Hindawi
Wireless Communications and Mobile Computing
Volume 2017, Article ID 6128437, 11 pages
https://doi.org/10.1155/2017/6128437

Research Article

Scalable and Soundness Verifiable Outsourcing Computation in Marine Mobile Computing

Kai Zhang,[1,2] Lifei Wei,[3] Xiangxue Li,[1,4] and Haifeng Qian[1]

[1] *Department of Computer Science and Technology, East China Normal University, Shanghai, China*
[2] *The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China*
[3] *College of Information Technology, Shanghai Ocean University, Shanghai, China*
[4] *Westone Cryptologic Research Center, Beijing, China*

Correspondence should be addressed to Haifeng Qian; hfqian@cs.ecnu.edu.cn

Received 29 June 2017; Accepted 13 September 2017; Published 10 December 2017

Academic Editor: Zhipeng Cai

Outsourcing computation with verifiability is a merging notion in cloud computing, which enables lightweight clients to outsource costly computation tasks to the cloud and efficiently check the correctness of the result in the end. This advanced notion is more important in marine mobile computing since the oceangoing vessels are usually constrained with less storage and computation resources. In such a scenario, vessels always firstly outsource data set and perform a function computing over them or at first outsource computing functions and input data set into them. However, vessels may choose which delegation computation type to outsource, which generally depends on the actual circumstances. Hence, we propose a scalable verifiable outsourcing computation protocol ($\mathscr{SV}$-$\mathscr{OC}$) in marine cloud computing at first and extract a single-mode version of it ($\mathscr{SM}$-$\mathscr{SV}$-$\mathscr{OC}$), where both protocols allow anyone who holds verification tokens to efficiently verify the computed result returned from cloud. In this way, the introduced "scalable" property lets vessels adjust the protocol to cope with different delegation situations in practice. We additionally prove both $\mathscr{SV}$-$\mathscr{OC}$ and $\mathscr{SM}$-$\mathscr{SV}$-$\mathscr{OC}$ achieving selective soundness in the random oracle model and evaluate their performance in the end.

1. Introduction

Cloud computing [1], a shared pool of massive configurable computing resources, provides resource-constrained clients with various capabilities to access computation resources in an on-demand way. The merging development of hardware (e.g., sensor, wearable-device unit) makes it possible for mobile devices [2, 3] feeling free to use and enjoy the cloud service in mobile computing category [4, 5].

This is especially important for the marine mobile computing filed since marine ecosystems should be exploited and treated seriously from both environmental side and economic side. In order to monitor the changes of marine ecosystems, scientific vessels need to perform a series of mathematical or statical analysis over collected data [6]. This includes calculating the average temperature of ocean in an instantaneous moment or during a time period and reporting the variance of the dissolved oxygen during 24 hours, 72 hours, 6 months, or more [7, 8].

However, the vessels are usually not supported by powerful data collection devices and large-scale computation processers. As a result, marine sensor units should collect marine data at first and send the collected data to vessels or base stations. Also, they may outsource some expensive computations to the cloud server and expect to use the result enjoyably after an efficient verification phase (since the cloud may return an incorrect answer for some profits). Moreover, a public verification method is preferable; namely, anyone holding the verification token can run the verification procedure in public.

Moreover, we notice that the vessel's usual outsourcing computation in marine mobile computing comes from the following two types (as in Table 1).

Type I. A client outsources a combined input tuple containing data set and function together as inputs at first and then types into an importing function over the outsourced data

Table 1: Outsourcing computation types in mobile computing.

Delegation type	Type I		Type II
Importing parts	(Function, data set)		(Data set)
	$\Downarrow$ $\Downarrow$	or	$\Downarrow$
Outsourced parts	(Data set, function)		(Function)
Remarks	Combining inputs		Single input

and an importing data set towards the outsourced function in a combined way.

Type II. A client outsources a function as an input at first and then types into an importing data set towards the outsourced function. (Here, we do not consider a delegation type where a client outsources a data set at first and takes inputs on it. A detailed analysis on this can be found in Section 5.)

Maybe, clients should flexibly switch Type I and Type II due to their actual demands in reality. If we design and deploy two respective outsourcing computation protocol systems for respective delegation type, there is no doubt that this will cause a big waste of resources, which is even not feasible in marine WSNs. Hence, a "scalable" property for an outsourcing computation protocol should be highlighted. Apart from this, some desirable features for verifiable outsourcing computation protocols in marine WSNs should also be considered seriously.

Therefore, we may have the following doubt: *whether an efficient scalable outsourcing computation protocol with public verifiability towards Type I or/and Type II delegation in marine mobile computing field exists or not?*

Our Results. To give an affirmative answer to this expectation, we manage to design a public verifiable outsourcing computation protocol for Type I outsourcing and moreover extend it to support Type II outsourcing as well, which are inspired by [9–11]. Specifically, our contributions in this work can be summarized as the following four parts:

(i) Aiming for securely performing Type I computation outsourcing, we put forward a scalable public verifiable outsourcing computation protocol in marine mobile computing, namely, $\mathscr{SV}$-$\mathscr{OC}$. This protocol allows anyone to use a granted verification token to verify the result originated from any vessel's Type I computation request.

(ii) By treating the outsourced data set as an "on-the-fly" input of $\mathscr{SV}$-$\mathscr{OC}$, we extract a single-mode version (i.e., for Type II computation) with adding a slight additional cost. As a result, vessels can just use a $\mathscr{SV}$-$\mathscr{OC}$ protocol enough for both Type I and Type II computation as they like, which shows the "scalable property's" flexibility at a maximum extent.

(iii) Both our $\mathscr{SV}$-$\mathscr{OC}$ and $\mathscr{SM}$-$\mathscr{SV}$-$\mathscr{OC}$ protocols are proven to achieve perfect correctness and selective soundness in the random oracle model. Furthermore, the efficiency analysis and concrete performance evaluations on both two protocols are provided.

