]). An NPVC protocol is correct if, for any function *f*, Setup produces the key pair by $(pk, sk) \leftarrow \mathsf{Setup}(\lambda, f)$ such that, $\forall x \in \mathsf{Domain}(f)$, the computations $k \leftarrow \mathsf{KeyGen}\ (1^\lambda)$, $(\sigma_x, \tau_x^{pv}) \leftarrow \mathsf{ProbGen}_{sk}(x, k)$

[2] See Remark 6 of [7] on this statement.

[3] In [16], the concrete NPVC protocol employs the key generation algorithm from an FHE scheme to generate a new value k

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

and $\sigma_y \leftarrow \mathsf{Compute}_{pk}(\sigma_x, \tau_x^{pv})$ can guarantee that $y \leftarrow \mathsf{Verify}_{sk}(\tau_x^{sv}, \sigma_y)$, where y=$f(x)$.

The security of an NPVC protocol means that a malicious $\mathcal{W}$ always fails to convince Verify to accept and output an invalid result. If a formal definition is presented, most notably, the adversary $\mathcal{W}_{\mathcal{A}}$ (e.g., the malicious $\mathcal{W}$) is given oracle access to produce encodings on multiple inputs.

Definition 3 (Security [16]). For an NPVC protocol NPVC_f, the following experiment associated with a Probabilistic Polynomial-Time (PPT) adversary $\mathcal{W}_{\mathcal{A}}$ is considered:
Experiment $\mathsf{Exp}_{\mathcal{W}_{\mathcal{A}}}^{\text{secure}}[\mathsf{NPVC}_f, f, \lambda]$:

```
$(pk, sk) \xleftarrow{\$} \mathsf{Setup}(\lambda, f)$;
For $i$=1,...,$\ell$
   $x_i \leftarrow \mathcal{W}_{\mathcal{A}}(1^{\lambda}, pk, x_1, (\sigma_{x_1}, \tau_{x_1}^{pv}), \ldots, x_{i-1}, (\sigma_{x_{i-1}}, \tau_{x_{i-1}}^{pv}))$;
   $(\tau_{x_i}^{pv}, \tau_{x_i}^{sv}) \leftarrow \mathsf{KeyGen}(1^{\lambda})$;
   $(\sigma_{x_i}, \tau_{x_i}^{pv}) \leftarrow \mathsf{ProbGen}_{sk}(x_i, (\tau_{x_i}^{pv}, \tau_{x_i}^{sv}))$;
$(i, \sigma'_y) \leftarrow \mathcal{W}_{\mathcal{A}}(pk, x_1, (\sigma_{x_1}, \tau_{x_1}^{pv}) \ldots, x_{\ell}, (\sigma_{x_\ell}, \tau_{x_\ell}^{pv}))$;
$y' \leftarrow \mathsf{Verify}_{sk}(\tau_{x_i}^{sv}, \sigma'_y)$;
If $y' \notin \{\perp, f(x_i)\}$, output 1; else, output 0;
```

Such an NPVC protocol NPVC_f is secure if, for any function f, any $\mathcal{W}_{\mathcal{A}}$ and any ℓ=$\mathsf{poly}(\lambda)$, there exists a negligible function negl such that

$$\Pr[\mathsf{Exp}_{\mathcal{W}_{\mathcal{A}}}^{\text{secure}}[\mathsf{NPVC}_f, f, \lambda] = 1] \leq \mathsf{negl}(\lambda).$$

2.1.3 Privacy Against A Chosen-Plaintext Attack

An NPVC protocol is private if a malicious $\mathcal{W}$ cannot distinguish the ProbGen outputs over two distinct inputs. Based on the CPA model, a formal definition is as follows:

Definition 4 (Privacy Against A Chosen-Plaintext Attack [16]). For an NPVC protocol NPVC_f, an experiment associated with a PPT adversary $\mathcal{W}_{\mathcal{A}}$ is considered:
Experiment $\mathsf{Exp}_{\mathcal{W}_{\mathcal{A}}}^{\text{priv}}[\mathsf{NPVC}_f, f, \lambda]$:

```
$(pk, sk) \xleftarrow{\$} \mathsf{Setup}(\lambda, f)$;
$(x_0, x_1) \leftarrow \mathcal{W}_{\mathcal{A}}^{\mathsf{PubProbGen}_{sk}}(1^{\lambda}, pk)$;
$b \xleftarrow{\$} \{0, 1\}$;
$(\tau_x^{pv}, \tau_x^{sv}) \leftarrow \mathsf{KeyGen}(1^{\lambda})$;
$(\sigma_{x_b}, \tau_x^{pv}) \leftarrow \mathsf{ProbGen}_{sk}(x_b, (\tau_x^{pv}, \tau_x^{sv}))$;
$b' \leftarrow \mathcal{W}_{\mathcal{A}}^{\mathsf{PubProbGen}_{sk}}(pk, x_0, x_1, (\sigma_{x_b}, \tau_x^{pv}))$;
If $b'$=$b$, output 1; else, output 0;
```

where σ_{x_b} refers to the challenge ciphertext. The oracle $\mathsf{PubProbGen}_{sk}$ first asks KeyGen to obtain a k=$(\tau_x^{pv}, \tau_x^{sv})$ and asks $\mathsf{ProbGen}_{sk}$ to produce (σ_x, τ_x^{pv}) and then simply sends back the encoded value σ_x and the public value τ_x^{pv}. The output from $\mathsf{PubProbGen}_{sk}$ is probabilistic. We define the advantage of $\mathcal{W}_{\mathcal{A}}$ in the experiment above as:

$$\begin{aligned}&\mathsf{Adv}_{\mathcal{W}_{\mathcal{A}}}^{\text{priv}}(\mathsf{NPVC}_f, f, \lambda)\\ &= \left|\Pr[\mathsf{Exp}_{\mathcal{W}_{\mathcal{A}}}^{\text{priv}}[\mathsf{NPVC}_f, f, \lambda] = 1] - \frac{1}{2}\right|\end{aligned}.$$

NPVC_f is IND-CPA private if, for any function f and any $\mathcal{W}_{\mathcal{A}}$, there exists a negligible function negl such that

$$\mathsf{Adv}_{\mathcal{W}_{\mathcal{A}}}^{\text{priv}}(\mathsf{NPVC}_f, f, \lambda) \leq \mathsf{negl}(\lambda).$$

2.1.4 Efficiency

Naturally, an NPVC protocol can appear advantageous for $\mathcal{C}$ when the execution time required for producing the encoded input and verifying and achieving the real output is smaller than the time required for computing the function f from scratch. Note that, if Setup is developed, the total execution time does not include the time required for computing it, which is attributed to the amortized sense from the intrinsic idea of the NPVC [16].

Definition 5 (Efficiency [16]). An NPVC protocol NPVC_f is efficient for any function f iff, for any input $x \in \mathsf{Domain}(f)$ and any output σ_y, the sum of the time spent by $\mathsf{KeyGen}(1^{\lambda})$ and $\mathsf{ProbGen}_{sk}(x, (\tau_x^{pv}, \tau_x^{sv}))$ to encode x and the time spent by $\mathsf{Verify}_{sk}(\tau_x^{sv}, \sigma_y)$ to verify σ_y is o(T), where T is the shortest known time for running y=$f(x)$.

2.2 Definition of Sparse Matrix Masking

We now give a formal definition about the SMM technique. Intuitively, for an $n_1 \times n_2$ matrix X, if it is protected by the SMM technique, this means that it is (multiplicative) masked by some randomized invertible sparse matrices of respective sizes $n_1 \times n_1$ and $n_2 \times n_2$.

Definition 6 (Sparse Matrix Masking). Let X be an $n_1 \times n_2$ plaintext matrix. Given two randomized invertible sparse matrices M_1 and M_2 of respective sizes $n_1 \times n_1$ and $n_2 \times n_2$, where $N_{ma\text{-}nz}^{M_1} \ll n_1$ and $N_{ma\text{-}nz}^{M_2} \ll n_2$, the matrix X is mapped into a masked matrix X' by X'=$M_1 \cdot X \cdot M_2$.

Assume that n_1=n_2=n. The computational complexity for the multiplication of two dense matrices is $O(n^3)$. However, for the SMM technique, the computational complexity is reduced abruptly to $O(n^2)$ since $N_{ma\text{-}nz}^{M_1}$, $N_{ma\text{-}nz}^{M_2} \ll n$ [7]. This implies that the SMM technique can be seen as an efficient data protection technique used in an NPVC protocol.

3 Our Modified Formal Definitions

The above definitions for an NPVC protocol and its properties, introduced by Gennaro et al. [16], were intended to be used for a solution that works for *any* function represented as a circuit. In this section we modify their definitions for two reasons:

First, from Remark 1, we have shown that Setup is *not* employed by some function specific solutions. In those protocols, only KeyGen is used to generate a new key k corresponding to a function input x. In this paper, we refer this type of NPVC protocols to as *specific-purpose-oriented NPVC protocol*. The use cases [7], [14], [15], [20], [21], [22] that we are going to discuss in the later part of this paper belong to this category. Then, we introduce a new syntax to cover this category.

Second, for the function specific solution, some special natures from the construction may cause the information leakage on the function input x or output y, which affects privacy in the solution. From the inherent efficiency/privacy trade-off in a function specific solution, the privacy that captures some special natures from the construction should be more suitable for being the basic property of a function specific solution compared with the privacy in Definition 4, and needs to be precisely discussed. However, it seems that

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

some solutions are short of such explorations for the privacy. For example, to the best of our knowledge, details about how the *low density* of a sparse matrix affects privacy of a SMM-based NPVC protocol have not been formalised. The use cases [7], [14], [15], [20], [21], [22] that we are going to discuss in the later part of this paper belong to this situation. Then, we introduce a new definition of the privacy property *with respect to density* for this type of NPVC protocol.

3.1 Used Syntax

Definition 7 (Specific-Purpose-Oriented NPVC Protocol). Let f be a concrete outsourced function. A model of the specific-purpose-oriented NPVC protocol for f can be a quadruple of PPT algorithms NPVC_f=(KeyGen, $\mathsf{ProbGen}$, $\mathsf{Compute}$, Verify) defined as follows:

- $k \leftarrow \mathsf{KeyGen}(1^{\lambda}, \gamma)$. Upon input of a security parameter λ and a special parameter γ related to the input from the domain of the function f, $\mathcal{C}$ generates a new key k that can be used by the subsequent algorithms.
- $\sigma_x \leftarrow \mathsf{ProbGen}(k, x)$. Given an input x from the domain of the function f and the key k, $\mathcal{C}$ calls this algorithm to produce an encoding σ_x corresponding to the input x.
- $\sigma_y \leftarrow \mathsf{Compute}(f, \sigma_x)$. Upon input of the target function f and the encoded value σ_x, $\mathcal{W}$ runs this algorithm to obtain an encoding σ_y, where y=$f(x)$.
- $y \cup \perp \leftarrow \mathsf{Verify}(k, (\sigma_x), \sigma_y)$. Given the key k, the encoded input σ_x and the encoded output σ_y, $\mathcal{C}$ can carry out this algorithm to achieve a real output y or an error $\perp$ based on the result verification.

Specifically, according to Definition 7, in a specific NPVC protocol used for a matrix-related computation the encoding generally refers to the matrix masking. The SMM-based NPVC protocols that we are going to discuss are such examples. The details are given in Section 4.

3.2 Privacy with Respect to Density

A specific NPVC protocol used for a matrix-related computation (e.g., the SMM-based NPVC protocols [7], [14], [15], [20], [21], [22]) consists of the above four-tuple (KeyGen, $\mathsf{ProbGen}$, $\mathsf{Compute}$, Verify). The masked input from $\mathsf{ProbGen}$ and output from $\mathsf{Compute}$ generally involve the protected matrix. This implies that the analysis by $\mathcal{W}_{\mathcal{A}}$ can be based on *the density of the matrix*, i.e., $\mathcal{W}_{\mathcal{A}}$ performs the guess on the masked inputs of two distinct inputs according to the distribution of the number of zero-valued elements from the masked input/output matrix. Taking this observation into account, we formally define the privacy property with respect to density for an NPVC protocol used for a matrix-related computation. Note that the adversary is defined as $\widetilde{\mathcal{W}}_{\mathcal{A}}$ in the indistinguishability experiment. Let X=$[X(i, j)]$∈x, X'=$[X'(i, j)]$∈σ_x and Y'=$[Y'(i, j)]$∈σ_y be an $n_1 \times n_2$ input matrix, a masked input matrix from $\mathsf{ProbGen}$ and an output matrix from $\mathsf{Compute}$. Let $\mathcal{F}_{\mathcal{MC}}$ be a family of functions about the matrix-related computations.

Definition 8 (Privacy with Respect to Density Against Passive Eavesdropping). For an NPVC protocol $\mathsf{NPVC}_{f_{\mathrm{MC}}}$ used for an $f_{\mathrm{MC}} \in \mathcal{F}_{\mathcal{MC}}$, the following experiment associated with a PPT adversary $\widetilde{\mathcal{W}}_{\mathcal{A}}$ is considered:

Experiment $\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{f_{\mathrm{MC}}}, \lambda, \gamma]$:

```
[1] $(X_0, X_1) \leftarrow \widetilde{\mathcal{W}}_{\mathcal{A}}(1^{\lambda}, \gamma)$;
[2] $b \xleftarrow{\$} \{0, 1\}$;
[3] $k \leftarrow \mathsf{KeyGen}(1^{\lambda}, \gamma)$;
[4] $X'_b \leftarrow \mathsf{ProbGen}(k, X_b)$;
[5] $N_0^{X'_b} \leftarrow \widetilde{\mathcal{W}}_{\mathcal{A}}^{\mathrm{cnt}_0(X'_b)}$;
[6] $N_0^{Y'_b} \leftarrow \widetilde{\mathcal{W}}_{\mathcal{A}}^{\mathrm{cnt}_0(\mathsf{Compute}(f_{\mathrm{MC}}, X'_b))}$; (If the output of Compute
    includes $Y'$)
[7] $b' \leftarrow \widetilde{\mathcal{W}}_{\mathcal{A}}(X_0, X_1, N_0^{X'_b}, (N_0^{Y'_b}))$;
[8] If $b'$=$b$, output 1; else, output 0;
```

where the function $\mathrm{cnt}_0(\cdot)$ is used to count the number of zero-valued elements from a masked matrix. In Step [5] and [6], $\widetilde{\mathcal{W}}_{\mathcal{A}}$ runs $\mathrm{cnt}_0(X'_b)$ and $\mathrm{cnt}_0(\mathsf{Compute}(f_{\mathrm{MC}}, X'_b))$ to obtain $N_0^{X'_b}$ and $N_0^{Y'_b}$. We define the advantage of $\widetilde{\mathcal{W}}_{\mathcal{A}}$ in the experiment above as:

$$\begin{aligned}&\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{f_{\mathrm{MC}}}, \lambda, \gamma)\\ &= \left|\Pr[\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{f_{\mathrm{MC}}}, \lambda, \gamma] = 1] - \tfrac{1}{2}\right|.\end{aligned}$$

$\mathsf{NPVC}_{f_{\mathrm{MC}}}$ is $\mathsf{IND\text{-}COA}$ private with respect to density if, for $\widetilde{\mathcal{W}}_{\mathcal{A}}$, there exists a negligible function negl such that

$$\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{f_{\mathrm{MC}}}, \lambda, \gamma) \leq \mathsf{negl}(\lambda).$$

Definition 9 (Privacy with Respect to Density Against a Chosen-Plaintext Attack). For an NPVC protocol $\mathsf{NPVC}_{f_{\mathrm{MC}}}$ used for an $f_{\mathrm{MC}} \in \mathcal{F}_{\mathcal{MC}}$, the following experiment associated with a PPT adversary $\widetilde{\mathcal{W}}_{\mathcal{A}}$ is considered:

Experiment $\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}}[\mathsf{NPVC}_{f_{\mathrm{MC}}}, \lambda, \gamma]$:

```
[1] $(X_0, X_1) \leftarrow \widetilde{\mathcal{W}}_{\mathcal{A}}^{\mathrm{cnt}_0(\mathsf{PubProbGen}(\mathsf{KeyGen}(1^{\lambda}, \gamma), \cdot))}(1^{\lambda}, \gamma)$;
[2] $b \xleftarrow{\$} \{0, 1\}$;
[3] $k \leftarrow \mathsf{KeyGen}(1^{\lambda}, \gamma)$;
[4] $X'_b \leftarrow \mathsf{ProbGen}(k, X_b)$;
[5] $N_0^{X'_b} \leftarrow \widetilde{\mathcal{W}}_{\mathcal{A}}^{\mathrm{cnt}_0(X'_b)}$;
[6] $N_0^{Y'_b} \leftarrow \widetilde{\mathcal{W}}_{\mathcal{A}}^{\mathrm{cnt}_0(\mathsf{Compute}(f_{\mathrm{MC}}, X'_b))}$; (If the output of Compute
    includes $Y'$)
[7] $b' \leftarrow \widetilde{\mathcal{W}}_{\mathcal{A}}^{\mathrm{cnt}_0(\mathsf{PubProbGen}(\mathsf{KeyGen}(1^{\lambda}, \gamma), \cdot))}(X_0, X_1, N_0^{X'_b}, (N_0^{Y'_b}))$;
[8] If $b'$=$b$, output 1; else, output 0;
```

In Step [1] and [7], if the output of $\mathsf{Compute}$ does *not* include a matrix, the oracle $\mathsf{PubProbGen}(\mathsf{KeyGen}(1^{\lambda}, \gamma), X)$ simply calls $\mathsf{ProbGen}(\mathsf{KeyGen}(1^{\lambda}, \gamma), X)$ to achieve X', and $\mathsf{KeyGen}(1^{\lambda}, \gamma)$ generates a new key k for each input X; else, $\mathsf{PubProbGen}$ needs to run $\mathsf{Compute}(f_{\mathrm{MC}}, \cdot)$ to return Y'. Note that, the answer of $\mathsf{PubProbGen}$ can be probabilistic. We define the advantage of $\widetilde{\mathcal{W}}_{\mathcal{A}}$ in the experiment above as:

$$\begin{aligned}&\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}}(\mathsf{NPVC}_{f_{\mathrm{MC}}}, \lambda, \gamma)\\ &= \left|\Pr[\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}}[\mathsf{NPVC}_{f_{\mathrm{MC}}}, \lambda, \gamma] = 1] - \tfrac{1}{2}\right|.\end{aligned}$$

$\mathsf{NPVC}_{f_{\mathrm{MC}}}$ is $\mathsf{IND\text{-}CPA}$ private with respect to density if, for $\widetilde{\mathcal{W}}_{\mathcal{A}}$, there exists a negligible function negl such that

$$\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}}(\mathsf{NPVC}_{f_{\mathrm{MC}}}, \lambda, \gamma) \leq \mathsf{negl}(\lambda).$$

From Definition 8 and 9, an NPVC protocol $\mathsf{NPVC}_{f_{\mathrm{MC}}}$ that holds $\mathsf{IND\text{-}CPA}$ privacy with respect to density must hold $\mathsf{IND\text{-}COA}$ privacy with respect to density. However, if it does not hold $\mathsf{IND\text{-}COA}$ privacy with respect to density, it also does not hold $\mathsf{IND\text{-}CPA}$ privacy with respect to density.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

4 Review of SMM-Based NPVC Protocols

In this section, we review the SMM-based NPVC protocols [7], [14], [15], [20], [21], [22], where f_{LE} denotes the solution of an LE, f_{LR} denotes the LR computation, f_{MI} denotes the MI computation, f_{MM} denotes the MM computation and f_{LU} denotes the LU decomposition. The special parameter γ in KeyGen of these protocols is the matrix size. Specifically, as discussed in Subsection 1.2, the protocols [7], [21] are state-of-the-art among this type of solution. Then, they are our main target solutions, which implies that those two protocols are used for our following work. For convenience, we abbreviate those two protocols as NPVC$_{\text{MDC}}$ [21] and NPVC$_{\text{LES}}$ [7], as adopted throughout this paper. Note that we divide the SMM-based NPVC protocols into two types (see Subsection 1.2), and our review follows the two types.