(iv) We motivate an intuition that the $\mathscr{SV}$-$\mathscr{OC}$ protocol can be viewed as a hierarchical public VC protocol towards only outsourced function (Type II), where the subjective function accepts the outsourced data which can be viewed as a hierarchical access control procedure.

1.1. Problem Statement

In this subsection, we present design goals and system overview for our introduced protocols.

Design Goals. To achieve both functionalities and privacy-preserving requirements for an outsourcing computation protocol in marine mobile computing, the design goals can be thought from the following five parts.

(1) Scalability. The protocol should be able to flexibly vary its shapes depending on the type of outsourcing computation.

(2) Public Verifiability. Anyone with verification tokens can check the correctness of the result.

(3) Public Delegation. Any client can outsource a computation assignment to the cloud once the system is set up.

(4) Correctness. A dishonest cloud cannot return an incorrect output that passes verification.

(5) Soundness. A public (verifiable) outsourcing computation protocol should be secure and sound (cf. Section 2.3).

System Description. Our $\mathscr{SV}$-$\mathscr{OC}$ or $\mathscr{SM}$-$\mathscr{SV}$-$\mathscr{OC}$ protocol consists of the following three entities.

(i) Cloud Server. It receives the outsourcing computation request from any vessel and returns a result.

(ii) Vessels (consisting of a pilot one and a number of nonpilot ones). They delegate outsourcing computation tasks to the cloud and expect to receive the correct computational outcome.

(iii) Satellite. It provides a wireless communication channel between cloud server and vessels.

High-Level Roadmap. Figure 1 gives a high-level system overview on a group verifiable outsourcing computation protocol, namely, both $\mathscr{SV}$-$\mathscr{OC}$ protocol and $\mathscr{SM}$-$\mathscr{SV}$-$\mathscr{OC}$ protocol. To be specific, the cloud server provides a verifiable outsourcing computation service for group vessels through the wireless channel supplied by the satellite. Note that a pilot vessel in a group of vessels initializes the public verifiable outsourcing computation service by outsourcing the delegation computing function (and accompanied outsourced data set), as well as sending the generated public system information to the whole system and the generated evaluation key information about computing function (and accompanied data set) to the cloud. In this way, any vessel in this group can delegate computations by directly typing inputs into the computing function (and accompanied data set). Then the cloud server performs a computation for the outsourcing request from a vessel. Finally, anyone who possesses a legal verification token (granted from the delegating vessel) is able to verify the

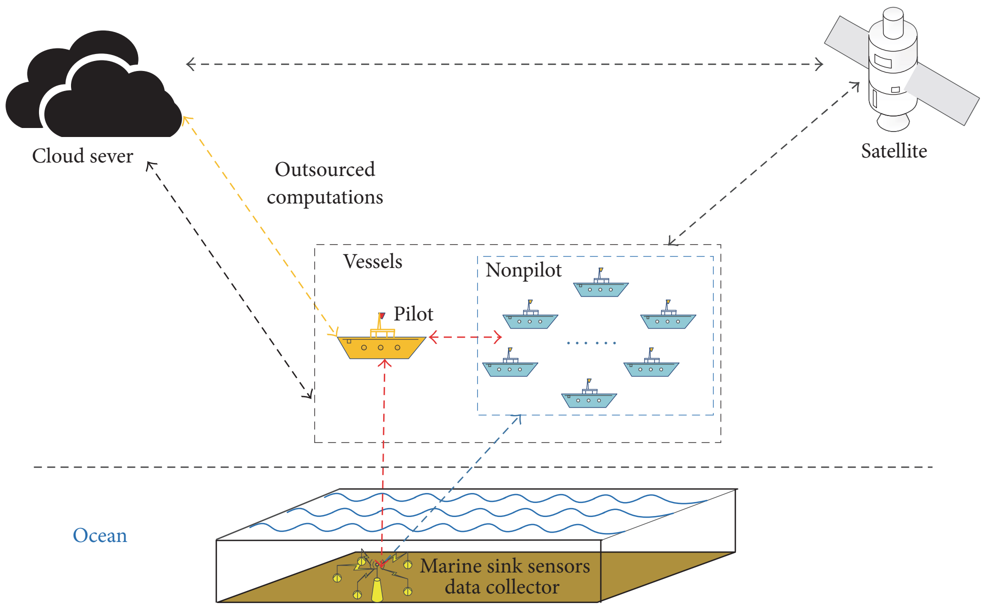

Figure 1: Scalable and soundness verifiable outsourcing computation in marine mobile computing.

result. We note that the above procedure path is highly similar to Type II (and Type I) outsourcing computation, that is, $\mathscr{SV}$-$\mathscr{OC}$ or $\mathscr{SM}$-$\mathscr{SV}$-$\mathscr{OC}$ protocol, respectively, where the only difference is the clients' outsourcing type and importing type.

1.2. Related Work. The studied problem is usually solved through a verification computation (VC) [12, 13] method, which starts with outsourcing a computing function to the cloud at first and then takes inputs on it. However, current VC protocols do not satisfy the listed design goals simultaneously in specific marine cloud computing. The other way to consider the verifiable outsourcing computation field is designed for running some verifiable delegations on outsourced data sets [14, 15], which is a little different from the formal VC concept where it differs in outsourcing whether it is a computing function or a data set at first. Also some works focused on performing computations towards outsourced functions (outsourcing at first) have been proposed [9, 13, 16–18]. For the public delegation and the public verifiable property, Applebaum et al.'s works did not satisfy them, as well as the work presented in [13, 14, 19]. Reference [11] presented $\mathscr{SV}$-$\mathscr{OC}$ protocol supported Type I computation outsourcing but neglected Type II one, so was the *hybrid* [20, 21] notion for verifiable computation failing the scalable property.