Moreover, the SMM technique employed by those protocols involves *two algorithms*, i.e., KeyGen and ProbGen. Specifically, sparse matrices are produced by KeyGen, and the masking process is executed by ProbGen.

4.1 Reviewing First Type of Protocols

In [15], Chen et al. presented a protocol for outsourcing the LE solving $A\cdot\vec{x}=\vec{s}$, denoted by $\Phi_{\text{LE}}=(A, \vec{s})$, to a public $\mathcal{W}$, where A is an $n\times n$ matrix and $\vec{s}$ is an $n\times 1$ vector, as follows:

- KeyGen(1^{λ}, n): Given λ and a matrix size n, $\mathcal{C}$ generates two new $n\times n$ diagonal matrices D_1 and D_2 and a new $n\times n$ permutation matrix P as a secret key k.
- ProbGen(k, Φ_{LE}): Given k and a Φ_{LE}, $\mathcal{C}$ generates a masked LE $\Phi'_{\text{LE}}=(A', \vec{s}')$, where $\boxed{A'=D_1\cdot(P\cdot A)\cdot D_2}$ and $\vec{s}'=D_1\cdot(P\cdot\vec{s})$. Note that, there exists only one nonzero element in a row (or a column) of the product $D_1\cdot P$.
- Compute(f_{LE}, Φ'_{LE}): Upon input of f_{LE} and Φ'_{LE}, $\mathcal{W}$ outputs an $n\times 1$ solution $\vec{x}'=f_{\text{LE}}(\Phi'_{\text{LE}})$ such that $A'\cdot\vec{x}'=\vec{s}'$.
- Verify(k, Φ'_{LE}, $\vec{x}'$): Upon input of k, Φ'_{LE} and $\vec{x}'$, $\mathcal{C}$ checks whether $A'\cdot\vec{x}'=\vec{s}'$ holds. If yes, $\mathcal{C}$ computes a real $n\times 1$ solution $\vec{x}=D_2\cdot\vec{x}'$; else, $\mathcal{C}$ outputs an error $\perp$.

In [14], Chen et al. proposed a protocol for outsourcing the LR computation $\vec{y}=X\cdot\beta$, denoted by $\Phi_{\text{LR}}=(X, \vec{y})$, to a public $\mathcal{W}$, where X, $\vec{y}$ and $\beta=(X^t\cdot X)^{-1}\cdot X^t\cdot\vec{y}$ are of respective sizes $n_1\times n_2$, $n_1\times 1$ and $n_2\times 1$, where $n_1>n_2$, as follows:

- KeyGen(1^{λ}, (n_1, n_2)): Given λ and matrix sizes (n_1, n_2), $\mathcal{C}$ generates two new diagonal matrices D_1 and D_2 of respective sizes $n_1\times n_1$ and $n_2\times n_2$ as a secret key k. Specifically, D_1 satisfies $D_1^t\cdot D_1=\varphi^2\cdot I$, where φ is a randomly chosen value and I is an $n_1\times n_1$ identity matrix.
- ProbGen(k, Φ_{LR}): Given k and a Φ_{LR}, $\mathcal{C}$ generates a masked LR problem $\Phi'_{\text{LR}}=(X', \vec{y}')$, where $\boxed{X'=D_1\cdot X\cdot D_2}$ and $\vec{y}'=D_1\cdot\vec{y}$.
- Compute(f_{LR}, Φ'_{LR}): Upon input of f_{LR} and Φ'_{LR}, $\mathcal{W}$ outputs a solution $\beta'=f_{\text{LR}}(\Phi'_{\text{LR}})$ and a proof Γ.
- Verify(k, Φ'_{LR}, Γ): Upon input of k, Φ'_{LR} and Γ, $\mathcal{C}$ outputs a solution $\beta=D_2\cdot\beta'$ and checks $\Delta\vec{y}=\vec{y}-X\cdot\beta$. If $\forall i\in\{1, \ldots, n_1\}$:$\Delta\vec{y}(i, 1)$ is small enough and falls into the normal error range, $\mathcal{C}$ accepts β; else, $\mathcal{C}$ outputs an error $\perp$.

In [22], Lei et al. built a protocol for outsourcing the MI computation $Y=X^{-1}$ to a public $\mathcal{W}$, where Y and X are $n\times n$ inverse matrices, as follows:

- KeyGen(1^{λ}, n): Given λ and a matrix size n, $\mathcal{C}$ generates two new $n\times n$ randomized permutation matrices P_1 and P_2 as a secret key k.
- ProbGen(k, X): Given k and an $n\times n$ nonsingular matrix X, $\mathcal{C}$ generates an $n\times n$ masked matrix $\boxed{X'=P_1\cdot X\cdot P_2^{-1}}$.
- Compute(f_{MI}, X'): Upon input of f_{MI} and X', $\mathcal{W}$ outputs an $n\times n$ matrix $Y'=f_{\text{MI}}(X')$ and a proof Γ.
- Verify(k, Y', X, I, Γ): Upon input of k, Y', X, an $n\times n$ identity matrix I and Γ, $\mathcal{C}$ computes $Y=P_2^{-1}\cdot Y'\cdot P_1$ and outputs $n\times 1$ vector $\vec{q}_i=Y\cdot(X\cdot\vec{r}_i)-I\cdot\vec{r}_i$, where $\vec{r}_i\overset{\$}{\leftarrow}\{0, 1\}^{n\times 1}$ and $i\in\{1, \ldots, l\}$, where $l\geq 20$. If $\forall i\in\{1, \ldots, l\}$:$\vec{q}_i$ is a vector of zeros, $\mathcal{C}$ accepts $Y=X^{-1}$; else, $\mathcal{C}$ outputs an error $\perp$.

In [20], Lei et al. gave a protocol for outsourcing the MM computation $Y=X_1\cdot X_2$ to a public $\mathcal{W}$, where X_1, X_2 and Y are of respective sizes $n_1\times n_2$, $n_2\times n_3$ and $n_1\times n_3$, as follows:

- KeyGen(1^{λ}, (n_1, n_2, n_3)): Given λ and matrix sizes (n_1, n_2, n_3), $\mathcal{C}$ generates three new randomized permutation matrices P_1, P_2 and P_3 of respective sizes $n_1\times n_1$, $n_2\times n_2$ and $n_3\times n_3$ as a secret key k.
- ProbGen(k, X_1, X_2): Given k and two matrices X_1 and X_2 of respective sizes $n_1\times n_2$, $n_2\times n_3$, $\mathcal{C}$ generates two masked matrices $\boxed{X'_1=P_1\cdot X_1\cdot P_2^{-1}}$ and $\boxed{X'_2=P_2\cdot X_2\cdot P_3^{-1}}$.
- Compute(f_{MM}, X'_1, X'_2): Upon input of f_{MM}, X'_1 and X'_2, $\mathcal{W}$ outputs an $n_1\times n_3$ matrix $Y'=f_{\text{MM}}(X'_1, X'_2)$ and a proof Γ.
- Verify(k, Y', X_1, X_2, Γ): Upon input of k, Y', X_1, X_2 and Γ, $\mathcal{C}$ computes $Y=P_1^{-1}\cdot Y'\cdot P_3$ and outputs $n_1\times 1$ vector $\vec{q}_i=X_1\cdot(X_2\cdot\vec{r}_i)-Y\cdot\vec{r}_i$, where $\vec{r}_i\overset{\$}{\leftarrow}\{0, 1\}^{n_3\times 1}$ and $i\in\{1, \ldots, l\}$, where $l\geq 20$. If $\forall i\in\{1, \ldots, l\}$:$\vec{q}_i$ is a vector of zeros, $\mathcal{C}$ accepts $Y=X_1\cdot X_2$; else, $\mathcal{C}$ outputs an error $\perp$.

Now, we introduce NPVC$_{\text{MDC}}$ [21] that is used to outsource the MD computation $\det(X)$ to a public $\mathcal{W}$, where values of elements of an $n\times n$ matrix $X=[X(i, j)]$ are considered to be within the range $[-(2^{\epsilon}-1), 2^{\epsilon}-1]$, where the constant $\epsilon>0$. In particular, the *block matrix technique* and the *SMM technique* are used to provide the data protection.

- KeyGen(1^{λ}, n): Given λ and a matrix size n, $\mathcal{C}$ produces a new secret key $k=\{m, B, D, P_1, P_2\}$, where m is a randomly chosen positive integer, B is an $n\times m$ random matrix, D is an $m\times m$ diagonal matrix, P_1 and P_2 are two $(n+m)\times(n+m)$ randomized permutation matrices. Specifically, B satisfies $N_0^B\overset{\$}{\leftarrow}\{0,\ldots,n\cdot m\}$. Values of elements of these matrices are within the range $[-(2^{\mathsf{posi}(\lambda)}-1), 2^{\mathsf{posi}(\lambda)}-1]$.
- ProbGen(k, X): Given k and an $n\times n$ nonsingular matrix X, $\mathcal{C}$ performs the following:

  [1] Build an $(n+m)\times(n+m)$ enlarged matrix $\boxed{E=\begin{bmatrix} X & B \\ 0 & D \end{bmatrix}}$ using B, D and an $m\times n$ zero matrix 0.

  [2] Generate a masked matrix $\boxed{E'=P_1\cdot E\cdot P_2^{-1}}$.
- Compute(f_{LU}, E'): Given f_{LU} and E', $\mathcal{W}$ runs $f_{\text{LU}}(E')$ to obtain a lower triangular matrix L and an upper triangular matrix U of size $(n+m)\times(n+m)$ such that $E'=L\cdot U$.
- Verify(k, E', (L, U)): Upon input of k, E', L and U, $\mathcal{C}$ first computes a value $y'=\prod_{i=1}^{(n+m)}L(i, i)\cdot U(i, i)$ and then outputs a solution $y=\frac{y'\cdot\det(P_2)}{\det(P_1)\cdot\det(D)}$. Finally, $\mathcal{C}$ generates $(n+m)\times 1$ vector $\vec{q}_i=L\cdot(U\cdot\vec{r}_i)-E'\cdot\vec{r}_i$, where $\vec{r}_i\overset{\$}{\leftarrow}\{0, 1\}^{(n+m)\times 1}$ and $i\in\{1, \ldots, l\}$, where $l\geq 10$. If $\forall i\in\{1, \ldots, l\}$:$\vec{q}_i$ is a vector of zeros, $\mathcal{C}$ accepts $\det(X)=y$; else, $\mathcal{C}$ outputs an error $\perp$.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

4.2 Reviewing Second Type of Protocol

Next, we introduce $\mathsf{NPVC}_{\text{LES}}$ [7] that is used to outsource the LE solving $A{\cdot}\vec{x}{=}\vec{s}$, denoted by $\Phi_{\text{LE}}{=}(A,\ \vec{s})$, to a public $\mathcal{W}$, where values of elements of an $n{\times}n$ nonsingular matrix $A{=}[A(i,\ j)]$ and an $n{\times}1$ vector $\vec{s}{=}[s_1,\ \ldots,\ s_n]^t$ are within the range $[-(2^{\epsilon}-1),\ 2^{\epsilon}-1]$, where the constant $\epsilon{>}0$.

- $\mathsf{KeyGen}(1^{\lambda},\ n)$: Given λ and a matrix size n, $\mathcal{C}$ chooses a new secret key $k{=}\{M_1,\ M_2,\ \vec{g}\}$, where M_1 and M_2 are two $n{\times}n$ random invertible sparse matrices [4], and $\vec{g}$ is an $n{\times}1$ random blinding coefficient vector. Specifically, values of elements of M_1, M_2 and $\vec{g}$ are within the range $[-(2^{\phi}-1),\ 2^{\phi}-1]$, where $\phi{=}\mathsf{posi}(\lambda){\geq}80$.
- $\mathsf{ProbGen}(k,\ \Phi_{\text{LE}})$: Given k and a Φ_{LE}, $\mathcal{C}$ generates a masked LE $\Phi'_{\text{LE}}{=}(A',\ \vec{s}^{*})$, where $\boxed{A'{=}M_1{\cdot}A{\cdot}M_2}$ and $\vec{s}^{*}{=}M_1{\cdot}\vec{s}'$, where $\vec{s}'{=}A{\cdot}\vec{g}{+}\vec{s}$.
- $\mathsf{Compute}(f_{\text{LE}},\ \Phi'_{\text{LE}})$: Given f_{LE} and Φ'_{LE}, $\mathcal{W}$ outputs an $n{\times}1$ solution $\vec{x}'{=}f_{\text{LE}}(\Phi'_{\text{LE}})$ such that $A'{\cdot}\vec{x}'{=}\vec{s}^{*}$.
- $\mathsf{Verify}(k,\ \Phi'_{\text{LE}},\ \vec{x}')$: Upon input of k, Φ'_{LE} and $\vec{x}'$, $\mathcal{C}$ checks whether $A'{\cdot}\vec{x}'{=}\vec{s}^{*}$ holds. If yes, $\mathcal{C}$ computes a real $n{\times}1$ solution $\vec{x}{=}M_2{\cdot}\vec{x}'{-}\vec{g}$; else, $\mathcal{C}$ outputs an error $\perp$.

5 Details of Privacy Analyses

We are now ready to use Definition 8 to formally analyze the SMM-based NPVC protocols [7], [14], [15], [20], [21], [22], and to demonstrate that they are not IND-COA private with respect to density. As discussed before, $\mathsf{NPVC}_{\text{MDC}}$ and $\mathsf{NPVC}_{\text{LES}}$ are state-of-the-art. Then, our analyses mainly focus on them. However, the analysis result of $\mathsf{NPVC}_{\text{MDC}}$ is suitable for the protocols [14], [15], [20], [22] since they are the *same* type. Note that we define different sets for three types of special matrices used by our proofs: $SM^{n\times n}{\subset}\mathbb{R}^{n\times n}$, $RDM^{n\times n}{\subset}\mathbb{R}^{n\times n}$ and $DM^{n\times n}{\subset}\mathbb{R}^{n\times n}$. Specifically, a matrix X in the set $SM^{n\times n}$ is a *sparse matrix* that satisfies $\forall i{\in}\{1,\ldots,n\}$ and $\exists! j{\in}\{1,\ldots,\ n\}{:}X(i,\ j){\neq}0$. A matrix X in the set $RDM^{n\times n}$ is a *dense matrix* in which each column $X(*,\ j)$ has $\lceil\frac{n}{2}\rceil{+}\varrho$ nonzero elements, where the positive integer $\varrho{\ll}\frac{n}{2}$ and $j{\in}\{1,\ \ldots,\ n\}$, and a matrix X in the set $DM^{n\times n}$ is a *dense matrix* that satisfies $\forall i,\ j{\in}\{1,\ \ldots,\ n\}{:}X(i,\ j){\neq}0$.

5.1 Privacy Analysis on $\mathsf{NPVC}_{\text{MDC}}$

We now explore privacy of $\mathsf{NPVC}_{\text{MDC}}$ using Definition 8. It was claimed that $\mathsf{NPVC}_{\text{MDC}}$ is private (with respect to density) in [21], but our analysis shows that this protocol is not IND-COA private with respect to density. To prove Theorem 1 and 2, we first introduce two helpful lemmas. Most notably, two plaintext matrices $X_0{\in}SM^{n\times n}$ and $X_1{\in}DM^{n\times n}$ are used in the following proofs. In particular, X_0 and X_1 are of *the same* size $n{\times}n$.

Lemma 1. Let $M^{msk}{\in}SM^{n\times n}$. For $X{\in}\mathbb{R}^{n\times n}$, compute X^{pre} = $M^{msk}{\cdot}X$ and X^{post} = $X{\cdot}M^{msk}$. Then, N_0^{X} = $N_0^{X^{pre}}$ = $N_0^{X^{post}}$.

Proof: See Appendix A for the proof. □

Lemma 2. For plaintext matrices $X_0 \in SM^{n\times n}$ and $X_1 \in DM^{n\times n}$, generate a secret key k. Obtain a masked matrix $E'_0 \leftarrow \mathsf{ProbGen}(k,\ X_0)$ or $E'_1 \leftarrow \mathsf{ProbGen}(k,\ X_1)$. If E'_0 is produced, $\Pr[N_0^{E'_0}{\in}[(n^2{-}n{+}m^2{-}m{+}n{\cdot}m),\ (n^2{-}n{+}m^2{-}m{+}2{\cdot}n{\cdot}m)]] = 1$, and if E'_1 is produced, $\Pr[N_0^{E'_1}{\in}[(m^2{-}m{+}n{\cdot}m),\ (m^2{-}m{+}2{\cdot}n{\cdot}m)]] = 1$.

Proof: See Appendix B for the proof. □

Theorem 1. The protocol $\mathsf{NPVC}_{\text{MDC}}$ is not IND-COA private with respect to density if m and n satisfy $m{<}(n{-}1)$.

Proof: For $\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}[\mathsf{NPVC}_{\text{MDC}},\ \lambda,\ n]$ in Definition 8, a distinguisher $\widetilde{\mathcal{D}_{\mathcal{MDC}}}$ is used by the adversary $\widetilde{\mathcal{W}_{\mathcal{A}}}$. The challenge ciphertext is denoted as $E'_b = P_1{\cdot}E_b{\cdot}{P_2}^{-1}$, where $E_b = \begin{bmatrix} X_b & B \\ 0 & D\end{bmatrix}$. The number $N_0^{E'_b}$ is obtained. Most notably, the integer m can be revealed by $\widetilde{\mathcal{W}_{\mathcal{A}}}$ directly according to the sizes of E'_b and X_0 (or X_1), i.e., $(m{+}n)$ and n. We define $\mathrm{N}_1 := [(n^2{-}n{+}m^2{-}m{+}n{\cdot}m),\ (n^2{-}n{+}m^2{-}m{+}2{\cdot}n{\cdot}m)]$, $\mathrm{N}_2 := [(m^2{-}m{+}n{\cdot}m),\ (m^2{-}m{+}2{\cdot}n{\cdot}m)]$ and $\mathrm{N}_3 := \mathbb{N}_{+}\setminus(\mathrm{N}_1{\cup}\mathrm{N}_2)$.

Distinguisher $\widetilde{\mathcal{D}_{\mathcal{MDC}}}$:

- If $N_0^{E'_b}{\in}\mathrm{N}_1$, X_0 is used to generate E'_b. $\widetilde{\mathcal{W}_{\mathcal{A}}}$ outputs $b'{=}0$.
- If $N_0^{E'_b}{\in}\mathrm{N}_2$, X_1 is used to produce E'_b. $\widetilde{\mathcal{W}_{\mathcal{A}}}$ outputs $b'{=}1$.

If $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}(\mathsf{NPVC}_{\text{MDC}},\ \lambda,\ n){\nleq}\mathsf{negl}(\lambda)$, $\mathsf{NPVC}_{\text{MDC}}$ is not IND-COA private with respect to density. In what follows, we show that $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}(\mathsf{NPVC}_{\text{MDC}},\ \lambda,\ n){=}\frac{1}{2}$ by choosing $X_0{\in}SM^{n\times n}$, $X_1{\in}DM^{n\times n}$, and setting an appropriate relationship between m and n.

$$\begin{aligned}
&\Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}[\mathsf{NPVC}_{\text{MDC}},\lambda,n]=1]\\
&=\Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}[\mathsf{NPVC}_{\text{MDC}},\lambda,n]=1|N_0^{E'_b}\in\mathrm{N}_1]\cdot\\
&\quad\Pr[N_0^{E'_b}\in\mathrm{N}_1]+\\
&\quad\Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}[\mathsf{NPVC}_{\text{MDC}},\lambda,n]=1|N_0^{E'_b}\in\mathrm{N}_2]\cdot\\
&\quad\Pr[N_0^{E'_b}\in\mathrm{N}_2]+\\
&\quad\Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}[\mathsf{NPVC}_{\text{MDC}},\lambda,n]=1|N_0^{E'_b}\in\mathrm{N}_3]\cdot\\
&\quad\Pr[N_0^{E'_b}\in\mathrm{N}_3]
\end{aligned},$$

From Lemma 2, $\Pr[N_0^{E'_b}\in(\mathrm{N}_1{\cup}\mathrm{N}_2)] = 1$. Let $m < (n{-}1)\ \forall m \in \mathbb{N}_{+}$ and $\forall n \in \{x|x{\in}\mathbb{N}_{+},\ x{\geq}3\}$, $n{\cdot}m < (n^2{-}n)$. Then, $(m^2{-}m{+}2{\cdot}n{\cdot}m) < (n^2{-}n{+}m^2{-}m{+}n{\cdot}m)$, which implies that $\mathrm{N}_1{\cap}\mathrm{N}_2 = \emptyset$. According to Lemma 2, we obtain

$$\begin{cases}\Pr[E'_b = E'_0|N_0^{E'_b}\in\mathrm{N}_1] = 1\\ \Pr[E'_b = E'_1|N_0^{E'_b}\in\mathrm{N}_2] = 1\end{cases}.$$

Therefore, regardless if $N_0^{E'_b}\in\mathrm{N}_1$ or $N_0^{E'_b}\in\mathrm{N}_2$, $\widetilde{\mathcal{W}_{\mathcal{A}}}$ can always succeed in $\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}[\mathsf{NPVC}_{\text{MDC}},\ \lambda,\ n]$ with probability 1. This makes us obtain $\Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}[\mathsf{NPVC}_{\text{MDC}},\ \lambda,\ n] = 1] = 1$ and $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}_{\mathcal{A}}}}(\mathsf{NPVC}_{\text{MDC}},\ \lambda,\ n) = \frac{1}{2} \nleq \mathsf{negl}(\lambda)$, which confirms our theorem. □

Even though $m \geq (n{-}1)$, $\mathsf{NPVC}_{\text{MDC}}$ may also not IND-COA private with respect to density. Accordingly, we state this result with a brief proof.

Theorem 2. The protocol $\mathsf{NPVC}_{\text{MDC}}$ is not IND-COA private with respect to density if m and n satisfy $(n{-}1) \leq m < \frac{(n-1)}{\varepsilon} - \frac{1}{n}$, where $\varepsilon{\in}(0,\ \frac{1}{2}]$ is non-negligible in λ.

[4] In [7], $\forall i,j{\in}\{1,\ldots,n\}{:}N_{nz}^{M_1(i,*)}{\leq}10$ and $N_{nz}^{M_2(*,j)}{\leq}10$ when $n{\geq}5000$.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

Proof: The proof is similar to the proof for the case $m < (n{-}1)$. We construct a distinguisher $\widehat{\mathcal{D}}_{\mathcal{MDC}}$ used by $\widetilde{\mathcal{W}}_{\mathcal{A}}$ for $\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}$ [$\mathsf{NPVC}_{\text{MDC}}$, λ, n]. Note that because $m \geq (n{-}1)$, $(m^2 - m + 2 \cdot n \cdot m) \geq (n^2 - n + m^2 - m + n{\cdot}m)$, which implies that $\mathrm{N}_1 \cap \mathrm{N}_2 \neq \emptyset$. Then, we *define* $\widehat{\mathrm{N}}_1 := [(m^2{-}m{+}n{\cdot}m), (n^2{-}n{+}m^2{-}m{+}n{\cdot}m))$, $\widehat{\mathrm{N}}_2 := [(n^2{-}n{+}m^2{-}m{+}n{\cdot}m), (m^2{-}m{+}2{\cdot}n{\cdot}m)]$ and $\widehat{\mathrm{N}}_3 := ((m^2{-}m{+}2{\cdot}n{\cdot}m), (n^2{-}n{+}m^2{-}m{+}2{\cdot}n{\cdot}m)]$.

Distinguisher $\widehat{\mathcal{D}}_{\mathcal{MDC}}$:

- If $N_0^{E'_b} \in \widehat{\mathrm{N}}_1$, E'_b is generated using X_1. $\widetilde{\mathcal{W}}_{\mathcal{A}}$ outputs $b'{=}1$.
- If $N_0^{E'_b} \in \widehat{\mathrm{N}}_3$, E'_b is generated using X_0. $\widetilde{\mathcal{W}}_{\mathcal{A}}$ outputs $b'{=}0$.
- If $N_0^{E'_b} \in \widehat{\mathrm{N}}_2$, $\widetilde{\mathcal{W}}_{\mathcal{A}}$ outputs at random $b' \xleftarrow{\$} \{0, 1\}$.

Because $N_0^B \xleftarrow{\$} \{0,\ldots,n{\cdot}m\}$ for B, $N_0^{E'_b}$ subjects to the discrete uniform distribution over $\widehat{\mathrm{N}}_1 \cup \widehat{\mathrm{N}}_2$ and $\widehat{\mathrm{N}}_2 \cup \widehat{\mathrm{N}}_3$ with the probability function $\Pr(N_0^{E'_b}){=}\frac{1}{(n \cdot m+1)}$. Moreover, according to Lemma 2 and $\widehat{\mathrm{N}}_1$, $\widehat{\mathrm{N}}_2$ and $\widehat{\mathrm{N}}_3$, we have

$$\begin{cases} \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_1] = \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_1 | E'_b = E'_1] \cdot \Pr[E'_b = E'_1] \\ \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_2] = \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_2 | E'_b = E'_1] \cdot \Pr[E'_b = E'_1] \\ \qquad\qquad\qquad + \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_2 | E'_b = E'_0] \cdot \Pr[E'_b = E'_0] \\ \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_3] = \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_3 | E'_b = E'_0] \cdot \Pr[E'_b = E'_0] \end{cases},$$

$\Pr[E'_b = E'_1 | N_0^{E'_b} \in \widehat{\mathrm{N}}_1] = 1$, $\Pr[E'_b = E'_1 | N_0^{E'_b} \in \widehat{\mathrm{N}}_2] = \Pr[E'_b = E'_0 | N_0^{E'_b} \in \widehat{\mathrm{N}}_2] = \frac{1}{2}$ and $\Pr[E'_b = E'_0 | N_0^{E'_b} \in \widehat{\mathrm{N}}_3] = 1$. Now, following the above analysis, for the equation

$$\begin{aligned} &\Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}[\mathsf{NPVC}_{\text{MDC}}, \lambda, n] = 1] \\ &= \Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}[\mathsf{NPVC}_{\text{MDC}}, \lambda, n] = 1 | N_0^{E'_b} \in \widehat{\mathrm{N}}_1] \cdot \\ &\quad \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_1] + \\ &\quad \Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}[\mathsf{NPVC}_{\text{MDC}}, \lambda, n] = 1 | N_0^{E'_b} \in \widehat{\mathrm{N}}_2] \cdot \\ &\quad \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_2] + \\ &\quad \Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}[\mathsf{NPVC}_{\text{MDC}}, \lambda, n] = 1 | N_0^{E'_b} \in \widehat{\mathrm{N}}_3] \cdot \\ &\quad \Pr[N_0^{E'_b} \in \widehat{\mathrm{N}}_3] \end{aligned},$$

we have

$$\begin{aligned} &\Pr[\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}[\mathsf{NPVC}_{\text{MDC}}, \lambda, n] = 1] \\ &= (1 \cdot \tfrac{n^2-n}{n \cdot m+1} \cdot \tfrac{1}{2}) + (\tfrac{1}{2} \cdot \tfrac{n \cdot m-(n^2-n)+1}{n \cdot m+1}) + (1 \cdot \tfrac{n^2-n}{n \cdot m+1} \cdot \tfrac{1}{2}) \\ &= \tfrac{1}{2} + \tfrac{n^2-n}{2 \cdot (n \cdot m+1)} \end{aligned}.$$

Then, we have $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}$($\mathsf{NPVC}_{\text{MDC}}$, λ, n) = $\frac{n^2-n}{2 \cdot (n \cdot m+1)}$. Because $m < \frac{(n-1)}{\varepsilon} - \frac{1}{n}$, the theorem is proven based on

$$\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}_{\text{MDC}}, \lambda, n) = \frac{n-1}{2 \cdot (m+\frac{1}{n})} > \frac{\varepsilon}{2} \quad .$$

□

Remark 2. In Subsection 5.2 of [21], Lei et al. generally consider the condition $m \ll (n{-}1)$ for the experimental analyses on $\mathsf{NPVC}_{\text{MDC}}$. For example, $m \in \{100, 200, 400\}$ for $n \in \{5000, 10000, 15000, 20000\}$, and $m{=}400$ is a choice of the privacy priority. However, our analysis shows that $\mathsf{NPVC}_{\text{MDC}}$ is not IND-COA private with respect to density when $m < (n{-}1)$. We also extend our analysis to obtain a consequence that $\mathsf{NPVC}_{\text{MDC}}$ is not IND-COA private with respect to density when $(n{-}1) \leq m < \frac{(n-1)}{\varepsilon} - \frac{1}{n}$. Actually, from Figure 3 of [21], we can find that $\mathsf{NPVC}_{\text{MDC}}$ is less efficient if m increases. This implies that a large positive integer m may cause $\mathsf{NPVC}_{\text{MDC}}$ to be meaningless. Therefore, our analysis presents a practical result for $\mathsf{NPVC}_{\text{MDC}}$ using a proper size m.

Moreover, the results in Theorem 1 and 2 can be used to address the privacy issues in the protocols [14], [15], [20], [22]. Based on Definition 8, we can directly show that these protocols are not IND-COA private with respect to density.

5.2 Privacy Analysis on $\mathsf{NPVC}_{\text{LES}}$

The above subsection shows that $\mathsf{NPVC}_{\text{MDC}}$ is not IND-COA private with respect to density. The main reason lies in the drawback that $P_1 \in SM^{(n+m)\times(n+m)}$ [5] and $P_2 \in SM^{(n+m)\times(n+m)}$ are employed by ProbGen to provide the randomized permutation. Intuitively, since $\mathsf{NPVC}_{\text{LES}}$ uses new sparse matrices $M_1 \notin SM^{n\times n}$ and $M_2 \notin SM^{n\times n}$ as the secret key to mask a plaintext LE problem $\Phi_{\text{LE}}{=}(A, \vec{s})$, it seems that this protocol should be IND-CPA private (with respect to density). Next, we show that $\mathsf{NPVC}_{\text{LES}}$ actually is not IND-COA private with respect to density, which addresses the pending problem from Chen et al. [7] (see Subsection 1.3). In particular, we consider that M_1 and M_2 satisfy $\forall i, j \in \{1, 2,\ldots, n\}{:}M_1(i, j) \geq 0$ (resp. $M_1(i, j) \leq 0$), $M_2(i, j) \geq 0$ (resp. $M_2(i, j) \leq 0$). $N^{M_1}_{ma\text{-}nz}$ denotes the maximum number of $\{N_{nz}^{M_1(1,*)}, N_{nz}^{M_1(2,*)},\ldots, N_{nz}^{M_1(n,*)}\}$ and $N^{M_2}_{ma\text{-}nz}$ denotes the maximum number of $\{N_{nz}^{M_2(*,1)}, N_{nz}^{M_2(*,2)},\ldots, N_{nz}^{M_2(*,n)}\}$.

According to Definition 8, we *first* consider the case that $\widetilde{\mathcal{W}}_{\mathcal{A}}$ selects two LEs Φ^0_{LE} involving a matrix $A_0 \in SM^{n\times n}$ and Φ^1_{LE} involving a matrix $A_1 \in DM^{n\times n}$.

Lemma 3. Let the plaintext matrix involved in an LE problem $A_0 \in SM^{n\times n}$, where $\forall i, j \in \{1,2,\ldots, n\}{:}A_0(i, j) \geq 0$. Produce a secret key $k{=}\{M_1, M_2, \vec{g}\}$. Obtain a masked matrix $A'_0{=}M_1 \cdot A_0 \cdot M_2$ by running ProbGen. Then,

$$\Pr[N_0^{A'_0} \geq 1] \geq \frac{(n-N^{M_1}_{ma\text{-}nz})! \cdot (n-N^{M_2}_{ma\text{-}nz})!}{n! \cdot (n-(N^{M_2}_{ma\text{-}nz}+N^{M_1}_{ma\text{-}nz}))!} \quad .$$

Proof: See Appendix C for the proof. □

Theorem 3. Let $\Delta N_{nz} = n - 2^{(N^{M_1}_{ma\text{-}nz}-1)} \cdot (N^{M_2}_{ma\text{-}nz} + N^{M_1}_{ma\text{-}nz}-1)$. The protocol $\mathsf{NPVC}_{\text{LES}}$ is not IND-COA private with respect to density if $\frac{\Delta N_{nz}}{4 \cdot n} > 0$ is non-negligible.

Proof: According to Definition 8, for $\mathsf{Exp}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}$ [$\mathsf{NPVC}_{\text{LES}}$, λ, n], $\widetilde{\mathcal{W}}_{\mathcal{A}}$ chooses two inputs Φ^0_{LE} and Φ^1_{LE}, in which matrices $A_0 \in SM^{n\times n}$ and $A_1 \in DM^{n\times n}$, where $A_0(i, j) \geq 0$ and $A_1(i, j) > 0$ $\forall i, j \in \{1, 2,\ldots, n\}$. Then, we construct a distinguisher $\widetilde{\mathcal{D}}_{\mathcal{LES}}$ employed by $\widetilde{\mathcal{W}}_{\mathcal{A}}$. Let the challenge ciphertext $A'_b{=}M_1 \cdot A_b \cdot M_2$, which is from a masked LE $\Phi'_{\text{LE}}{=}(A'_b, \vec{s}^{\,*}_b)$. The number $N_0^{A'_b}$ is obtained.

Distinguisher $\widetilde{\mathcal{D}}_{\mathcal{LES}}$:

- If $N_0^{A'_b} \geq 1$, A_0 is employed to obtain A'_b. $\widetilde{\mathcal{W}}_{\mathcal{A}}$ outputs $b'{=}0$.
- If $N_0^{A'_b}{=}0$, $\widetilde{\mathcal{W}}_{\mathcal{A}}$ outputs at random $b' \xleftarrow{\$} \{0, 1\}$.