We note that all approaches to construct VC protocols except for functional encryption-based method failed to provide *public delegation* property for a verifiable outsourcing protocol towards a group of clients. From this point of view, our proposed solution is more enjoyable for such a scenario. More importantly, current works fail to achieve all the mentioned design goals simultaneously.

Organization. In Section 2, we introduce the system model and security definition for our protocol. Section 3 gives the $\mathscr{SV}$-$\mathscr{OC}$ protocol and its security analysis is provided in Section 4. An extracted version for single-mode public verifiable outsourcing computation protocol towards just outsourced function is shown in Section 5. Section 6 evaluates the performance and Section 7 gives a conclusion.

2. Background Knowledge

Notations 1. We denote by $s \overset{\$}{\leftarrow} S$ the fact that s is picked uniformly at random from a finite set S. We denote PPT as a probabilistic polynomial-time algorithm. We use $\cdot$ to denote multiplication (or group operation) as well as component-wise multiplication.

2.1. Outsourcing Functions' Description Using Access Structures

Definition 2. A (monotone) *access structure* $\mathbb{A} = (\mathbf{M} \in Z_p^{\ell \times \ell'}, \rho : [\ell] \rightarrow \mathscr{U})$ for set universe $\mathscr{U}$. One may hold the fact for an attribute set $\psi \subseteq \mathscr{U}$: $\mathbb{A}$ accepts $\psi \Leftrightarrow \mathbf{1} \in \mathsf{span}\langle \mathbf{M}_\psi \rangle$. Here, $1 = (1, 0, \ldots, 0) \in \mathbb{Z}_p^{\ell'}$ is a row vector; as $\mathbf{M}_j$ represents the jth row vector of matrix $\mathbf{M}$, a linear span $\mathsf{span}\langle \mathbf{M}_\psi \rangle$ is a collection of vectors $\mathbf{M}_\psi = \{\mathbf{M}_j : \rho(j) \in \psi\}$ over $\mathbb{Z}_p$.

Remark 3. In this paper, we mainly focus on giving a verifiable outsourcing computation protocol for Boolean formula delegating functions. When we manage to enable our protocol to be usable for multibits F rather than one bit (Boolean formula), we usually take the following steps to realize:

(1) Split the computing function F in to some subfunctions $f_1, \ldots, f_n$, where f_i is the ith output bit of the computing function F.

(2) Now we can run the $\mathscr{SV}$-$\mathscr{OC}$ and $\mathscr{SM}$-$\mathscr{SV}$-$\mathscr{OC}$ (for Boolean formula function) with conducting each subfunction f_i.

Therefore, we can obtain a scalable outsourcing computation protocol for (polynomial many) multibits output for $\mathsf{F} \in \mathscr{F}$, where $\mathscr{F}$ can be implemented by a polynomial-size Boolean formula's circuit. In this case, any outsourcing function $F \in \mathscr{F}$ can be computed by a polynomial-size Boolean formula and can thus be described by a (monotone) access structure [22]. We therefore use the access structures to symbolize the aiming outsourced (Boolean) functions $\mathscr{F}$ throughout this paper.

2.2. Underlying Security Guarantee. The security of our protocol relies on the decisional q-BDHE assumption. Let $\mathbb{G}$, $\mathbb{G}_T$ be two cyclic groups of prime order p and a generator g of group $\mathbb{G}$ along with an efficient computable map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. Randomly choose generators $g, h \overset{\$}{\leftarrow} \mathbb{G}$ and $\alpha \overset{\$}{\leftarrow} \mathbb{Z}_p$ and a tuple $\mathscr{D} = (h^{\alpha}, g^{\alpha^2}, \ldots, g^{\alpha^q}, g^{\alpha^{q+2}}, \ldots, g^{2q}, Z)$, and an adversary $\mathscr{A}$ should distinguish a computed value $e(g,h)^{\alpha^{q+1}}$ from a random element Z in $\mathbb{G}_T$. Finally, $\mathscr{A}$ outputs $b \in \{0,1\}$ having an advantage ϵ in solving the decisional q-BDHE problem if $|\Pr[\mathscr{A}(g,h,\mathscr{D},e(g^{\alpha^{q+1}},h)) = 0] - \Pr[\mathscr{A}(g,h,\mathscr{D},Z) = 0]| \geq \epsilon$.

Definition 4. One says that the decisional q-BDHE assumption holds in $(\mathbb{G}, \mathbb{G}_T)$ if, for any PPT adversary $\mathscr{A}$, its advantage in above game is negligible in security parameter λ.

2.3. Definition for Scalable Verifiable Outsourcing Computation. In this subsection, we present the system definition, correctness definition, security definition, and privacy definition for a scalable verifiable outsourcing computation protocol.

System Definition. A scalable verifiable outsourcing computation $\mathscr{SV}$-$\mathscr{OC}$ protocol is composed of the following four PPT algorithms:

(i) $\mathsf{KeyGen}(\mathsf{F}, \psi, 1^{\lambda})$: given a security parameter 1^{λ}, on input a function F and an accompanied outsourced data set ψ, the pilot vessel outputs a public key $\textsc{pk}_{\mathsf{F}}$ and an evaluation key $\textsc{ek}_{\mathsf{F}}$.

(ii) $\mathsf{ProbGen}(\textsc{pk}_{\mathsf{F}}, \omega, \mathsf{G})$: on input $\textsc{pk}_{\mathsf{F}}$, any (pilot or non-pilot) vessel can use it to encode an input ω into a problem description $\sigma_{\omega,\mathsf{G}}$, as well as outputting a verification key $\textsc{vk}_{\omega,\mathsf{G}}$.

(iii) $\mathsf{Compute}(\textsc{ek}_{\mathsf{F},\psi}, \sigma_{\omega,\mathsf{G}})$: on input $\textsc{ek}_{\mathsf{F},\psi}$ and a problem description $\sigma_{\omega,\mathsf{G}}$, the data center (cloud) computes an outcome σ_{ot}.