In the following, the result $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}$ ($\mathsf{NPVC}_{\text{LES}}$, λ, n) $\not\leq$ $\mathsf{negl}(\lambda)$ can be shown when values of n, $N^{M_1}_{ma\text{-}nz}$ and

[5] The definition of $SM^{(n+m)\times(n+m)}$ is the same as the definition of $SM^{n\times n}$ except $(n + m) > n$.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

$N^{M_2}_{\textit{ma-nz}}$ satisfy the condition that $\frac{\Delta N_{nz}}{4\cdot n}{>}0$ is non-negligible. This illustrates that $\mathsf{NPVC}_{\mathrm{LES}}$ is actually not $\mathsf{IND\text{-}COA}$ private with respect to density.

$$\begin{aligned}&\Pr[\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n] = 1]\\ &= \Pr[\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n] = 1 | N_0^{A'_b} = 0]\cdot\\ &\quad\Pr[N_0^{A'_b} = 0]+\\ &\quad\Pr[\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n] = 1 | N_0^{A'_b} \geq 1]\cdot\\ &\quad\Pr[N_0^{A'_b} \geq 1]\end{aligned}\ .$$

Since $A_1 \in DM^{n\times n}$, we have $\Pr[N_0^{A'_1}{=}0] = 1$. If $A_b = A_0 \in SM^{n\times n}$, we obtain $\Pr[\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n] = 1 \mid N_0^{A'_b}{\geq}1] = 1$. When $N_0^{A'_b} = 0$, $\widetilde{\mathcal{D}}_{\mathcal{LES}}$ has to make a random guess, i.e., $\Pr[\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n] = 1 \mid N_0^{A'_b} = 0] = \frac{1}{2}$. According to $\Pr[N_0^{A'_b}{=}0] + \Pr[N_0^{A'_b}{\geq}1] = 1$, we have

$$\begin{aligned}&\Pr[\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n] = 1]\\ &= \tfrac{1}{2}\cdot(1-\Pr[N_0^{A'_b} \geq 1]) + 1\cdot\Pr[N_0^{A'_b} \geq 1]\\ &= \tfrac{1}{2} + \tfrac{1}{2}\cdot\Pr[N_0^{A'_b} \geq 1]\\ &= \tfrac{1}{2} + \tfrac{1}{2}\cdot(\Pr[N_0^{A'_0} \geq 1]\cdot\tfrac{1}{2} + \Pr[N_0^{A'_1} \geq 1]\cdot\tfrac{1}{2})\\ &= \tfrac{1}{2} + \tfrac{1}{4}\cdot\Pr[N_0^{A'_0} \geq 1]\end{aligned}\ .$$

Then, based on Definition 8 and Lemma 3, we obtain that

$$\begin{aligned}&\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n)\\ &= \tfrac{1}{4}\cdot\Pr[N_0^{A'_0} \geq 1] \geq \tfrac{(n-N^{M_1}_{\textit{ma-nz}})!\cdot(n-N^{M_2}_{\textit{ma-nz}})!}{4\cdot n!\cdot(n-(N^{M_2}_{\textit{ma-nz}}+N^{M_1}_{\textit{ma-nz}}))!}\end{aligned}\ . \tag{1}$$

Since $\frac{(n-N^{M_1}_{\textit{ma-nz}})!\cdot(n-N^{M_2}_{\textit{ma-nz}})!}{n!\cdot(n-(N^{M_2}_{\textit{ma-nz}}+N^{M_1}_{\textit{ma-nz}}))!}{>}(1{-}\frac{N^{M_2}_{\textit{ma-nz}}+N^{M_1}_{\textit{ma-nz}}-1}{n})^{N^{M_1}_{\textit{ma-nz}}}$, according to the binomial theorem,

$$\begin{aligned}&\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n)\\ &> \tfrac{1}{4}\cdot\textstyle\sum_{\rho=0}^{N^{M_1}_{\textit{ma-nz}}}\binom{N^{M_1}_{\textit{ma-nz}}}{\rho}\cdot(-\frac{N^{M_2}_{\textit{ma-nz}}+N^{M_1}_{\textit{ma-nz}}-1}{n})^{\rho}\end{aligned}\ .$$

We know that if $\rho{:=}2{\cdot}z{+}1$ ($z{\in}\mathbb{N}_{+}$), the sum of binomial coefficients $\sum_{\rho=1}^{o_{ma}} = 2^{(N^{M_1}_{\textit{ma-nz}}-1)}$, where o_{ma} is the largest odd number in $\mathbb{Z}_{(N^{M_1}_{\textit{ma-nz}}+1)}$. Therefore,

$$\begin{aligned}&\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n)\\ &> \tfrac{1}{4}\cdot(\binom{N^{M_1}_{\textit{ma-nz}}}{0} - 2^{N^{M_1}_{\textit{ma-nz}}-1}\cdot\tfrac{N^{M_2}_{\textit{ma-nz}}+N^{M_1}_{\textit{ma-nz}}-1}{n})\\ &\quad= \tfrac{n-2^{(N^{M_1}_{\textit{ma-nz}}-1)}\cdot(N^{M_2}_{\textit{ma-nz}}+N^{M_1}_{\textit{ma-nz}}-1)}{4\cdot n}\end{aligned}\ .$$

Since $\frac{n-2^{(N^{M_1}_{\textit{ma-nz}}-1)}\cdot(N^{M_2}_{\textit{ma-nz}}+N^{M_1}_{\textit{ma-nz}}-1)}{4\cdot n}{>}0$ is non-negligible, we demonstrate the theorem. □

Remark 3. When $N^{M_1}_{\textit{ma-nz}}$ and $N^{M_2}_{\textit{ma-nz}}$ are two chosen small positive integers, $f(n) = \frac{n-2^{(N^{M_1}_{\textit{ma-nz}}-1)}\cdot(N^{M_2}_{\textit{ma-nz}}+N^{M_1}_{\textit{ma-nz}}-1)}{4\cdot n} > 0$ must be non-negligible for the large enough size n. Specifically, the function $f(n)$ can be seen as the *lower bound* related to the adversary's advantage to prove Theorem 3. Assume that the size of a matrix n = 5000 (resp. 10000), and $N^{M_1}_{\textit{ma-nz}}$, $N^{M_2}_{\textit{ma-nz}}{\in}\{1, 2,\ldots, 9\}$. According to Theorem 3, $\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n) > \frac{81}{2500}$ (resp. $\frac{353}{2500}$), which is non-negligible. However, this does not imply that $\widetilde{\mathcal{W}}_{\mathcal{A}}$ cannot win with a non-negligible advantage when $N^{M_1}_{\textit{ma-nz}}{\geq}10$ and $N^{M_2}_{\textit{ma-nz}}{\geq}10$. From Inequality 1, when n=5000 and $N^{M_1}_{\textit{ma-nz}}{=}N^{M_2}_{\textit{ma-nz}}{=}10$, $\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n){\geq}\frac{667}{2722}$, which is non-negligible. For Theorem 3, we want to employ the accurate lower bound, which is appropriate for *extremely* sparse matrices M_1 and M_2 that are used by the experimental evaluation of [7], to show that $\mathsf{NPVC}_{\mathrm{LES}}$ is not $\mathsf{IND\text{-}COA}$ private with respect to density.

Furthermore, we extend our privacy analysis for $\mathsf{NPVC}_{\mathrm{LES}}$ and consider the case that $\widetilde{\mathcal{W}}_{\mathcal{A}}$ chooses two LEs Φ^0_{LE} including a dense matrix $A_0{\in}RDM^{n\times n}$ and Φ^1_{LE} including a dense matrix $A_1{\in}DM^{n\times n}$.

Lemma 4. Let the plaintext matrix of an LE problem $A_0{\in}RDM^{n\times n}$, where $\forall i, j{\in}\{1{,}2{,}\ldots, n\}{:}A_0(i, j){\geq}0$. Generate a secret key $k{=}\{M_1, M_2, \vec{g}\}$. Obtain a masked matrix $A'_0{=}M_1{\cdot}A_0{\cdot}M_2$ by running $\mathsf{ProbGen}$. Then,

$$\Pr[N_0^{A'_0} \geq 1] \geq (\tfrac{(\frac{n}{2}-\varrho)!\cdot(n-N^{M_1}_{\textit{ma-nz}})!}{n!\cdot(\frac{n}{2}-\varrho-N^{M_1}_{\textit{ma-nz}})!})^{N^{M_2}_{\textit{ma-nz}}}\ .$$

Proof: See Appendix D for the proof. □

Theorem 4. Let $\widehat{\Delta N_{nz}}{=}\frac{2\cdot(\varrho+N^{M_1}_{\textit{ma-nz}}-1)}{n}$. The protocol $\mathsf{NPVC}_{\mathrm{LES}}$ is not $\mathsf{IND\text{-}COA}$ private with respect to density if $\frac{1}{4}{\cdot}(\frac{1-\widehat{\Delta N_{nz}}}{2})^{N^{M_1}_{\textit{ma-nz}}\cdot N^{M_2}_{\textit{ma-nz}}}{>}0$ is non-negligible.

Proof: This proof is similar to that of Theorem 3. This implies that we also want to show the result $\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n) \nleq \mathsf{negl}(\lambda)$. In particular, according to Definition 8, for $\mathsf{Exp}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}[\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n]$, $\widetilde{\mathcal{W}}_{\mathcal{A}}$ selects two inputs Φ^0_{LE} and Φ^1_{LE} involving respective matrices $A_0 \in RDM^{n\times n}$ and $A_1 \in DM^{n\times n}$, where $A_0(i, j) \geq 0$ and $A_1(i, j) > 0\ \forall i, j \in \{1, 2,\ldots, n\}$. Note that $\widetilde{\mathcal{W}}_{\mathcal{A}}$ *can* choose ϱ. Moreover, the distinguisher $\widetilde{\mathcal{D}}_{\mathcal{LES}}$ is employed by $\widetilde{\mathcal{W}}_{\mathcal{A}}$.

According to Inequality 1, we know that

$$\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n) = \tfrac{1}{4}\cdot\Pr[N_0^{A'_0} \geq 1]\ \cdot$$

From Lemma 4, this advantage satisfies

$$\begin{aligned}&\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n)\\ &\geq \tfrac{1}{4}\cdot(\tfrac{(\frac{n}{2}-\varrho)!\cdot(n-N^{M_1}_{\textit{ma-nz}})!}{n!\cdot(\frac{n}{2}-\varrho-N^{M_1}_{\textit{ma-nz}})!})^{N^{M_2}_{\textit{ma-nz}}}\end{aligned}\ .$$

Since

$$\begin{aligned}&(\tfrac{(\frac{n}{2}-\varrho)!\cdot(n-N^{M_1}_{\textit{ma-nz}})!}{n!\cdot(\frac{n}{2}-\varrho-N^{M_1}_{\textit{ma-nz}})!})^{N^{M_2}_{\textit{ma-nz}}}\\ &> (\tfrac{\frac{n}{2}-\varrho}{n}\cdot\tfrac{\frac{n}{2}-(\varrho+1)}{n}\cdot\tfrac{\frac{n}{2}-(\varrho+2)}{n}\cdot\ldots\cdot\tfrac{\frac{n}{2}-(\varrho+N^{M_1}_{\textit{ma-nz}}-1)}{n})^{N^{M_2}_{\textit{ma-nz}}}\\ &> (\tfrac{1}{2} - \tfrac{\varrho+N^{M_1}_{\textit{ma-nz}}-1}{n})^{N^{M_1}_{\textit{ma-nz}}\cdot N^{M_2}_{\textit{ma-nz}}}\end{aligned}\ ,$$

we have

$$\begin{aligned}&\mathsf{Adv}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}^{\text{priv-dens}^{\text{coa}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n)\\ &> \tfrac{1}{4}\cdot(\tfrac{1}{2} - \tfrac{\varrho+N^{M_1}_{\textit{ma-nz}}-1}{n})^{N^{M_1}_{\textit{ma-nz}}\cdot N^{M_2}_{\textit{ma-nz}}}\end{aligned}\ .$$

Then, according to the condition that $\frac{1}{4}\cdot(\frac{1}{2} - \frac{\varrho+N^{M_1}_{\textit{ma-nz}}-1}{n})^{N^{M_1}_{\textit{ma-nz}}\cdot N^{M_2}_{\textit{ma-nz}}} > 0$ is non-negligible, we confirm our result. □

Remark 4. The condition that $\frac{1}{4}{\cdot}(\frac{1}{2}{-}\frac{\varrho+N^{M_1}_{\textit{ma-nz}}-1}{n})^{N^{M_1}_{\textit{ma-nz}}\cdot N^{M_2}_{\textit{ma-nz}}}{>}0$ is non-negligible is reasonable for the proof of Theorem

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

TABLE 1
Simulation Results on $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}_{\mathrm{MDC}}, \lambda, n)$

n	300	400	500	600	1000
Adv ($m<(n-1)$)	0.500	0.500	0.500	0.500	0.500
Adv ($m\geq(n-1)$)	0.365	0.375	0.365	0.390	0.395

TABLE 2
Simulation Results on $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n)$

n	$\mathsf{NPVC}_{\mathrm{LES}}(A_0 \in SM^{n\times n})$			$\mathsf{NPVC}_{\mathrm{LES}}(A_0 \in RDM^{n\times n}, \varrho=10)$		
	$N_{ma\text{-}nz}$	Adv		$N_{ma\text{-}nz}$	Adv	
		Exp.	$\geq$Theor.		Exp.	$\geq$Theor.
300	5	0.290	0.230	5	0.065	$1.094\cdot10^{-9}$
	10	0.270	0.177	10	0.030	$3.376\cdot10^{-35}$
	50	0.215	$1.083\cdot10^{-5}$	50	0.015	$2.791\cdot10^{-952}$
	70	0.020	$1.043\cdot10^{-10}$	70	0.010	$1.680\cdot10^{-1999}$
400	6	0.290	0.228	5	0.070	$1.797\cdot10^{-9}$
	10	0.290	0.194	10	0.040	$3.263\cdot10^{-34}$
	60	0.245	$6.057\cdot10^{-6}$	50	0.030	$3.268\cdot10^{-894}$
	75	0.080	$6.683\cdot10^{-9}$	70	0.015	$5.558\cdot10^{-1835}$
500	6	0.275	0.232	5	0.035	$2.407\cdot10^{-9}$
	10	0.260	0.204	10	0.025	$1.231\cdot10^{-33}$
	60	0.250	$6.880\cdot10^{-5}$	50	0.035	$1.570\cdot10^{-862}$
	80	0.180	$5.592\cdot10^{-8}$	70	0.030	$9.992\cdot10^{-1750}$
600	6	0.290	0.235	5	0.065	$2.918\cdot10^{-9}$
	10	0.260	0.211	10	0.045	$2.942\cdot10^{-33}$
	60	0.220	$3.158\cdot10^{-4}$	50	0.030	$1.511\cdot10^{-842}$
	85	0.210	$1.920\cdot10^{-7}$	70	0.015	$2.299\cdot10^{-1697}$
1000	6	0.275	0.241	5	0.075	$4.267\cdot10^{-9}$
	10	0.270	0.226	10	0.050	$1.630\cdot10^{-32}$
	60	0.260	$5.425\cdot10^{-3}$	50	0.020	$5.316\cdot10^{-805}$
	90	0.255	$3.373\cdot10^{-5}$	70	0.015	$1.185\cdot10^{-1601}$

4. Specifically, for the parameters $\varrho\ll\frac{n}{2}$, $N^{M_1}_{ma\text{-}nz}\ll n$ and $N^{M_2}_{ma\text{-}nz}\ll n$, they are generally assumed to be some small positive integers, e.g., ϱ=20 and $N^{M_1}_{ma\text{-}nz}=N^{M_2}_{ma\text{-}nz}$=10 for $n\geq5000$. Then, if ϱ, $N^{M_1}_{ma\text{-}nz}$ and $N^{M_2}_{ma\text{-}nz}$ are some chosen integers, $f(n)=\frac{1}{4}\cdot(\frac{1}{2}-\frac{\varrho+N^{M_1}_{ma\text{-}nz}-1}{n})^{N^{M_1}_{ma\text{-}nz}\cdot N^{M_2}_{ma\text{-}nz}}$ must be non-negligible for the large enough size n.

5.3 Computer Simulations of Theoretical Analyses

To confirm the above analysis results on $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$, we implemented *real* example experiments by computing the advantage that the adversary $\widetilde{\mathcal{W}}_{\mathcal{A}}$ answers correctly. In the experiments, $\widetilde{\mathcal{W}}_{\mathcal{A}}$ chooses a number of plaintext matrix pairs (X_0, X_1) and plaintext LE pairs (Φ^0_{LE}, Φ^1_{LE}) satisfying X_0, $A_0\in SM^{n\times n}$ (or $A_0\in RDM^{n\times n}(\varrho=10)$) and X_1, $A_1\in DM^{n\times n}$, where $A_0(i,j)\geq0$ and $A_1(i,j)>0$ $\forall i, j\in\{1, 2,\ldots, n\}$. Specifically, for each size $n\in\{300, 400, 500, 600, 1000\}$, the new secret keys k_{MDC} and k_{LES} are used to mask each plaintext matrix X_b and plaintext LE problem $\Phi^b_{\mathrm{LE}}=(A_b, \vec{s}_b)$ over 200 experiments, respectively. Note that m belonging to k_{MDC} satisfies $m\xleftarrow{\$}\{101, 102,\ldots, (n-2)\}$ for each experiment of the case $m<(n-1)$, and $m\xleftarrow{\$}\{(n-1), n, \ldots, (2\cdot n-3)\}$ for each experiment of the case $m\geq(n-1)$. All simulation experiments were executed in MATLAB R2015b running on a 2.20GHz Core(TM) i7-6650U CPU with 16.00GB of RAM. Table 1 and 2 list the experimental results.

In Table 1, Adv denotes $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}_{\mathrm{MDC}}, \lambda, n)$. In Table 2, Adv denotes $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n)$, and $N_{ma\text{-}nz}$ is the $N^{M_1}_{ma\text{-}nz}$ (and $N^{M_2}_{ma\text{-}nz}$), which implies that we choose the case $N^{M_1}_{ma\text{-}nz}=N^{M_2}_{ma\text{-}nz}$. From the experimental results of Table 1, for $\mathsf{NPVC}_{\mathrm{MDC}}$, $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}_{\mathrm{MDC}}, \lambda, n)$ matches the theoretical advantages of Theorem 1 (i.e., Adv=0.5) and 2 (i.e., Adv≈0.347), which confirms that $\mathsf{NPVC}_{\mathrm{MDC}}$ is not IND-COA private with respect to density. Moreover, from the experimental results about $\mathsf{NPVC}_{\mathrm{LES}}$ in Table 2, when $N_{ma\text{-}nz}\ll n$ (e.g., $N_{ma\text{-}nz}\in\{5, 6, 10\}$), regardless if $A_0\in SM^{n\times n}$ or $A_0\in RDM^{n\times n}$, $\mathsf{Adv}^{\text{priv-dens}^{\text{coa}}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}_{\mathrm{LES}}, \lambda, n)$ is absolutely non-negligible, which is in accord with the deduction of Theorem 3 and 4 and shows that $\mathsf{NPVC}_{\mathrm{LES}}$ is not IND-COA private with respect to density if M_1 and M_2 are extremely sparse matrices.

6 MODIFICATION FOR $\mathsf{NPVC}_{\mathrm{MDC}}$ AND $\mathsf{NPVC}_{\mathrm{LES}}$

Since $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$ are not IND-COA private with respect to density, we attempt to propose a construction for making them hold privacy with respect to density and keeping *all* other properties (e.g., correctness and security). Then, this construction *simply* focuses on modifying the algorithms KeyGen and ProbGen that are directly related to privacy. Specifically, we still follow the direction of the SMM technique to develop a construction that can make the targeted NPVC protocol IND-CPA private with respect to density. This implies that a $\mathcal{C}$ only generates (extremely) *sparse matrices* to mask the input with a computational complexity $O(n^2)$. We believe that the *efficient* solution holding IND-CPA privacy with respect to density is already interesting. The details of the construction are described below.