(iv) $\mathsf{Verify}(\textsc{vk}_{\omega,\mathsf{G}}, \sigma_{\mathrm{ot}})$: with input of the cloud's output σ_{ot}, anyone returns an output $\mathsf{ot} \in \{0,1\}^*$ or $\perp$ (rejects the cloud's answer σ_{ot} using $\textsc{vk}_{\omega,\mathsf{G}}$).

Correctness Definition. Given a security parameter λ, for any outsourced data set $\psi \in \mathscr{U}'$ and outsourced function $\mathsf{F} \in \mathscr{F}$ and any subjective function $\mathsf{G} \in \mathscr{F}'$ and for any objective data set $\omega \in \mathscr{U}$, $(\sigma_{\omega,\mathsf{G}}, \textsc{vk}_{\omega,\mathsf{G}}) \overset{\$}{\leftarrow} \mathsf{ProbGen}(\textsc{pk}_{\mathsf{F}}, \omega, \mathsf{G})$, $\sigma_{\mathrm{ot}} \overset{\$}{\leftarrow} \mathsf{Compute}(\textsc{ek}_{\mathsf{F},\psi}, \sigma_{\omega,\mathsf{G}})$, then

$$\Pr\left[\mathsf{Verify}\left(\textsc{vk}_{\omega,\mathsf{G}}, \sigma_{\mathrm{ot}}\right) = \mathsf{F}\left(\omega, \mathsf{G}\right)\right] = 1. \tag{1}$$

Security Definition. We define a security experiment against adaptive (adaptively chosen outsourced function and data sets) adversaries, which is played by a challenger and a stateful adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$.

A $\mathscr{SV}$-$\mathscr{OC}$ protocol achieves *selective soundness* if for all PPT adversaries $\mathscr{A}$ and for any $\mathsf{F} \in \mathscr{F}$ and $\psi \in \mathscr{U}'$, $\mathscr{A}$'s winning advantage

$$\begin{aligned} &\Pr\left[\mathsf{Exp}^{\mathsf{Sel\text{-}Soundness}}_{\mathscr{A},\mathscr{SV}\text{-}\mathscr{OC}}\left(\mathsf{F}, \psi, \lambda, q\right)\right] \\ &\quad = \Pr\left[\mathsf{Exp}^{\mathsf{Sel\text{-}Soundness}}_{\mathscr{A},\mathscr{SV}\text{-}\mathscr{OC}}\left(\mathsf{F}, \psi, \lambda\right) = 1\right] \end{aligned} \tag{2}$$

under the following condition,

$$\mathsf{Experiment}\ \mathsf{Exp}^{\mathsf{Sel\text{-}Soundness}}_{\mathscr{A},\mathscr{SV}\text{-}\mathscr{OC}}(\mathsf{F}, \psi, \lambda)$$

$$\begin{aligned}
&(\omega^*, \mathsf{G}^*) \overset{\$}{\leftarrow} \mathscr{A} \\
&(\textsc{pk}_{\mathsf{F}}, \textsc{ek}_{\mathsf{F}}) \leftarrow \mathsf{Setup}(1^{\lambda}, \mathsf{F}, \psi); \\
&(\omega^*, \mathsf{G}^*, \mathsf{st}) \leftarrow \mathscr{A}_1^{\mathscr{O}(\cdot)}(\textsc{pk}_{\mathsf{F}}); \\
&(\sigma_{\omega^*,\mathsf{G}^*}, \textsc{vk}_{\omega^*,\mathsf{G}^*}) \leftarrow \mathsf{ProbGen}(\textsc{pk}_{\mathsf{F}}, \omega^*, \mathsf{G}^*) \\
&(\sigma^*_{\mathrm{ot}}) \leftarrow \mathscr{A}_2^{\mathscr{O}(\cdot)}(\mathsf{st}, \sigma_{\omega^*,\mathsf{G}^*}, \textsc{vk}_{\mathsf{F},\psi}, \textsc{ek}_{\mathsf{F}}); \\
&\mathsf{ot}^* \leftarrow \mathsf{Verify}(\textsc{vk}_{\omega^*,\mathsf{G}^*}, \sigma^*_{\mathrm{ot}}); \\
&\text{If } \mathsf{ot}^* \notin \{\perp, \mathsf{F}(\omega^*)\} \text{ outputs “1”},
\end{aligned}$$

is negligible in security parameter λ, where $\mathscr{O}(\cdot)$ means that the adversary $\mathscr{A}_1/\mathscr{A}_2$ can submit q pairs $\{(\mathsf{F}_i, \psi_i)\}_{i=[q]}$ that make the experiment always output “1.”

Privacy Definition. The clients' outsourced/input computing function and data set are altogether kept hidden from the adversary's view. Moreover, the cloud's output for the problem solution does also not leak any information on the problem description. In this paper, we consider these as *outsourcing privacy*, *input privacy,* and *output privacy*.

3. Our Scalable Verifiable Outsourcing Computation Protocol: $\mathscr{SV}$-$\mathscr{OC}$

Inspired by the dual-policy attribute-based encryption (ABE) scheme [10, 23], we present the first publicly verifiable outsourcing computation protocol towards both (Boolean formula) outsourced functions and outsourced data sets altogether, which also relies on our introduced variant transformation [11] of the general relationship between ABE and public VC [9].