6.1 Special Primitives

Two primitives are considered for our SMM construction:

6.1.1 χ-Sparse Matrix Pair for Masking

First, we introduce a χ-sparse matrix pair (M^{χ}_{pre}, M^{χ}_{post}) holding the property that *at most* three nonzero elements in a row (resp. a column) of M^{χ}_{pre} (resp. M^{χ}_{post}). This matrix pair is produced by KeyGen to mask a nonsingular matrix.

Definition 10 (χ-Sparse Matrix Pair (M^{χ}_{pre}, M^{χ}_{post})). For an $n\times n$ sparse matrix M, if it holds:

[1] $\forall i\in\{1, 2,\ldots, n\}$ and $\exists j', j''\in\{1, 2,\ldots, n\}$ $(j'\neq j'')$:$M(i, j')\neq0$ and $M(i, j'')\neq0$.

[2] When i=j or i=$n-(j-1)$, $M(i, j)\neq0$ $\forall i, j\in\{1, 2,\ldots, n\}$.

We define M as a *pre*-multiplication χ-sparse matrix, denoted M^{χ}_{pre}. If it satisfies:

[1] $\forall j\in\{1, 2,\ldots, n\}$ and $\exists i', i''\in\{1, 2,\ldots, n\}$ $(i'\neq i'')$:$M(i', j)\neq0$ and $M(i'', j)\neq0$.

[2] When i=j or i=$n-(j-1)$, $M(i, j)\neq0$ $\forall i, j\in\{1, 2,\ldots, n\}$.

We define M as a *post*-multiplication χ-sparse matrix, denoted M^{χ}_{post}. A pair (M^{χ}_{pre}, M^{χ}_{post}) of size $n\times n$ is defined as a χ-sparse matrix pair when j'=i' and j''=i'' for $\{M^{\chi}_{pre}(*, j'), M^{\chi}_{pre}(*, j'')\}$ and $\{M^{\chi}_{post}(i', *), M^{\chi}_{post}(i'', *)\}$.

If i=j, $M^{\chi}_{pre}(i, j)\neq0$ and $M^{\chi}_{post}(i, j)\neq0$ $\forall i, j\in\{1, 2,\ldots, n\}$, we call M^{χ}_{pre} and M^{χ}_{post} the Type-I χ-sparse matrices. If i=$n-(j-1)$, $M^{\chi}_{pre}(i, j)\neq0$ and $M^{\chi}_{post}(i, j)\neq0$ $\forall i, j\in\{1, 2,\ldots, n\}$, we call M^{χ}_{pre} and M^{χ}_{post} the Type-II χ-sparse matrices. From Definition 10, the product of M^{χ}_{pre} and M^{χ}_{post} can be a *dense*

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

matrix. This is the *basic merit* of the χ-sparse matrix pair (M^{χ}_{pre}, M^{χ}_{post}). For showing the proposed χ-sparse matrices, some toy examples of size 6×6 are given in Appendix H.

Lemma 5. For a pair (M^{χ}_{pre}, M^{χ}_{post}) of size $n\times n$, assume that $\forall i, j\in\{1,\ldots, n\}$ and $\exists z, z'\in\{1, 2,\ldots, n\}$ (w.l.o.g., $z<z'$) : $M^{\chi}_{pre}(i, z)\neq 0$, $M^{\chi}_{pre}(i, z')\neq 0$, $M^{\chi}_{post}(z, j)\neq 0$ and $M^{\chi}_{post}(z', j)\neq 0$. Then, $\det(M^{\chi}_{pre})$ and $\det(M^{\chi}_{post})$ are given by Equation 6. Specifically, if z=1, in Equation 6a, $\prod_{i=1}^{(z-1)} M^{\chi}_{pre}(i, i)$ and $\prod_{i=(n-z+2)}^{n} M^{\chi}_{pre}(i, (n-i+1))$ are ignored. In Equation 6b, $\prod_{i=1}^{(z-1)} M^{\chi}_{post}(i, i)$ and $\prod_{i=1}^{(z-1)} M^{\chi}_{post}(i, (n-i+1))$ are ignored. If z'=n, in Equation 6a, $\prod_{i=(z'+1)}^{n} M^{\chi}_{pre}(i, i)$ and $\prod_{i=1}^{(n-z')} M^{\chi}_{pre}(i, (n-i+1))$ are ignored. In Equation 6b, $\prod_{i=(z'+1)}^{n} M^{\chi}_{post}(i, i)$ and $\prod_{i=(z'+1)}^{n} M^{\chi}_{post}(i, (n-i+1))$ are ignored.

Proof: See Appendix E for the proof. □

In Appendix I, we use some simple toy examples to demonstrate Lemma 5.

Lemma 6. Assume that $\forall i, j\in\{1, 2,\ldots, n\}$ and $\exists z, z'\in\{1, 2,\ldots, n\}$ (w.l.o.g., $z<z'$) : $M^{\chi}_{pre}(i, z)\neq 0$, $M^{\chi}_{pre}(i, z')\neq 0$, $M^{\chi}_{post}(z, j)\neq 0$ and $M^{\chi}_{post}(z', j)\neq 0$. Then, M^{χ}_{pre} and M^{χ}_{post} of size $n\times n$ are invertible matrices iff the following conditions hold:

[1] When M^{χ}_{pre} is a Type-I χ-sparse matrix, $M^{\chi}_{pre}(z, z)\cdot M^{\chi}_{pre}(z', z')\neq M^{\chi}_{pre}(z, z')\cdot M^{\chi}_{pre}(z', z)$.

[2] When M^{χ}_{pre} is a Type-II χ-sparse matrix, $M^{\chi}_{pre}((n-z'+1), z')\cdot M^{\chi}_{pre}((n-z+1), z)\neq M^{\chi}_{pre}((n-z'+1), z)\cdot M^{\chi}_{pre}((n-z+1), z')$.

[3] When M^{χ}_{post} is a Type-I χ-sparse matrix, $M^{\chi}_{post}(z, z)\cdot M^{\chi}_{post}(z', z') \neq M^{\chi}_{post}(z', z)\cdot M^{\chi}_{post}(z, z')$.

[4] When M^{χ}_{post} is a Type-II χ-sparse matrix, $M^{\chi}_{post}(z, (n-z+1)) \cdot M^{\chi}_{post}(z', (n-z'+1)) \neq M^{\chi}_{post}(z, (n-z'+1)) \cdot M^{\chi}_{post}(z', (n-z+1))$.

Proof: See Appendix F for the proof. □

To demonstrate Lemma 6, we also give some simple examples in Appendix J.

6.1.2 Adaptive Adjustment Function

In general, for the matrix operations X'=$M^{\chi}_{pre}\cdot X\cdot M^{\chi}_{post}$, where X is an $n\times n$ nonsingular matrix, the pre-multiplication X^{*}=$M^{\chi}_{pre}\cdot X$ can be expressed as X^{*}=$M^{\chi}_{pre}\cdot\prod_{i=1}^{N_{P_i}} P_i$, where P_i is an elementary matrix and N_{P_i} is the number of P_i. This implies that M^{χ}_{pre} performs finite elementary column operations to obtain X^{*}, and the columns $M^{\chi}_{pre}(*, j')$ and $M^{\chi}_{pre}(*, j'')$, where $M^{\chi}_{pre}(i, j')\neq 0$ and $M^{\chi}_{pre}(i, j'')\neq 0$ $\forall i\in\{1, 2,\ldots, n\}$, may be switched with two other columns. This may make the post-multiplication X'=$X^{*}\cdot M^{\chi}_{post}$ leak some information with respect to density. Then, an adaptive adjustment function adj(·, ·) is defined to retain the basic merit of the pair (M^{χ}_{pre}, M^{χ}_{post}) used for masking X. The purpose of the function adj(·, ·) is to adjust *the row indices* of rows $M^{\chi}_{post}(i', *)$ and $M^{\chi}_{post}(i'', *)$, where $M^{\chi}_{post}(i', j)\neq 0$ and $M^{\chi}_{post}(i'', j)\neq 0$ $\forall j\in\{1, 2,\ldots, n\}$, according to indices of nonzero elements in X so that X' cannot leak some information with respect to density.

Definition 11 (Adaptive Adjustment Function). The adaptive adjustment function, denoted $M^{\chi}_{adj\text{-}post}$=adj($X$, M^{χ}_{post}), where X is an $n\times n$ nonsingular plaintext matrix, is specified by the following steps:

[1] Assume that $M^{\chi}_{post}(i', j)\neq 0$ and $M^{\chi}_{post}(i'', j)\neq 0$ $\forall j\in\{1, 2,\ldots, n\}$, where $i', i''\in\{1, 2,\ldots, n\}$. Then, the first (or second) nonzero element in respective rows i' and i'' of X, e.g., $X(i', j_{i'})$ and $X(i'', j_{i''})$, can be determined. Note that $j_{i'}\neq j_{i''}$. Record the column indices $j_{i'}$ and $j_{i''}$.

[2] According to the indices $j_{i'}$ and $j_{i''}$, output the matrix $M^{\chi}_{adj\text{-}post}$ after running the following switching:

a) $M^{\chi}_{post}(i', *) \leftrightarrow M^{\chi}_{post}(j_{i'}, *)$, $M^{\chi}_{post}(i'', *) \leftrightarrow M^{\chi}_{post}(j_{i''}, *)$.

b) $M^{\chi}_{post}(*, j_{i'}) \leftrightarrow M^{\chi}_{post}(*, i')$ (or $M^{\chi}_{post}(*, (n-j_{i'}+1)) \leftrightarrow M^{\chi}_{post}(*, (n-i'+1))$), $M^{\chi}_{post}(*, j_{i''}) \leftrightarrow M^{\chi}_{post}(*, i'')$ (or $M^{\chi}_{post}(*, (n-j_{i''}+1)) \leftrightarrow M^{\chi}_{post}(*, (n-i''+1))$).

Remark 5. If there exist $N_{nz}^{X(i',*)}\geq 2$ and $N_{nz}^{X(i'',*)}\geq 2$ nonzero elements in the respective rows i' and i'' of X, the elements $X(i', j_{i'})$ and $X(i'', j_{i''})$ can be chosen uniformly at random from the $N_{nz}^{X(i',*)}$ entries and the $N_{nz}^{X(i'',*)}$ entries to extract the column indices $j_{i'}$ and $j_{i''}$, respectively. This makes adj(·, ·) randomly produce $M^{\chi}_{adj\text{-}post}$.

6.2 Details of Our SMM Construction

We now present a SMM construction used to modify $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$. Specifically, since the proposed SMM construction simply involves two algorithms KeyGen and ProbGen of the NPVC protocols, we give a high-level description of these two algorithms, which are used to be substituted for the corresponding ingredients in those two algorithms of $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$.

Definition 12 (Proposed SMM Construction). The proposed *SMM* construction involving a two-tuple of PPT algorithms (KeyGen, ProbGen) is used to hide a nonsingular matrix. We define this SMM construction as follows:

- KeyGen(1^{λ}, n): Given λ and a matrix size n, specify two matrix sets $\chi_{pre}(\lambda)$ and $\chi_{post}(\lambda)$, where $\chi_{pre}(\lambda)$ includes $n\times n$ Type-I and Type-II invertible pre-multiplication χ-sparse matrices, and $\chi_{post}(\lambda)$ includes the corresponding $n\times n$ Type-I and Type-II invertible post-multiplication χ-sparse matrices. For each matrix in $\chi_{pre}(\lambda)$ and $\chi_{post}(\lambda)$, values of nonzero elements are chosen from the range $[-(2^{\eta}-1), 0)\bigcup(0, 2^{\eta}-1]$ uniformly at random, where η=posi(λ)$\geq\lambda$. Then, generate a fresh χ-sparse matrix pair (M^{χ}_{pre}, M^{χ}_{post}) of size $n\times n$ by (M^{χ}_{pre}, M^{χ}_{post})$\xleftarrow{\$}$($\chi_{pre}(\lambda)$, $\chi_{post}(\lambda)$), and output this pair as a part of a secret key k.
- ProbGen(k, X): Upon input of k and an $n\times n$ nonsingular matrix X from the outsourced data, where $\forall i, j\in\{1, 2, \ldots, n\}$:$X(i, j)\in[-(2^{\epsilon}-1), 2^{\epsilon}-1]$, where $\epsilon>0$, $\mathcal{C}$ performs the following: run $\boxed{M^{\chi}_{adj\text{-}post} \leftarrow \mathrm{adj}(X, M^{\chi}_{post})}$ first. Then, output $\boxed{X'=M^{\chi}_{pre}\cdot X\cdot M^{\chi}_{adj\text{-}post}}$ as a masked matrix of X.

To clearly show our SMM construction, a simple toy example is given in Appendix K to guide the readers.

6.3 Descriptions of Modified Protocols

Definition 12 gives a construction on how to mask a nonsingular matrix by employing sparse matrices. In what follows, the *key points* of the proposed SMM construction-based modifications for $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$ are described. *Most notably*, for the modified protocols $\mathsf{NPVC}^{\chi}_{\mathrm{MDC}}$ and $\mathsf{NPVC}^{\chi}_{\mathrm{LES}}$, algorithms Compute and Verify are nearly the same as those algorithms of $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

$$\det(M_{pre}^{\chi})\begin{cases}=\prod_{i=1}^{n}M_{pre}^{\chi}(i,i)-M_{pre}^{\chi}(z,z')\cdot M_{pre}^{\chi}(z',z)\cdot\prod_{i=1}^{(z-1)}M_{pre}^{\chi}(i,i)\cdot\prod_{i=(z+1)}^{(z'-1)}M_{pre}^{\chi}(i,i)\cdot\prod_{i=(z'+1)}^{n}M_{pre}^{\chi}(i,i)\quad(\text{Type}-\text{I})\\ =(-1)^{\frac{(n-2)\cdot(n+5)}{2}}\cdot(M_{pre}^{\chi}((n-z'+1),z)\cdot M_{pre}^{\chi}((n-z+1),z')\cdot\prod_{i=1}^{(n-z')}M_{pre}^{\chi}(i,(n-i+1))\cdot\\ \quad\prod_{i=(n-z'+2)}^{(n-z)}M_{pre}^{\chi}(i,(n-i+1))\cdot\prod_{i=(n-z+2)}^{n}M_{pre}^{\chi}(i,(n-i+1))-\prod_{i=1}^{n}M_{pre}^{\chi}(i,(n-i+1)))\quad(\text{Type}-\text{II})\end{cases},\tag{6a}$$

$$\det(M_{post}^{\chi})\begin{cases}=\prod_{i=1}^{n}M_{post}^{\chi}(i,i)-M_{post}^{\chi}(z',z)\cdot M_{post}^{\chi}(z,z')\cdot\prod_{i=1}^{(z-1)}M_{post}^{\chi}(i,i)\cdot\prod_{i=(z+1)}^{(z'-1)}M_{post}^{\chi}(i,i)\cdot\prod_{i=(z'+1)}^{n}M_{post}^{\chi}(i,i)\quad(\text{Type}-\text{I})\\ =(-1)^{\frac{(n-2)\cdot(n+5)}{2}}\cdot(M_{post}^{\chi}(z,(n-z'+1))\cdot M_{post}^{\chi}(z',(n-z+1))\cdot\prod_{i=1}^{(z-1)}M_{post}^{\chi}(i,(n-i+1))\cdot\\ \quad\prod_{i=(z+1)}^{(z'-1)}M_{post}^{\chi}(i,(n-i+1))\cdot\prod_{i=(z'+1)}^{n}M_{post}^{\chi}(i,(n-i+1))-\prod_{i=1}^{n}M_{post}^{\chi}(i,(n-i+1)))\quad(\text{Type}-\text{II})\end{cases}.\tag{6b}$$

For $\mathsf{NPVC}_{\mathrm{MDC}}$, a *direct* modification, denoted by $\mathsf{NPVC}_{\mathrm{MDC}}^{\chi}$'s Version-I, depends on the original construction. It can be stated as follows:

- KeyGen(1^{λ}, n): In this algorithm, $\mathcal{C}$ following Definition 12 randomly chooses a new χ-sparse matrix pair (M_{pre}^{χ}, M_{post}^{χ}) of size $(n{+}m)\times(n{+}m)$ *not* two permutation matrices P_1 and P_2. The other steps are the same as those of $\mathsf{NPVC}_{\mathrm{MDC}}$.
- ProbGen(k, X): In this algorithm, $\mathcal{C}$ still builds an $(n{+}m)\times(n{+}m)$ enlarged matrix E for an $n\times n$ plaintext matrix X (see Subsection 4.1 about $\mathsf{NPVC}_{\mathrm{MDC}}$) and generates a masked matrix $E'{=}M_{pre}^{\chi}{\cdot}E{\cdot}M_{adj\text{-}post}^{\chi}$ following Definition 12.
- Compute(f_{LU}, E'): See $\mathsf{NPVC}_{\mathrm{MDC}}$ about this algorithm.
- Verify(k, E', (L, U)): After receiving two decomposed matrices L and U from $\mathcal{W}$, $\mathcal{C}$ also computes a value $y'{=}\prod_{i=1}^{(n+m)}L(i,\ i){\cdot}U(i,\ i)$ and obtains a solution $y{=}\frac{y'}{\det(M_{pre}^{\chi})\cdot\det(M_{adj\text{-}post}^{\chi})\cdot\det(D)}$ (see Lemma 5 about the computations of $\det(M_{pre}^{\chi})$ and $\det(M_{adj\text{-}post}^{\chi})$). The result-verification process is the same as that of $\mathsf{NPVC}_{\mathrm{MDC}}$.

Of course, a *more efficient* modification, denoted by $\mathsf{NPVC}_{\mathrm{MDC}}^{\chi}$'s Version-II, does *not* need to build an $(n{+}m)\times(n{+}m)$ matrix E in ProbGen. We define it as follows:

- KeyGen(1^{λ}, n): This algorithm is the same as that in Definition 12. In this algorithm, $\mathcal{C}$ simply chooses a new pair (M_{pre}^{χ}, $M_{adj\text{-}post}^{\chi}$) of size $n\times n$.
- ProbGen(k, X): This algorithm is also the same as that in Definition 12.
- Compute(f_{LU}, E'): See $\mathsf{NPVC}_{\mathrm{MDC}}$ about this algorithm.
- Verify(k, E', (L, U)): After receiving two decomposed matrices L and U from $\mathcal{W}$, $\mathcal{C}$ can achieve a solution y according to $y'{=}\prod_{i=1}^{n}L(i,\ i){\cdot}U(i,\ i)$ and $y{=}\frac{y'}{\det(M_{pre}^{\chi})\cdot\det(M_{adj\text{-}post}^{\chi})}$ (see Lemma 5 about the computations of $\det(M_{pre}^{\chi})$ and $\det(M_{adj\text{-}post}^{\chi})$). The result-verification process is the same as that of $\mathsf{NPVC}_{\mathrm{MDC}}$.

From the above descriptions about algorithms of two types of modifications, we believe that Version-II is *much simpler than* Version-I. The detailed efficiency comparison is given in Subsection 6.5.

For $\mathsf{NPVC}_{\mathrm{LES}}$, we denote the proposed SMM construction -based modification by $\mathsf{NPVC}_{\mathrm{LES}}^{\chi}$. Based on Definition 12, it can be described as follows:

- KeyGen(1^{λ}, n): In this algorithm, $\mathcal{C}$ randomly generate a new χ-sparse matrix pair (M_{pre}^{χ}, M_{post}^{χ}) of size $n\times n$ (*not* two random sparse matrices M_1 and M_2) and a new vector $\vec{g}$ of size $n\times 1$. Note that $\vec{g}$ is a blinding coefficient vector that satisfies $\vec{g}(i,\ 1){\in}[-(2^{\eta}{-}1),\ 0)\bigcup(0,\ 2^{\eta}{-}1]\ \forall i{\in}\{1,\ 2,\ldots,\ n\}$. Then, $k{=}\{(M_{pre}^{\chi},\ M_{post}^{\chi}),\ \vec{g}\}$.
- ProbGen(k, Φ_{LE}): In this algorithm, for an LE problem $\Phi_{\mathrm{LE}}{=}(A,\ \vec{s})$, $\mathcal{C}$ employs k to output $A'{=}M_{pre}^{\chi}{\cdot}A{\cdot}M_{adj\text{-}post}^{\chi}$ and $\vec{s}^{*}{=}M_{pre}^{\chi}{\cdot}\vec{s}'$, where $\vec{s}'{=}A{\cdot}\vec{g}{+}\vec{s}$. Then, $\Phi'_{\mathrm{LE}}{=}(A',\ \vec{s}^{*})$.
- Compute(f_{LE}, Φ'_{LE}): See $\mathsf{NPVC}_{\mathrm{LES}}$ about this algorithm.
- Verify(k, Φ'_{LE}, $\vec{x}'$): After receiving a result $\vec{x}'$ from $\mathcal{W}$, $\mathcal{C}$ also follows the result-verification process of $\mathsf{NPVC}_{\mathrm{LES}}$ to check $\vec{x}'$. If the verification is passed, $\mathcal{C}$ can obtain a solution $\vec{x}{=}M_{adj\text{-}post}^{\chi}{\cdot}\vec{x}'{-}\vec{g}$.

Note that $\mathcal{C}$ does *not* need to compute the inverse matrix in *all* modified protocols. This is crucial for our SMM construction-based modifications. Since the main difference between the modified protocols $\mathsf{NPVC}_{\mathrm{MDC}}^{\chi}$ and $\mathsf{NPVC}_{\mathrm{LES}}^{\chi}$ and the original protocols $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$ is only the used SMM technique, some properties, e.g., correctness and security, of these protocols are the same. This implies that $\mathsf{NPVC}_{\mathrm{MDC}}^{\chi}$ and $\mathsf{NPVC}_{\mathrm{LES}}^{\chi}$ are *correct* and *secure* when $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$ hold these properties. However, for $\mathsf{NPVC}_{\mathrm{MDC}}^{\chi}$ and $\mathsf{NPVC}_{\mathrm{LES}}^{\chi}$, some other properties related to our SMM construction (i.e., privacy and efficiency) need to be carefully explored.

6.4 Privacy Analysis

We next analyze the privacy of the modified protocols. Specifically, since the proposed SMM construction is used to mask the matrix, the privacy related to other input and output (i.e., y, $\vec{s}$ and $\vec{x}$) of $\mathsf{NPVC}_{\mathrm{MDC}}^{\chi}$ and $\mathsf{NPVC}_{\mathrm{LES}}^{\chi}$ obviously follows from that of $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$. Then, IND-CPA privacy with respect to density is explored below.

Lemma 7. Let X be an $n\times n$ nonsingular matrix, and let X' be a masked output from ProbGen of the modified protocol. Assume that, for M_{pre}^{χ} and $M_{adj\text{-}post}^{\chi}$, $M_{pre}^{\chi}(i,\ h)\neq 0$, $M_{pre}^{\chi}(i,\ h')\neq 0$, $M_{adj\text{-}post}^{\chi}(h_{adj},\ j)\neq 0$ and $M_{adj\text{-}post}^{\chi}(h'_{adj},\ j)\neq 0$ $\forall i,\ j{\in}\{1,\ 2,\ldots,\ n\}$. Then, X' must satisfy $\forall i,\ j{\in}\{1,\ 2,\ldots,\ n\}: X'(i,\ j) = x'_1 + x'_2 + \Delta X'(i,\ j)$, where $x'_1 = M_{pre}^{\chi}(i,\ i/(n{-}i{+}1))\cdot M_{adj\text{-}post}^{\chi}(j/(n{-}j{+}1),\ j)\cdot X(i/(n{-}i{+}1),\ j)$, $x'_2 = ((M_{pre}^{\chi}(i,\ h)\cdot X(h,\ j) + M_{pre}^{\chi}(i,\ h')\cdot X(h',\ j))\cdot M_{adj\text{-}post}^{\chi}(j,\ j) + (M_{adj\text{-}post}^{\chi}(h_{adj},\ j)\cdot X(i/(n{-}i{+}1),\ h_{adj}) + M_{adj\text{-}post}^{\chi}(h'_{adj},\ j)\cdot X(i/(n{-}i{+}1),\ h'_{adj}))\cdot M_{pre}^{\chi}(i,\ i/(n{-}i{+}1)))$ and $\Delta X'(i,\ j){=}(M_{pre}^{\chi}(i,\ h){\cdot}X(h,\ h_{adj}){+}M_{pre}^{\chi}(i,\ h'){\cdot}X(h',\ h_{adj})){\cdot}M_{adj\text{-}post}^{\chi}(h_{adj},\ j){+}(M_{pre}^{\chi}(i,\ h){\cdot}X(h,\ h'_{adj}){+}M_{pre}^{\chi}(i,\ h'){\cdot}X(h',\ h'_{adj})){\cdot}M_{adj\text{-}post}^{\chi}(h'_{adj},\ j)$ with $X(h,\ h_{adj}){\neq}0$ and $X(h',\ h'_{adj}){\neq}0$.

Proof: See Appendix G for the proof. □

Theorem 5. The protocol $\mathsf{NPVC}_{f_{\mathrm{MC}}}^{\chi}$ that envelops itself into the proposed SMM construction is IND-CPA private with respect to density.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

Proof: The proof follows from a series of hybrid games between a challenger and $\widetilde{\mathcal{W}}_{\mathcal{A}}$ (see Definition 9 about the initial game). In particular, let $\vartheta(\lambda, n) = \mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}, \lambda, n)$, where $\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}$ denotes $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$ or $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$. We show that $\vartheta(\lambda, n)$ must be negligible.

Game₀: Let this game be the game of Definition 9. We claim that $\mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_0), \lambda, n) = \vartheta(\lambda, n)$.

Game₁: According to Lemma 7, we define Game_1 by modifying $\mathsf{ProbGen}$ of $\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_0)$ so that $\forall i, j \in \{1, 2, \ldots, n\} : X'(i, j) = X(i/(n-i+1), j) + \Delta X'(i, j)$. This implies that the challenger uses an $n \times n$ matrix $\Delta X'$ to (additively) mask X. Then, we have $\mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_0), \lambda, n) \leq \mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_1), \lambda, n)$.

Game₂: From Lemma 7, $X(h, h_{adj}) \neq 0$ and $X(h', h'_{adj}) \neq 0$. Then, we continue to modify Game_1 so as to redefine $\Delta X'(i, j) = M^{\mathcal{X}}_{pre}(i, h) \cdot M^{\mathcal{X}}_{adj\text{-}post}(h_{adj}, j) + M^{\mathcal{X}}_{pre}(i, h') \cdot M^{\mathcal{X}}_{adj\text{-}post}(h'_{adj}, j)$. That is, the challenger now fixes $X(h, h_{adj}) = X(h', h'_{adj}) = 1$ and $X(h', h_{adj}) = X(h, h'_{adj}) = 0$ to generate $\Delta X'(i, j)$. Hence, $\mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_1), \lambda, n) \leq \mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_2), \lambda, n)$.

Game₃: In this game, we modify the definition of $\Delta X'(i, j)$ yet again. To obtain $X'(i, j)$, the challenger replaces $M^{\mathcal{X}}_{pre}(i, h) \cdot M^{\mathcal{X}}_{adj\text{-}post}(h_{adj}, j)$ and $M^{\mathcal{X}}_{pre}(i, h') \cdot M^{\mathcal{X}}_{adj\text{-}post}(h'_{adj}, j)$ with two nonzero truly random values $\theta_1 \overset{\$}{\leftarrow} \Psi(\lambda)$ and $\theta_2 \overset{\$}{\leftarrow} \Psi(\lambda)$, where $\Psi(\lambda) := [-(2^{2 \cdot \eta} - 2^{\eta+1} + 1), 0) \bigcup (0, 2^{2 \cdot \eta} - 2^{\eta+1} + 1]$. This implies that $\Delta X'(i, j) = \theta_1 + \theta_2$. Therefore, for $\widetilde{\mathcal{W}}_{\mathcal{A}}$, we must have $\mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_2), \lambda, n) - \mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_3), \lambda, n) \leq \mathsf{negl}(\lambda)$.

Combined with the previous analyses on $\widetilde{\mathcal{W}}_{\mathcal{A}}$'s success advantages between games Game_0 and Game_3, we find that $\vartheta(\lambda, n) \leq \mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_3), \lambda, n) + \mathsf{negl}(\lambda)$. Moreover, in Game_3, since $\Delta X'$ can be regarded as a uniformly random matrix for $\widetilde{\mathcal{W}}_{\mathcal{A}}$, where $\forall i, j \in \{1, 2, \ldots, n\} : \Delta X'(i, j) \overset{\$}{\leftarrow} [-(2^{2 \cdot \eta+1} - 2^{\eta+2} + 2), 2^{2 \cdot \eta+1} - 2^{\eta+2} + 2]$, X' is also uniformly random. Specifically, $[-(2^{2 \cdot \eta+1} - 2^{\eta+2} + 2), 2^{2 \cdot \eta+1} - 2^{\eta+2} + 2] \supset [-(2^{\lambda} - 1), 2^{\lambda} - 1]$. This means that $\mathsf{Adv}^{\text{priv-dens}}_{\widetilde{\mathcal{W}}_{\mathcal{A}}}(\mathsf{NPVC}^{\mathcal{X}}_{f_{\text{MC}}}(\mathsf{Game}_3), \lambda, n) < \frac{1}{2^{(1+\lambda)} - 1}$. Therefore, $\vartheta(\lambda, n) < \frac{1}{2^{(1+\lambda)} - 1} + \mathsf{negl}(\lambda) = \mathsf{negl}(\lambda)$, which concludes the proof. □

6.5 Efficiency Comparisons

The proposed SMM construction is *only* employed to make $\mathsf{NPVC}_{\text{MDC}}$ and $\mathsf{NPVC}_{\text{LES}}$ hold IND-CPA privacy with respect to density. This implies that the communication complexities over outsourced data and $\mathcal{W}$'s computational complexities of the modified protocols $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$ involving Version-I and Version-II (see Subsection 6.3) and $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$ are at most the same as those of $\mathsf{NPVC}_{\text{MDC}}$ and $\mathsf{NPVC}_{\text{LES}}$. Then, we present the computational complexities of $\mathcal{C}$'s detailed algorithms of $\mathsf{NPVC}_{\text{MDC}}$, $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$, $\mathsf{NPVC}_{\text{LES}}$ and $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$ in Table 3, respectively.

The results in Table 3 show that $\mathcal{C}$'s computational complexities of $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$'s Version-I and $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$ are the same as those of $\mathsf{NPVC}_{\text{MDC}}$ and $\mathsf{NPVC}_{\text{LES}}$. This illustrates that our modifications can maintain the efficiency of $\mathsf{NPVC}_{\text{MDC}}$ and $\mathsf{NPVC}_{\text{LES}}$. $\mathcal{C}$'s computational complexity of $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$'s Version-II is reduced, which makes it more efficient compared with $\mathsf{NPVC}_{\text{MDC}}$.

TABLE 3
Computational Complexities of $\mathcal{C}'$s Algorithms†

$\mathcal{C}'$s Algorithm	$\mathsf{NPVC}_{\text{MDC}}$ [21]	$\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$		$\mathsf{NPVC}_{\text{LES}}$ [7]	$\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$
		I	II		
ProbGen	$O(n^2+(n+1) \cdot m)$		$O(n^2)$		
Verify	Checking	Solving	Checking		Solving
	$O((n+m)^2)$	$O(n+m)$	$O(n^2)$		$O(n)$

† The computational complexities are determined by the operation of multiplication.

For discussing $\mathcal{C}$'s concrete efficiency, we summarize $\mathcal{C}$'s computational costs for $\mathsf{NPVC}_{\text{MDC}}$, $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$, $\mathsf{NPVC}_{\text{LES}}$ and $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$ when the size of the plaintext matrix n satisfies $n \in \{8000, 9000, 10000, 11000, 12000\}$. Specifically, all the experiments for these NPVC protocols were run under the same implementation environment as that in Subsection 5.3 to obtain $\mathcal{C}$'s costs when the plaintext matrix is of size n. The computational timings from the experiments are presented in Table 4 and 5, and the corresponding growth trends of these timings are shown in Figure 1.

From Table 4, for $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$ and $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$, the computational costs of $\mathcal{C}$'s detailed algorithms (i.e., KeyGen, $\mathsf{ProbGen}$ and Verify) are given. As shown in this table, KeyGen is fast for each NPVC protocol, and the computational cost can be considered practically acceptable. From Table 5, when compared with $\mathsf{NPVC}_{\text{MDC}}$, $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$'s Version-I, in which $\mathcal{C}$ still needs to build an enlarged matrix for a plaintext matrix, does not cut down $\mathcal{C}$'s total cost, but $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$'s Version-II, where $\mathcal{C}$ does not construct an enlarged matrix, largely reduces $\mathcal{C}$'s total cost. This demonstrates that our SMM construction-based Version-II is *more suitable for* real applications. For $\mathsf{NPVC}_{\text{LES}}$ and $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$, there is only a small difference between their timings. This confirms that our modification for $\mathsf{NPVC}_{\text{LES}}$ can also achieve a low computational cost. In order to show $\mathcal{C}$'s efficiency comparisons between the outsourced computations (i.e., f_{MD} and f_{LE}), the original protocols and the modified protocols visually, the growth trends of $\mathcal{C}$'s total costs in f_{MD}, f_{LE}, $\mathsf{NPVC}_{\text{MDC}}$, $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$, $\mathsf{NPVC}_{\text{LES}}$ and $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$ are presented in Figure 1. This figure directly illustrates that our SMM construction-based modifications are efficient.

Moreover, to show $\mathcal{C}$'s cost gain, two cost ratios, denoted by $\frac{f_{\text{MD}}}{\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}}$ and $\frac{f_{\text{LE}}}{\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}}$, are presented in Table 5 and Figure 2. Most notably, these two cost ratios related to the availability of the NPVC protocols must satisfy $\frac{f_{\text{MD}}}{\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}} > 1$ and $\frac{f_{\text{LE}}}{\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}} > 1$ [21]. As evidenced by Table 5 and Figure 2, all values of the two cost ratios are greater than 1, and the value of $\frac{f_{\text{LE}}}{\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}}$ is much greater than 1. This illustrates that the modified protocols $\mathsf{NPVC}^{\mathcal{X}}_{\text{MDC}}$ and $\mathsf{NPVC}^{\mathcal{X}}_{\text{LES}}$ can achieve the considerable speedup for $\mathcal{C}$ when the size n increases.

7 Conclusions

In this work, we first address the privacy issues of the SMM-based NPVC protocols [7], [14], [15], [20], [21], [22] that we

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

TABLE 4
Computational Costs of $\mathcal{C}'$s Algorithms (ms)

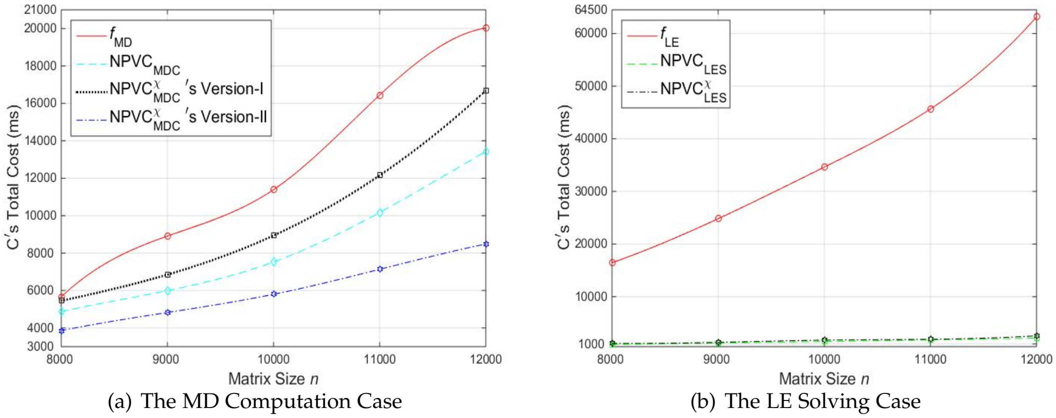
n	$\text{NPVC}^{\chi}_{\text{MDC}}$ ‡						$\text{NPVC}^{\chi}_{\text{LES}}$		
	Version-I			Version-II					
	KeyGen	ProbGen	Verify	KeyGen	ProbGen	Verify	KeyGen	ProbGen	Verify
8000	425.1	3814.5	1211.7	255.1	706.0	2895.4	244.3	855.1	28.2
9000	541.9	4846.7	1460.0	373.7	890.1	3560.6	278.4	1000.2	36.1
10000	669.1	6442.5	1824.8	392.2	1110.4	4294.2	347.7	1337.1	46.8
11000	781.2	9069.6	2300.7	530.9	1373.6	5228.2	311.9	1533.3	57.4
12000	907.7	12921.4	2857.7	630.7	1610.3	6243.9	571.5	1910.8	75.7

‡ l=10 in Verify of $\text{NPVC}^{\chi}_{\text{MDC}}$'s Version-I, and l=40 in Verify of $\text{NPVC}^{\chi}_{\text{MDC}}$'s Version-II.