Specifically specifying the example in Section 1, the pilot vessel first initializes the $\mathscr{SV}$-$\mathscr{OC}$ service by inputting an outsourced function and an accompanied data set to generate a public key $\textsc{pk}_{\mathsf{F}}$ and an evaluation key $\textsc{ek}_{\mathsf{F}}$ and sends them to the cloud and other vessels. Thus, any vessel in this fleet can directly input the objective input ω for F and an accompanied computation function G over data set G along with randomly chosen messages m, m' altogether, to generate a problem description $\sigma_{\omega,\mathsf{G}}$ and a verification key $\textsc{vk}_{\omega,\mathsf{G}}$. Once receiving $\textsc{pk}_{\mathsf{F}}$ and $\textsc{ek}_{\mathsf{F}}$, the cloud computes the problem result σ_{ot} on the problem $\sigma_{\omega,\mathsf{G}}$. Finally the vessel (or a legal granted anyone) can use the verification key $\textsc{vk}_{\omega,\mathsf{G}}$ to efficiently check the result $\sigma_{\omega,\mathsf{G}}$'s correctness.

3.1. System Initialization Phase. Given an outsourced function $\mathsf{F} \in \mathscr{F}$ with input size n as inputs, define two hash functions $\mathsf{H} : \mathbb{Z}_p \to \mathbb{G}$, $\mathsf{H}' : \mathbb{Z}_p \to \mathbb{G}$. The pilot vessel randomly chooses $g, \overline{g} \xleftarrow{\$} \mathbb{G}$ and $s, \overline{s}, \alpha, \overline{\alpha} \xleftarrow{\$} \mathbb{Z}_p$. Then it generates and outputs two master public/secret key pairs of information pieces:

$$\begin{aligned} \text{MPK} &:= \left(g, e(g,g)^s, g^\alpha, \mathsf{H}, \mathsf{H}'\right), \\ \text{MSK} &:= (\gamma, \alpha); \\ \overline{\text{MPK}} &:= \left(\overline{g}, e(\overline{g},\overline{g})^{\overline{s}}, \overline{g}^{\overline{\alpha}}, \mathsf{H}, \mathsf{H}'\right), \\ \overline{\text{MSK}} &:= (\overline{\gamma}, \overline{\alpha}). \end{aligned} \quad (3)$$

3.2. Evaluation Key Generation Phase. For an encoded objective outsourced function $\mathsf{F} \in \mathscr{F}$'s access structure $\mathbb{A}' := (N \in \mathbb{Z}_p^{\ell' \times k'}, \pi : [\ell'] \to [n'])$, as well as a subjective outsourced data set $\psi \subset \mathscr{U}'$, pick a random vector $\mathbf{v} \xleftarrow{\$} \mathbb{Z}_p^{k'}$ such that $\overrightarrow{1}\mathbf{v} = \gamma + \alpha r$ for $r \xleftarrow{\$} \mathbb{Z}_p$ and set $\eta_i = N_i \cdot \mathbf{v}$, $i \in [\ell']$. Output

$$\begin{aligned} \text{SK}_{\mathsf{F},\psi} &:= \left(K, \{K_x\}_{x\in\psi}, \{K_i', K_i''\}_{i\in[\ell']}\right) \\ &= \left(g^r, \{\mathsf{H}(x)^r\}_{x\in\psi}, \{g^{\eta_i}\mathsf{H}'(\pi(i))^{-r_i}, g^{r_i}\}_{i\in[\ell']}\right). \end{aligned} \quad (4)$$

Similarly, we obtain the corresponding secret key $\text{SK}_{\overline{\mathsf{F}},\psi}$ using uniformly and randomly chosen independent "$\overline{\text{XX}}$"-type variables. (Here, we omit the descriptions on the sampling process on "$\overline{\text{XX}}$", since it is almost same as that for $\text{SK}_{\mathsf{F},\psi}$) Then,

$$\begin{aligned} \overline{\text{SK}}_{\overline{\mathsf{F}},\psi} &:= \left(\overline{K}, \{\overline{K}_{\overline{x}}\}_{\overline{x}\in\psi}, \{\overline{K}_i, K_i'\}_{i\in[\overline{\ell}']}\right) \\ &= \left(\overline{g}^{\overline{r}}, \{\mathsf{H}(\overline{x})^{\overline{r}}\}_{\overline{x}\in\psi}, \{\overline{g}^{\overline{\eta}_i}\mathsf{H}'(\overline{\pi}(i))^{-\overline{r}_i}, \overline{g}^{\overline{r}_i}\}_{i\in[\overline{\ell}']}\right), \end{aligned} \quad (5)$$

where $\overline{\mathsf{F}}$ denotes the complement function of the outsourced function F. Hence, output the public key and the evaluation key information as

$$\begin{aligned} \text{PK}_{\mathsf{F}} &:= (\text{MPK}, \overline{\text{MPK}}) \\ \text{EK}_{\mathsf{F}} &:= \left(\text{SK}_{\mathsf{F},\psi}, \overline{\text{SK}}_{\overline{\mathsf{F}},\psi}\right). \end{aligned} \quad (6)$$

3.3. Problem Generation Phase. Given an objective data set $\omega \subset \mathscr{U}$ and the access structure $\mathbb{A} := (M \in \mathbb{Z}_p^{\ell\times k}, \rho : [\ell] \to [n])$ of an encoded subjective function $\mathsf{G} \in \mathscr{G}$ altogether as inputs, randomly choose a random vector $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_p^k$ such that $\mathbf{1u} = s$ for $s \xleftarrow{\$} \mathbb{Z}_p$ and set $\lambda_i = M_i \cdot \mathbf{u}$, $i \in [\ell]$. Pick two messages $m, \overline{m}$ and output

$$\begin{aligned} \text{CT}_{\omega,\mathsf{G}} &:= \left(C, C', \{C_i\}_{i\in[\ell]}, \{C_x''\}_{x\in\omega}\right) \\ &= \left(\mathscr{M}\cdot e(g,g)^s, g^s, \{g^{\alpha\lambda_i}\mathsf{H}(\rho(i))^{-s}\}_{i\in[\ell]}, \{\mathsf{H}'(x)^s\}_{x\in\omega}\right) \end{aligned} \quad (7)$$

and similarly we generate $\overline{\text{CT}}_{\omega,\mathsf{G}}$ (by introducing new "$\overline{\text{XX}}$"-type parameters to generate $\overline{\text{CT}}_{\omega,\mathsf{G}}$ by using $\overline{\text{MBK}}$):

$$\begin{aligned} \overline{\text{CT}}_{\omega,\mathsf{G}} &:= \left(\overline{C}, \overline{C}', \{\overline{C}_i\}_{i\in[\ell]}, \{\overline{C}_x''\}_{x\in\omega}\right) \\ &= \left(\overline{m}\cdot e(\overline{g},\overline{g})^s, \overline{g}^s, \{\overline{g}^{\alpha\lambda_i}\mathsf{H}(\rho(i))^{-s}\}_{i\in[\ell]}, \{\mathsf{H}'(x)^s\}_{x\in\omega}\right). \end{aligned} \quad (8)$$

Hence, output the problem description and the verification key information as

$$\begin{aligned} \sigma_{\omega,\mathsf{G}} &:= (\text{CT}_{\omega,\mathsf{G}}, \overline{\text{CT}}_{\omega,\mathsf{G}}), \\ \text{VK}_{\omega,\mathsf{G}} &:= (H(m), H(\overline{m})), \end{aligned} \quad (9)$$

where H is a one-way function.

3.4. Compute Phase. Upon the problem description $\sigma_{\omega,\mathsf{G}}$ and the evaluation key $\text{EK}_{\mathsf{F},\psi}$, compute

$$\begin{aligned} m' &\longleftarrow C\cdot\frac{\prod_{i\in\{i|\rho(i)\in\psi\}}\left(e(C_i,K)\cdot e\left(C', K_{\rho(j)}\right)\right)^{u_i}}{\prod_{j\in\{i|\pi(i)\in\omega\}}\left(e\left(K_j', C'\right)\cdot e\left(K_j'', C_{\pi(j)}''\right)\right)^{v_j}}, \\ \overline{m}' &\longleftarrow \overline{C} \\ &\quad \cdot\frac{\prod_{i\in\{i|\rho(i)\in\psi\}}\left(e\left(\overline{C}_i,\overline{K}\right)\cdot e\left(\overline{C}', \overline{K}_{\rho(j)}\right)\right)^{u_i}}{\prod_{j\in\{i|\overline{\pi}(i)\in\omega\}}\left(e\left(\overline{K}_j', \overline{C}'\right)\cdot e\left(\overline{K}_j'', \overline{C}_{\overline{\pi}(j)}''\right)\right)^{v_j}}. \end{aligned} \quad (10)$$

Output the problem solution $\sigma_{\text{ot}} := (m', \overline{m}')$.

Here, we note that this compute process can be realized efficiently (reducing the number of pairing operations) but just add a few exponentiation operations as a tradeoff.

3.5. Verification Phase. Input $\text{VK}_{\omega,\mathsf{G}} = (H(m), H(\overline{m}))$ and $\sigma_{\text{ot}} = (m', \overline{m}')$. Output

$$\delta := \begin{cases} 0, & \text{if } H\left(m'\right) = H(m); \\ 1, & \text{if } H\left(\overline{m}'\right) = H(\overline{m}),\ H\left(m'\right) \neq H(m); \\ \perp, & \text{if } H\left(\overline{m}'\right) \neq H(\overline{m}),\ H\left(m'\right) \neq H(m). \end{cases} \quad (11)$$

Remark 5. The verifiability of $\mathscr{SV}$-$\mathscr{OC}$ is mainly against the outsourced function since the concept of the complement data sets of ψ does not make sense in practice compared to $\overline{\mathsf{F}}$. Hence, our $\mathscr{SV}$-$\mathscr{OC}$ can be served as a hierarchical public VC protocol towards just outsourced function, which regards the subjective function accepting outsourced data set as a hierarchical (fine-grained) access control condition.

4. Security Analysis

In this section, we give correctness and efficiency analysis on our $\mathscr{SV}$-$\mathscr{OC}$ protocol at first and sketch a security analysis and privacy analysis on it as well.

4.1. Correctness Analysis. Based on the correctness of [10] dual-policy attribute-based encryption along with our modified transformation [11] between ABE and public VC in terms of [9], the correctness follows straightforwardly when both the following two conditions hold: (1) the outsourced function F accepts the data set ω; (2) the outsourced data set ψ satisfies the function G.

In the compute phase, the recovery process of $\overline{m}'$ is parallel to that of $\overline{m}$. Here, we just show the correctness of the $\overline{m}$ case:

Table 2: Size analysis of our $\mathcal{SV}$-$\mathcal{OC}$ protocol. In the table, "$|\mathbb{G}|$, $|\mathbb{G}_T|$" denote the size of a group element in groups $\mathbb{G}$ and $\mathbb{G}_T$, respectively; "n, ℓ," respectively, represent the maximum input size and the row's number in the sharing matrix $\mathbf{M}$.

	Description	Sizes
pk_{F}	Public key	$4\lvert\mathbb{G}\rvert + 2\lvert\mathbb{G}_T\rvert$
ek_{F}	Evaluation key	$(4\ell + 2n + 2)\lvert\mathbb{G}\rvert$
$\sigma_{\omega,\mathsf{G}}$	Problem description	$(2\ell + 2\mathrm{n} + 2)\lvert\mathbb{G}\rvert + 2\lvert\mathbb{G}_T\rvert$
$\mathrm{vk}_{\omega,\mathsf{G}}$	Verification key	$2\lvert\mathbb{G}\rvert$