TABLE 5
Comparisons of $\mathcal{C}'$s Total Computational Costs (ms)

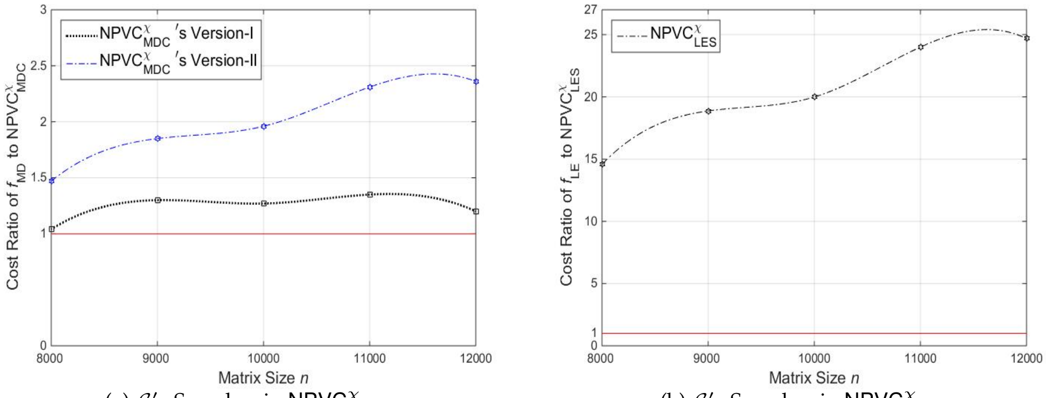
n	f_{MD} £	NPVC_{MDC} § [21]	$\text{NPVC}^{\chi}_{\text{MDC}}$ ¶		$\frac{f_{\text{MD}}}{\text{NPVC}^{\chi}_{\text{MDC}}}$		f_{LE} £	NPVC_{LES} ♭ [7]	$\text{NPVC}^{\chi}_{\text{LES}}$	$\frac{f_{\text{LE}}}{\text{NPVC}^{\chi}_{\text{LES}}}$
			Version-I	Version-II	Version-I	Version-II				
8000	5658.4	4874.3	5451.3	3856.5	1.04	1.47	16445.0	919.0	1127.6	14.58
9000	8909.3	5992.6	6848.6	4824.4	1.30	1.85	24812.8	1186.5	1314.7	18.87
10000	11389.6	7525.0	8936.4	5796.8	1.27	1.96	34615.7	1471.0	1731.6	19.99
11000	16445.1	10171.8	12151.5	7132.7	1.35	2.31	45666.4	1773.6	1902.6	24.00
12000	20025.8	13428.9	16686.8	8484.9	1.20	2.36	63167.4	2109.2	2558.0	24.69

£ The outsourced computations f_{MD} and f_{LE} are computed by $\mathcal{C}$ locally.
§These costs are lower bounds of the computational costs for NPVC_{MDC}. Specifically, l=10 in NPVC_{MDC}.
¶ l=10 in Verify of $\text{NPVC}^{\chi}_{\text{MDC}}$'s Version-I, and l=40 in Verify of $\text{NPVC}^{\chi}_{\text{MDC}}$'s Version-II.
♭ $N^{M_1}_{ma\text{-}nz}$=$N^{M_2}_{ma\text{-}nz}$=4 for NPVC_{LES}.



(a) The MD Computation Case (b) The LE Solving Case

Fig. 1. $\mathcal{C}'$s Efficiency Comparisons Between Outsourced Computations, Original Protocols and Modified Protocols



(a) $\mathcal{C}'$s Speedup in $\text{NPVC}^{\chi}_{\text{MDC}}$ (b) $\mathcal{C}'$s Speedup in $\text{NPVC}^{\chi}_{\text{LES}}$

Fig. 2. $\mathcal{C}'$s Speedup in Modified Protocols

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

know of according to the proposed concept of privacy with respect to density. We show that all of these protocols are not IND-COA private with respect to density. Since $\mathsf{NPVC}_{\mathrm{MDC}}$ [21] and $\mathsf{NPVC}_{\mathrm{LES}}$ [7] are state-of-the-art among the other protocols, we give detailed analyses of these two protocols. The analysis result on $\mathsf{NPVC}_{\mathrm{LES}}$ addresses Chen et al.'s pending problem [7]. Then, in order to make $\mathsf{NPVC}_{\mathrm{MDC}}$ and $\mathsf{NPVC}_{\mathrm{LES}}$ hold privacy with respect to density, we introduce a new SMM construction that combines the χ-sparse matrix pair with the adaptive adjustment function. We prove that, using our SMM construction, the modified protocols $\mathsf{NPVC}^{\chi}_{\mathrm{MDC}}$ and $\mathsf{NPVC}^{\chi}_{\mathrm{LES}}$ hold IND-CPA privacy with respect to density. Finally, we present the comparisons between the modified protocols and the original protocols to show that the modified protocols that do *not* require a $\mathcal{C}$ operating matrix inversion can keep the same level of high performance as the originals.

In our future work, we will explore the efficient SMM-based NPVC protocols for other matrix-related computations (e.g., the MI computation) that can hold IND-CPA privacy with respect to density.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61302161, in part by the Doctoral Fund, Ministry of Education, China, under Grant 20130181120076. The first author of this research Liang Zhao is supported by the International Visiting Program for Excellent Young Scholars of SCU.

References

[1] P. Ananth, N. Chandran, V. Goyal, B. Kanukurthi, and R. Ostrovsky, "Achieving privacy in verifiable computation with multiple servers-without FHE and without pre-processing," in *PKC*, 2014, pp. 149–166.

[2] M.J. Atallah, K.N. Pantazopoulos, J.R. Rice, and E.E. Spafford, "Secure outsourcing of scientific computations," *Adv. Comput.*, vol. 54, pp. 215–272, Jan. 2002.

[3] D. Benjamin and M.J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in *PST*, 2008, pp. 240–245.

[4] M. Blanton, M.J. Atallah, K.B. Frikken, and Q. Malluhi, "Secure and efficient outsourcing of sequence comparisons," in *ESORICS*, 2012, pp. 505–522.

[5] B. Chevallier-Mames, J.S. Coron, N. McCullagh, D. Naccache, and M. Scott, "Secure delegation of elliptic-curve pairing," in *CARDIS*, 2010, pp. 24–35.

[6] S. Canard, J. Devigne, and O. Sanders, "Delegating a pairing can be both secure and efficient," in *ACNS*, 2014, pp. 549–565.

[7] X. Chen, X. Huang, J. Li, J. Ma, W. Lou, and D.S. Wong, "New algorithms for secure outsourcing of large-scale systems of linear equations," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 69–78, Jan. 2015.

[8] S.G. Choi, J. Katz, R. Kumaresan, and C. Cid, "Multi-client non-interactive verifiable computation," in *TCC*, 2013, pp. 499–518.

[9] K.M. Chung, Y. Kalai, and S. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *CRYPTO*, 2010, pp. 483–501.

[10] X. Chen, J. Li, X. Huang, J. Ma, and W. Lou, "New publicly verifiable databases with efficient updates," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 5, pp. 546–556, Sep./Oct. 2015.

[11] X. Chen, J. Li, J. Ma, Q. Tang, and W. Lou, "New algorithms for secure outsourcing of modular exponentiations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2386–2396, Sep. 2014.

[12] X. Chen, J. Li, J. Weng, J. Ma, and W. Lou, "Verifiable computation over large database with incremental updates," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 3184–3195, Oct. 2016.

[13] D. Chaum and T.P. Pedersen, "Wallet databases with observers," in *CRYPTO*, 1992, pp. 89–105.

[14] F. Chen, T. Xiang, X. Lei, and J. Chen, "Highly efficient linear regression outsourcing to a cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 4, pp. 499–508, Oct./Dec. 2014.

[15] F. Chen, T. Xiang, and Y. Yang, "Privacy-preserving and verifiable protocols for scientific computation outsourcing to the cloud," *J. Parallel Distrib. Comput.*, vol. 74, no. 3, pp. 2141–2151, Mar. 2014.

[16] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: outsourcing computation to untrusted workers," in *CRYPTO*, 2010, pp. 465–482.

[17] S.D. Gordon, J. Katz, F. Liu, E. Shi, and H. Zhou, "Multi-client verifiable computation with stronger security guarantees," in *TCC*, 2015, pp. 144–168.

[18] S. Hohenberger and A. Lysyanskaya, "How to securely outsource cryptographic computations," in *TCC*, 2005, pp. 264–282.

[19] R.S. Katti, S.K. Srinivasan, and A. Vosoughi, "On the security of randomized arithmetic codes against ciphertext-only attacks," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 19–27, Mar. 2011.

[20] X. Lei, X. Liao, T. Huang, and F. Heriniaina, "Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud," *Inform. Sci.*, vol. 280, pp. 205–217, Oct. 2014.

[21] X. Lei, X. Liao, T. Huang, and H. Li, "Cloud computing service: the case of large matrix determinant computation ," *IEEE Trans. Serv. Comput.*, vol. 8, no. 5, pp. 688–700, Sep./Oct. 2015.

[22] X. Lei, X. Liao, T. Huang, H. Li, and C. Hu, "Outsourcing large matrix inversion computation to a public cloud," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 78–87, Jan./Jun. 2013.

[23] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: nearly practical verifiable computation," in *IEEE SP*, 2013, pp. 238–252.

[24] S. Salinas, C. Luo, X. Chen, and P. Li, "Efficient secure outsourcing of large-scale linear systems of equations," in *INFOCOM*, 2015, pp. 1035–1043.

[25] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *INFOCOM*, 2011, pp. 820–828.

[26] C. Wang, K. Ren, J. Wang, and Q. Wang, "Harnessing the cloud for securely outsourcing large-scale systems of linear equations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 6, pp. 1172–1181, Jun. 2013.

[27] Y. Yu, Y. Luo, D. Wang, S. Fu, and M. Xu, "Efficient, secure and non-iterative outsourcing of large-scale systems of linear equations," in *ICC*, 2016, pp. 1–6.

[28] K. Zhou, M.H. Afifi, and J. Ren, "ExpSOS: secure and verifiable outsourcing of exponentiation operations for mobile cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2518–2531, Nov. 2017.

[29] Y. Zhang and M. Blanton, "Efficient secure and verifiable outsourcing of matrix multiplications," in *ISC*, 2014, pp. 158–178.

[30] Z. Zhang, X. Chen, J. Ma, and J. Shen, "SLDS: secure and location-sensitive data sharing scheme for cloud-assisted cyber-physical systems," *Future Generat. Comput. Syst.*, 2018. [Online]. Available: https://doi.org/10.1016/j.future.2018.01.025

[31] X. Zhang, T. Jiang, K.C. Li, A. Castiglione, and X. Chen, "New publicly verifiable computation for batch matrix multiplication," *Inform. Sci.*, 2017. [Online]. Available: https://doi.org/10.1016/j.ins.2017.11.063

Appendix A
Proof of Lemma 1

Proof: W.l.o.g., consider $X^{pre}=M^{msk}\cdot X$. Because $M^{msk} \in SM^{n\times n}$, $\forall i \in \{1,\ldots,n\}$ and $\exists! j \in \{1,\ldots,n\}$:$M^{msk}(i, j) \neq 0$. This implies that M^{msk} can be equal to an $n\times n$ identity matrix I by premultiplying a series of elementary matrices R^{rs}_1, R^{rs}_2, $\ldots$, $R^{rs}_{n'}$ and R^{rm}_1, R^{rm}_2, $\ldots$, R^{rm}_n, i.e., $M^{msk} = R^{rs}_1 \cdot R^{rs}_2 \ldots\cdot R^{rs}_{n'} \cdot R^{rm}_1 \cdot R^{rm}_2 \ldots\cdot R^{rm}_n \cdot I$, where R^{rs}_1, $R^{rs}_2,\ldots,R^{rs}_{n'}$ are matrices for the row-switching transformation and R^{rm}_1, $R^{rm}_2,\ldots,R^{rm}_n$ are matrices for the row-multiplying transformation. Then, $X^{pre}=(R^{rs}_1\cdot R^{rs}_2\ldots\cdot R^{rs}_{n'})\cdot(R^{rm}_1\cdot R^{rm}_2\ldots\cdot R^{rm}_n)\cdot X$. According to the definition of the matrix set $\{R^{rm}_1, R^{rm}_2,\ldots,R^{rm}_n\}$, the product $(R^{rm}_1\cdot R^{rm}_2\ldots\cdot R^{rm}_n)$ can be denoted by a diagonal

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

matrix R^{rm}_{diag}:=diag(r_1,...,r_n), where $r_i \neq 0$ $\forall i \in \{1, 2, \ldots, n\}$. Then, $X^{pre}=(R^{rs}_1 \cdot R^{rs}_2 \ldots \cdot R^{rs}_{n'}) \cdot R^{rm}_{diag} \cdot X$. For the matrix X^{rm} generated by $X^{rm}=R^{rm}_{diag} \cdot X$, $N^{X^{rm}}_0=N^X_0$. Moreover, the computations $(R^{rs}_1 \cdot R^{rs}_2 \ldots \cdot R^{rs}_{n'}) \cdot X^{rm}$ simply achieve the swapping for rows of the matrix X^{rm}. Therefore, $N^{X^{pre}}_0=N^X_0$.

A similar procedure is used to prove $N^{X^{post}}_0=N^X_0$ when $X^{post}=X \cdot M^{msk}$. Note that $X^{post} = X \cdot (R^{cm}_1 \cdot R^{cm}_2 \ldots \cdot R^{cm}_n) \cdot (R^{cs}_1 \cdot R^{cs}_2 \ldots \cdot R^{cs}_{n*})$ is employed by the proof, where R^{cm}_1, R^{cm}_2, ..., R^{cm}_n are elementary matrices for the column-multiplying transformation and R^{cs}_1, R^{cs}_2, ..., R^{cs}_{n*} are elementary matrices for the column-switching transformation. □

Appendix B Proof of Lemma 2

Proof: Assume that the plaintext matrix is X_0. According to ProbGen of NPVC$_{\mathrm{MDC}}$, $E_0=\begin{bmatrix} X_0 & B \\ 0 & D \end{bmatrix}$ and $E'_0 = P_1 \cdot E_0 \cdot {P_2}^{-1}$, where $P_1 \in SM^{(n+m)\times(n+m)}$ and $P_2 \in SM^{(n+m)\times(n+m)}$. $N^{E_0}_0$ involves four components: $N^{X_0}_0$, N^B_0, N^D_0 and N^0_0. Then, $N^{E_0}_0 = n^2-n+N^B_0+m^2-m+n \cdot m$. Because $N^B_0 \xleftarrow{\$} \{0, \ldots, n \cdot m\}$, $N^{E_0}_0 \in [(n^2-n+m^2-m+n \cdot m), (n^2-n+m^2-m+2 \cdot n \cdot m)]$. Moreover, the fact $P^{-1}_2 \in SM^{(n+m)\times(n+m)}$ can be drawn by applying the adjoint matrix-based matrix inversion approach straightforwardly. Then, based on Lemma 1, $N^{E'_0}_0=N^{E_0}_0$, which implies that $\Pr[N^{E'_0}_0 \in [(n^2-n+m^2-m+n \cdot m), (n^2-n+m^2-m+ 2 \cdot n \cdot m)]]=1$.

If the plaintext matrix is X_1, a similar proof procedure can be performed to show that $N^{E'_1}_0 \in [(m^2 - m + n \cdot m), (m^2 - m + 2 \cdot n \cdot m)]$ with probability 1, where $E'_1=P_1 \cdot E_1 \cdot {P_2}^{-1}$ and $E_1=\begin{bmatrix} X_1 & B \\ 0 & D \end{bmatrix}$. In particular, $N^{X_1}_0=0$, and $N^{E'_1}_0$ only considers N^B_0, N^D_0 and N^0_0. □

Appendix C Proof of Lemma 3

Proof: The hybrid argument is used to prove Lemma 3. Let $\Pr[N^{A'_0}_0 \geq 1]$ denote the probability of the general case that $\exists i$, $j \in \{1,2, \ldots, n\}$:$A'_0(i, j)$=0 when M_1 and M_2 are randomly generated. A series of hybrid games are defined as follows:

Game_0: Define this game to be identical to the above general case. If $\exists i$, $j \in \{1, \ldots, n\}$:$A'_0(i, j)$=0, Game_0 outputs 1. Let $\Pr[\mathsf{Game}_0(\lambda, n)=1]=\Pr[N^{A'_0}_0 \geq 1]$.

Game_1: We modify Game_0 so that $N^{M_1(i,*)}_{nz}=N^{M_1}_{ma\text{-}nz}$ $\forall i \in \{1, 2, \ldots, n\}$ and $N^{M_2(*,j)}_{nz}=N^{M_2}_{ma\text{-}nz}$ $\forall j \in \{1, 2, \ldots, n\}$. This implies that A'_0 is generated by $A'_0=M^*_1 \cdot A_0 \cdot M^*_2$, where M^*_1 and M^*_2 are sparse matrices with $N^{M_1}_{ma\text{-}nz}$ nonzero elements in each row and $N^{M_2}_{ma\text{-}nz}$ nonzero elements in each column, respectively. Hence, we must have $\Pr[\mathsf{Game}_1(\lambda, n)=1] \leq \Pr[\mathsf{Game}_0(\lambda, n)=1]$.

Game_2: We now define this game by restricting the condition to obtain $A'_0(i, j)$=0 in Game_1, where $i, j \in \{1, 2, \ldots, n\}$, so that $\exists i, j \in \{1,2, \ldots, n\}$, $\forall A^*_0(i, j') \neq 0$ and $\forall M^*_2(i', j) \neq 0 : j' \neq i'$, where $A^*_0=M^*_1 \cdot A_0$ and j', $i' \in \{1, 2, \ldots, n\}$. Then, we claim that $\Pr[\mathsf{Game}_2(\lambda, n)=1] \leq \Pr[\mathsf{Game}_1(\lambda, n)=1]$. Specifically, for this game,

$$\begin{aligned} &\Pr[\mathsf{Game}_2(\lambda, n) = 1] \\ &\geq \frac{n-N^{M_2}_{ma\text{-}nz}}{n} \cdot \frac{n-(N^{M_2}_{ma\text{-}nz}+1)}{n-1} \ldots \cdot \frac{n-(N^{M_2}_{ma\text{-}nz}+N^{M_1}_{ma\text{-}nz}-1)}{n-(N^{M_1}_{ma\text{-}nz}-1)}, \\ &= \frac{(n-N^{M_1}_{ma\text{-}nz})! \cdot (n-N^{M_2}_{ma\text{-}nz})!}{n! \cdot (n-(N^{M_2}_{ma\text{-}nz}+N^{M_1}_{ma\text{-}nz}))!} \end{aligned}$$

which we want to show.

The proof can be completed by combining the analyses from the above hybrid games. □

Appendix D Proof of Lemma 4

Proof: We also use the hybrid argument to prove Lemma 4. Let $\Pr[N^{A'_0}_0 \geq 1]$ be the probability about the general case that $\exists i$, $j \in \{1,2, \ldots, n\}$:$A'_0(i, j)$=0 when M_1 and M_2 are randomly generated. Then, a sequence of three games that are similar to the games from the proof of Lemma 3 (see Appendix C) are presented below:

Game_0: Define this game to be identical to the above general case. If $\exists i$, $j \in \{1, \ldots, n\}$:$A'_0(i, j)$=0, Game_0 outputs 1. Therefore, $\Pr[\mathsf{Game}_0(\lambda, n)=1]=\Pr[N^{A'_0}_0 \geq 1]$.

Game_1: This game is the same as Game_1 in the proof of Lemma 3. Hence, we can also obtain $\Pr[\mathsf{Game}_1(\lambda, n)=1] \leq \Pr[\mathsf{Game}_0(\lambda, n)=1]$.

Game_2: We define this game by restricting the condition to obtain $A'_0(i, j)$=0 in Game_1, where i, $j \in \{1, 2, \ldots, n\}$, so that $\exists i$, $j \in \{1,2, \ldots, n\}$, $\forall A^*_0(i, j')=0$ and $\forall M^*_2(i', j) \neq 0$:$j'=i'$, where $A^*_0=M^*_1 \cdot A_0$ and j', $i' \in \{1, 2, \ldots, n\}$. Then, we claim that $\Pr[\mathsf{Game}_2(\lambda, n)=1] \leq \Pr[\mathsf{Game}_1(\lambda, n)=1]$. Specifically, for this game, we have

$$\begin{aligned} &\Pr[\mathsf{Game}_2(\lambda, n) = 1] \\ &\geq \left(\frac{\frac{n}{2}-\varrho}{n} \cdot \frac{\frac{n}{2}-(\varrho+1)}{n-1} \cdot \frac{\frac{n}{2}-(\varrho+2)}{n-2} \ldots \cdot \frac{\frac{n}{2}-(\varrho+N^{M_1}_{ma\text{-}nz}-1)}{n-(N^{M_1}_{ma\text{-}nz}-1)}\right)^{N^{M_2}_{ma\text{-}nz}} . \\ &= \left(\frac{(\frac{n}{2}-\varrho)! \cdot (n-N^{M_1}_{ma\text{-}nz})!}{n! \cdot (\frac{n}{2}-(\varrho+N^{M_1}_{ma\text{-}nz}))!}\right)^{N^{M_2}_{ma\text{-}nz}} \end{aligned}$$

By combining the analyses from the above hybrid games, we can confirm the shown result in the lemma. □

Appendix E Proof of Lemma 5

Proof: For M^{χ}_{pre} and M^{χ}_{post} of size $n \times n$, det(M^{χ}_{pre}) = det($(M^{\chi}_{pre})^t$), and the matrix structure of M^{χ}_{post} is the same as that of $(M^{\chi}_{pre})^t$ when M^{χ}_{pre} and M^{χ}_{post} are the same type. Then, the calculation method of det(M^{χ}_{post}) is the same as that of det(M^{χ}_{pre}). In what follows, we consider the case det(M^{χ}_{post}) under the condition z, $z' \in \{2, 3, \ldots, (n-1)\}$.

- If M^{χ}_{post} is a Type-I matrix, det(M^{χ}_{post}) = $(-1)^{(1+1)} \cdot M^{\chi}_{post}(1, 1) \cdot \ldots (-1)^{(1+1)} \cdot M^{\chi}_{post}((z-1), (z-1)) \cdot (-1)^{(2+2)} \cdot M^{\chi}_{post}((z+1), (z+1)) \cdot \ldots (-1)^{(2+2)} \cdot M^{\chi}_{post}((z'-1), (z'-1)) \cdot (-1)^{(3+3)} \cdot M^{\chi}_{post}((z'+1), (z'+1)) \cdot \ldots (-1)^{(3+3)} \cdot M^{\chi}_{post}(n, n) \cdot (M^{\chi}_{post}(z, z) \cdot M^{\chi}_{post}(z', z') - M^{\chi}_{post}(z', z) \cdot M^{\chi}_{post}(z, z'))$, which implies that det(M^{χ}_{post}) = $\prod^{n}_{i=1} M^{\chi}_{post}(i, i) - M^{\chi}_{post}(z', z) \cdot M^{\chi}_{post}(z, z') \cdot \prod^{(z-1)}_{i=1} M^{\chi}_{post}(i, i) \cdot \prod^{(z'-1)}_{i=(z+1)} M^{\chi}_{post}(i, i) \cdot \prod^{n}_{i=(z'+1)} M^{\chi}_{post}(i, i)$.
- If M^{χ}_{post} is a Type-II matrix, det(M^{χ}_{post}) = $(-1)^{(n+1)} \cdot M^{\chi}_{post}(1, n) \cdot \ldots (-1)^{(n-z+3)} \cdot M^{\chi}_{post}((z-1), (n-z+2))$

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

$\cdot\ (-1)^{(n-z+2)} \cdot M^{\chi}_{post}((z+1),\ (n-z)) \cdot \ldots(-1)^{(n-z'+4)} \cdot M^{\chi}_{post}((z'-1), (n-z'+2)) \cdot (-1)^{(n-z'+3)} \cdot M^{\chi}_{post}((z'+1), (n-z')) \cdot \ldots(-1)^{(3+1)} \cdot M^{\chi}_{post}(n,\ 1) \cdot (M^{\chi}_{post}(z,\ (n-z'+1)) \cdot M^{\chi}_{post}(z',\ (n-z+1)) - M^{\chi}_{post}(z,\ (n-z+1)) \cdot M^{\chi}_{post}(z',\ (n-z'+1)))$. S-ince the sequence $\{(n+1),\ldots, (n-z+2),\ldots, (n-z'+3), 4\}$ is an arithmetic progression with a common difference of -1, $\det(M^{\chi}_{post}) = (-1)^{\frac{(n-2)\cdot(n+5)}{2}} \cdot (M^{\chi}_{post}(z,\ (n-z'+1)) \cdot M^{\chi}_{post}(z',\ (n-z+1)) \cdot \prod_{i=1}^{(z-1)} M^{\chi}_{post}(i,\ (n-i+1)) \cdot \prod_{i=(z+1)}^{(z'-1)} M^{\chi}_{post}(i,\ (n-i+1)) \cdot \prod_{i=(z'+1)}^{n} M^{\chi}_{post}(i,\ (n-i+1)) - \prod_{i=1}^{n} M^{\chi}_{post}(i,\ (n-i+1)))$. □

APPENDIX F
PROOF OF LEMMA 6

Proof: The proof is immediate. According to Lemma 5, for M^{χ}_{pre}, if the condition 1 or 2 in Lemma 6 is satisfied, $\det(M^{\chi}_{pre}) \neq 0$, which illustrates that M^{χ}_{pre} is invertible. Moreover, if M^{χ}_{post} holds for the condition 3 or 4 in Lemma 6, $\det(M^{\chi}_{post}) \neq 0$, which shows that M^{χ}_{post} is also invertible. □

APPENDIX G
PROOF OF LEMMA 7

Proof: The lemma can be proven by simply performing the computations $X'=M^{\chi}_{pre}\cdot X\cdot M^{\chi}_{adj\text{-}post}$. Note that, because the function adj$(\cdot,\cdot)$ is used to determine the column indices (i.e., h_{adj} and h'_{adj}) of two nonzero elements in the respective rows h and h' of X and complete $M^{\chi}_{post}(h,\ *)\leftrightarrow M^{\chi}_{post}(h_{adj},\ *)$ and $M^{\chi}_{post}(h',\ *)\leftrightarrow M^{\chi}_{post}(h'_{adj},\ *)$, $X(h,\ h_{adj})$ and $X(h',\ h'_{adj})$ *must be nonzero*, which implies that $M^{\chi}_{pre}(i,\ h)\cdot X(h,\ h_{adj})\cdot M^{\chi}_{adj\text{-}post}(h_{adj},\ j)\neq 0$ and $M^{\chi}_{pre}(i,\ h')\cdot X(h',\ h'_{adj})\cdot M^{\chi}_{adj\text{-}post}(h'_{adj},\ j)\neq 0$. □

APPENDIX H
TOY EXAMPLES OF χ-SPARSE MATRICES

We provide some 6×6 toy examples for showing the proposed χ-sparse matrices. Specifically, $M^{\chi}_{pre}(i,\ 2)\neq 0$, $M^{\chi}_{pre}(i,\ 4)\neq 0$, $M^{\chi}_{post}(2,\ j)\neq 0$ and $M^{\chi}_{post}(4,\ j)\neq 0$ $\forall i,\ j\in\{1,\ 2,\ldots,\ 6\}$. Note that those χ-sparse matrices are *not* row diagonally dominant matrices.

- Type-I χ-Sparse Matrices: $(M^{\chi}_{pre},\ M^{\chi}_{post})$=

$$\left(\begin{pmatrix} 7 & 2 & 0 & 5 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 & 0 \\ 0 & 8 & 6 & 2 & 0 & 0 \\ 0 & 10 & 0 & 9 & 0 & 0 \\ 0 & 5 & 0 & 8 & 3 & 0 \\ 0 & 17 & 0 & 6 & 0 & 12 \end{pmatrix}, \begin{pmatrix} 5 & 0 & 0 & 0 & 0 & 0 \\ 1 & 7 & 16 & 2 & 6 & 15 \\ 0 & 0 & 3 & 0 & 0 & 0 \\ 8 & 10 & 12 & 9 & 7 & 16 \\ 0 & 0 & 0 & 0 & 21 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 \end{pmatrix}\right).$$

- Type-II χ-Sparse Matrices: $(M^{\chi}_{pre},\ M^{\chi}_{post})$=

$$\left(\begin{pmatrix} 0 & 2 & 0 & 5 & 0 & 7 \\ 0 & 3 & 0 & 1 & 6 & 0 \\ 0 & 8 & 0 & 2 & 0 & 0 \\ 0 & 10 & 17 & 9 & 0 & 0 \\ 0 & 19 & 0 & 7 & 0 & 0 \\ 3 & 5 & 0 & 8 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 5 \\ 1 & 18 & 7 & 16 & 2 & 6 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 8 & 10 & 20 & 12 & 9 & 7 \\ 0 & 11 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}\right).$$

APPENDIX I
TOY EXAMPLES FOR ILLUSTRATION OF LEMMA 5

The determinants of 6×6 matrices in Appendix H are computed using Equation 6. Specifically, if M^{χ}_{pre} and M^{χ}_{post} are Type-I χ-sparse matrices, $\det(M^{\chi}_{pre}) = (7 \cdot 3 \cdot 6 \cdot 9 \cdot 3 \cdot 12) - (7 \cdot 1 \cdot 6 \cdot 10 \cdot 3 \cdot 12) = 25704$ and $\det(M^{\chi}_{post}) = (5 \cdot 7 \cdot 3 \cdot 9 \cdot 21 \cdot 7) - (5 \cdot 10 \cdot 3 \cdot 2 \cdot 21 \cdot 7) = 94815$. If M^{χ}_{pre} and M^{χ}_{post} are Type-II χ-sparse matrices, $\det(M^{\chi}_{pre}) = (-1)^{22} \cdot ((7 \cdot 6 \cdot 8 \cdot 17 \cdot 7 \cdot 3) - (7 \cdot 6 \cdot 2 \cdot 17 \cdot 19 \cdot 3)) = 38556$ and $\det(M^{\chi}_{post}) = (-1)^{22} \cdot ((5 \cdot 9 \cdot 3 \cdot 7 \cdot 11 \cdot 7) - (5 \cdot 2 \cdot 3 \cdot 20 \cdot 11 \cdot 7)) = 26565$.

APPENDIX J
TOY EXAMPLES FOR ILLUSTRATION OF LEMMA 6

We give examples according to 6×6 matrices in Appendix H. For M^{χ}_{pre}, assume that it is a Type-I χ-sparse matrix. $\det(M^{\chi}_{pre}) = (7 \cdot 6 \cdot 3 \cdot 12 \cdot M^{\chi}_{pre}(2,\ 2) \cdot M^{\chi}_{pre}(4,\ 4)) - (7 \cdot 6 \cdot 3 \cdot 12 \cdot M^{\chi}_{pre}(2,\ 4) \cdot M^{\chi}_{pre}(4,\ 2))$. Since $M^{\chi}_{pre}(2,\ 2) \cdot M^{\chi}_{pre}(4,\ 4) \neq M^{\chi}_{pre}(2,\ 4) \cdot M^{\chi}_{pre}(4,\ 2)$, we have $\det(M^{\chi}_{pre}) \neq 0$. This shows that M^{χ}_{pre} satisfying the condition 1 in Lemma 6 must be invertible. Assume that M^{χ}_{pre} is a Type-II χ-sparse matrix. $\det(M^{\chi}_{pre}) = (-1)^{22} \cdot ((7 \cdot 6 \cdot 17 \cdot 3 \cdot M^{\chi}_{pre}(3,\ 2) \cdot M^{\chi}_{pre}(5,\ 4)) - (7 \cdot 6 \cdot 17 \cdot 3 \cdot M^{\chi}_{pre}(3,\ 4) \cdot M^{\chi}_{pre}(5,\ 2)))$. Because $M^{\chi}_{pre}(3,\ 2) \cdot M^{\chi}_{pre}(5,\ 4) \neq M^{\chi}_{pre}(3,\ 4) \cdot M^{\chi}_{pre}(5,\ 2)$, we obtain $\det(M^{\chi}_{pre}) \neq 0$. This indicates that M^{χ}_{pre} satisfying the condition 2 in Lemma 6 must also be invertible. For M^{χ}_{post}, a similar discussion can be used to demonstrate our result in Lemma 6.

APPENDIX K
TOY EXAMPLE FOR ILLUSTRATION OF OUR SMM CONSTRUCTION

We provide a simple toy example for showing the details of our SMM construction as follows:

- According to KeyGen, choose and output a pair $(M^{\chi}_{pre},\ M^{\chi}_{post})$ of size 6×6=

$$\left(\begin{pmatrix} 2 & 1 & 0 & 5 & 0 & 0 \\ 0 & 5 & 0 & 7 & 0 & 0 \\ 0 & 2 & 3 & 2 & 0 & 0 \\ 0 & 8 & 0 & 3 & 0 & 0 \\ 0 & 9 & 0 & 8 & 5 & 0 \\ 0 & 3 & 0 & 1 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 6 & 0 & 0 & 0 & 0 & 0 \\ 3 & 9 & 1 & 5 & 8 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 8 & 6 & 5 & 1 & 9 & 3 \\ 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}\right).$$

- Upon input of $(M^{\chi}_{pre},\ M^{\chi}_{post})$ and a 6×6 nonsingular matrix X=

$$\begin{pmatrix} 7 & 0 & 2 & 0 & 0 & 3 \\ 0 & 9 & 0 & 0 & 1 & 5 \\ 1 & 2 & 8 & 2 & 0 & 0 \\ 2 & 0 & 0 & 3 & 9 & 0 \\ 0 & 0 & 5 & 0 & 7 & 0 \\ 0 & 1 & 0 & 1 & 0 & 6 \end{pmatrix}.$$

According to ProbGen, run $M^{\chi}_{adj\text{-}post}\leftarrow$adj$(X,\ M^{\chi}_{post})$ to obtain $M^{\chi}_{adj\text{-}post}$=

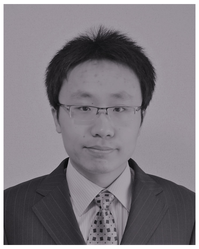
$$\begin{pmatrix} 1 & 6 & 5 & 8 & 9 & 3 \\ 5 & 9 & 1 & 3 & 8 & 2 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2018.2861699, IEEE Transactions on Dependable and Secure Computing

Then, compute $X'=M_{pre}^{\mathrm{X}}\cdot\mathrm{X}\cdot M_{adj\text{-}post}^{\mathrm{X}}$ to obtain $X'=$

$$\begin{pmatrix} 69 & 225 & 133 & 309 & 610 & 112 \\ 239 & 489 & 115 & 373 & 962 & 182 \\ 127 & 258 & 83 & 200 & 395 & 89 \\ 366 & 684 & 102 & 318 & 875 & 242 \\ 421 & 825 & 186 & 515 & 1604 & 300 \\ 142 & 264 & 38 & 124 & 326 & 104 \end{pmatrix}.$$



Liang Zhao (M′17) received his M.S. degree in computer science in 2009 from Chongqing University, China, and his Ph.D. degree in informatics in 2012 from Kyushu University, Japan. He is currently an assistant professor at the College of Cybersecurity, Sichuan University, China, and a visiting researcher in the Surrey Centre for Cyber Security, University of Surrey, U.K., and a researcher at the HIFIVE Lennon Laboratory, Chengdu HiFive Technology Co., Ltd., China. His research interests include applied cryptography, cryptographic protocols and cryptanalysis.

Liqun Chen (M′09) is currently a professor in the Surrey Centre for Cyber Security, University of Surrey, U.K.. Prior to this appointment, she was a principal research scientist at Hewlett-Packard Laboratories (HP Labs) in Bristol, U.K., which she joined in 1997. She has developed several cryptographic schemes adopted by the International Standards and some of them have been implemented in Trusted Platform Modules. She has an extensive publication record and holds a large number of granted patents in cryptography and information security. She has been a member of the editorial board for some journals, including the International Journal of Information Security and the Computer Journal. Moreover, she received her B.S. degree (1982), M.S. degree (1985) and Ph.D. degree (1988) in information science and engineering from Southeast University, China.

1545-5971 (c) 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.