$$C \cdot \frac{\prod_{i\in\{i\mid\rho(i)\in\psi\}}\left(e\left(C_i, K\right)\cdot e\left(C', K_{\rho(j)}\right)\right)^{u_i}}{\prod_{j\in\{i\mid\pi(i)\in\omega\}}\left(e\left(K'_j, C'\right)\cdot e\left(K''_j, C''_{\pi(j)}\right)\right)^{v_j}} = C \cdot \frac{\prod_{i\in\{i\mid\rho(i)\in\psi\}}\left(e\left(g^{\alpha\lambda_i}\mathsf{H}\left(\rho\left(i\right)\right)^{-s}, g^{r}\right)\cdot e\left(g^{s}, \mathsf{H}\left(\rho\left(i\right)\right)^{r}\right)\right)^{u_i}}{\prod_{j\in\{i\mid\pi(i)\in\omega\}}\left(e\left(g^{\eta_i}\mathsf{H}'\left(\pi\left(j\right)\right)^{-r_j}, g^{s}\right)\cdot e\left(g^{r_j}, \mathsf{H}'\left(\pi\left(j\right)\right)^{s}\right)\right)^{v_j}} = C \cdot \frac{\prod_{i\in\{i\mid\rho(i)\in\psi\}}e\left(g^{\alpha\lambda_i}, g^{r}\right)^{u_i}}{\prod_{j\in\{i\mid\pi(i)\in\omega\}}e\left(g^{\eta_j}, g^{s}\right)^{v_j}} = C \cdot \frac{e\left(g^{\alpha s}, g^{r}\right)}{e\left(g^{\gamma+\alpha r}\right), g^{s}} = C \cdot \frac{1}{e\left(g, g\right)^{\gamma s}} = m, \tag{12}$$

where the fourth equation follows the linear reconstruction property of Definition 2, and we have

$$\sum_{i\in\{i\mid\rho(i)\in\psi\}} u_i\lambda_i = s, \qquad \sum_{j\in\{i\mid\pi(i)\in\omega\}} v_i\eta_i = \gamma + \alpha r. \tag{13}$$

Remark 6. The correctness of the above compute phase is similar to that of the decryption process in [10].

4.2. Efficiency Analysis. In this part, we give a time and a size efficiency analysis for $\mathcal{SV}$-$\mathcal{OC}$. Concretely, Table 3 lists the dominant time operations (i.e., pairing, exponentiation, and multiplication) in group that belongs to each step of $\mathcal{SV}$-$\mathcal{OC}$, and moreover Table 2 gives the size calculations.

Remark 7. The compute phase's overhead can be optimized up to $(2n + 4)\mathsf{Pairing} + 8n\mathsf{Exp}_{\mathbb{G}_T}$.

In the $\mathcal{SV}$-$\mathcal{OC}$ protocol, Step (1) and Step (2) are altogether done by the pilot vessel, any vessel can perform Step (3), and the data center (e.g., cloud) completes Step (4) along with the fact that anyone can carry out Step (5).

As the bandwidth between each entity across this marine WSNs is low [5, 24], the low parameter size is highly demanded. From Table 3, we find that most operations that need high cost reside in the data center side. Consequently, the pilot vessel can certainly afford the VC service initialization computation overhead. In this way, the overhead of the problem description paid by any vessel is short, and anyone's verification cost on the result is very little as well. Therefore, the efficiency of the obtained $\mathcal{SV}$-$\mathcal{OC}$ is enjoyably applicable to the marine wireless sensor networks.

4.3. Security Analysis

Theorem 8 (main theorem). *Let $\mathcal{F}$ be a class of Boolean functions (implemented by a family of circuits $\mathcal{C}$), and let $\overline{\mathcal{F}} = \{\overline{\mathsf{F}} \mid \mathsf{F} \in \mathcal{F}\}$ be a class of the complement function $\overline{\mathsf{F}}$ of each function F and the class of the outsourced data set $\mathcal{U}' = \{\psi \mid \psi \in \mathcal{U}'\}$ and H be any one-way function. Suppose Definition 4 holds; then the $\mathcal{SV}$-$\mathcal{OC}$ protocol in Section 3 achieves selective soundness property according to the security definition in Section 2.3.*

We can easily reduce the security of $\mathcal{SV}$-$\mathcal{OC}$ with adaptive soundness to the adaptive security of the dual-policy ABE [10] and the general transformation between them, since one can obtain the $\mathcal{SV}$-$\mathcal{OC}$ protocol by running the ABE scheme twice along with other techniques. More technical details can be found in Section 4.2 of [10] and Appendix A of [9].

4.4. Privacy Analysis. During the $\mathcal{SV}$-$\mathcal{OC}$ protocol's process carried out, the specific contents of the outsourced part and the input part are encoded as another form. Specifically, the clients' outsourced computing function and accompanied data set are encoded as an evaluation key EK_{F} and any client's input ω, and G is encoded as a problem generation $\mathrm{CT}_{\omega,\mathsf{G}}$, in such a way that the cloud cannot obtain any knowledge about the *outsourcing privacy* and *input privacy*. For the output privacy, the random message m is also hidden by a owe-way function H; thus the cloud can just get $H(m)$ and is unable to recover m from it (except a negligible advantage) which is considered to achieve *output privacy* as well.

5. Extracted Single-Mode Version of $\mathcal{SV}$-$\mathcal{OC}$ Protocol

In some cases, the clients (e.g., vessels) may just outsource either data sets or computing function to the cloud; therefore we have to ask the following question:

> *Can we transform the dual-mode verifiable outsourcing computation into a single-mode one?*

Intuitively, setting one of the outsourced data sets and outsourced function as "on-the-fly" input of $\mathcal{SV}$-$\mathcal{OC}$ protocol, we hence assume obtaining two single-mode public VC protocols towards respective outsourced function and outsourced data sets. However, this assumption fails due to the nonexistence of a single-mode $\mathcal{SV}$-$\mathcal{OC}$ for outsourcing data sets. The reasons are as follows:

(1) Firstly, we should observe that the complement class of the outsourced data sets ψ does not make any sense in practice, which is not similar to the relation between $\overline{\mathsf{F}}$ and F. It is also not easy to obtain the complement class of ψ.

(2) Secondly, one can run the key-policy ABE (KP-ABE) mode of dual-policy ABE (DP-ABE) in [10] twice for respective F and $\overline{\mathsf{F}}$, but the relation between ciphertext-policy ABE and public VC is not known so far. In this way, the checkability of the single-mode $\mathcal{SV}$-$\mathcal{OC}$ over outsourcing data sets cannot achieve "1."

Hence, we can just obtain the single-mode variant of $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ for outsourced computing function at first, namely, Type II delegation type.

TABLE 3: Group operations analysis in each phase of our $\mathcal{SV}$-$\mathcal{OC}$ protocol. In the table, "Pairing" represents a paring operation in the protocol; $\mathsf{Exp}_{\mathbb{G}}$ and $\mathsf{Exp}_{\mathbb{G}_T}$ denote an exponentiation operation in groups $\mathbb{G}$ and $\mathbb{G}_T$, respectively; similarly, $\mathsf{Mul}_{\mathbb{G}}$ and $\mathsf{Mul}_{\mathbb{G}_T}$ denote a multiplication operation in group $\mathbb{G}$ and $\mathbb{G}_T$, respectively; "n, ℓ," respectively, represent the maximum input size and the row's number in the sharing matrix **M**.

Step	Description	Performer	Operations
(1)	System initialization	Pilot vessel	$2\mathsf{Pairing} + 2\mathsf{Exp}_{\mathbb{G}_T} + 2\mathsf{Exp}_{\mathbb{G}}$
(2)	Evaluation key generation	Pilot vessel	$(6\ell + 2n)\mathsf{Exp}_{\mathbb{G}} + (2n + 2\ell)\mathsf{H} + 2\ell\mathsf{Mul}_{\mathbb{G}}$
(3)	Problem generation	Any vessel	$(4\ell + 2)\mathsf{Exp}_{\mathbb{G}} + (2\ell + 2n + 2)\mathsf{H} + 2\ell\mathsf{Mul}_{\mathbb{G}}$
(4)	Compute	Cloud	$8n\mathsf{Pairing} + 2n\mathsf{Exp}_{\mathbb{G}_T}$
(5)	Verification	Anyone	$2\mathsf{H}$

5.1. Construction for Single-Mode $\mathcal{SV}$-$\mathcal{OC}$ for Just Outsourcing Functions: $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$.

Inspired by the KP-ABE mode of dual-policy ABE [10] and our $\mathcal{SV}$-$\mathcal{OC}$ protocol, we give the *single-mode publicly verifiable outsourcing computation towards outsourcing computing functions*' construction.

(1) System Initialization Phase. This step is same as that of $\mathcal{SV}$-$\mathcal{OC}$ except for adding special data T_0 as a new input.

(2) Evaluation Key Generation Phase. This stage is same as that of $\mathcal{SV}$-$\mathcal{OC}$ except by randomly choosing $r_0, \overline{r}_0 \xleftarrow{\$} \mathbb{Z}_p$ and setting

$$\begin{aligned} K_0' &= g^{\gamma+\alpha r}\mathsf{H}(T_0)^{-r_0}, \\ K_0'' &= g^{r_0}, \\ \overline{K}_0' &= \overline{g}^{\overline{\gamma}+\overline{\alpha r}}\mathsf{H}(T_0)^{-\overline{r}_0}, \\ \overline{K}_0'' &= \overline{g}^{\overline{r}_0}. \end{aligned} \tag{14}$$

Hence the evaluation key behaves as

$$\begin{aligned} \mathrm{EK}_{\mathsf{F}} &:= \left(\mathrm{SK}_{\mathsf{F},\psi}, \overline{\mathrm{SK}}_{\overline{\mathsf{F}},\psi}\right) = \Big(K, \{K_x\}_{x\in\psi}, \{K_i', K_i''\}_{i\in[\ell']}, \\ &K_0', K_0'', \overline{K}, \{\overline{K}_{\overline{x}}\}_{\overline{x}\in\psi}, \{\overline{K}_i, \overline{K}_i'\}_{i\in[\overline{\ell}']}, \overline{K}_0', \overline{K}_0''\Big). \end{aligned} \tag{15}$$

(3) Problem Generation Phase. This is almost same as that of $\mathcal{SV}$-$\mathcal{OC}$ except for sampling $s \xleftarrow{\$} \mathbb{Z}_p$ and setting

$$\begin{aligned} C &= m \cdot e(g,g)^s, \\ C' &= g^{\alpha s}, \\ C_0 &= g^s, \\ \{C_x''\}_{x\in\omega} &= \{\mathsf{H}(x)^s\}_{x\in\omega}, \\ \overline{C} &= \overline{m} \cdot e(\overline{g},\overline{g})^s, \\ \overline{C}' &= \overline{g}^{\alpha s}, \\ \overline{C}_0 &= \overline{g}^s, \\ \{\overline{C}_x''\}_{x\in\omega} &= \{\mathsf{H}(x)^s\}_{x\in\omega}. \end{aligned} \tag{16}$$

Hence, the problem description behaves as

$$\begin{aligned} \sigma_{\omega,\mathsf{G}} &:= \left(\mathrm{CT}_{\omega,\mathsf{G}}, \overline{\mathrm{CT}}_{\omega,\mathsf{G}}\right) \\ &= \left(C, C', C_0, \{C_x''\}_{x\in\omega}, \overline{C}, \overline{C}', \overline{C}_0, \{\overline{C}_x''\}_{x\in\omega}\right). \end{aligned} \tag{17}$$

TABLE 4: Size analysis of our single-mode $\mathcal{SV}$-$\mathcal{OC}$ protocol. In the table, "$|\mathbb{G}|$, $|\mathbb{G}_T|$" denote the size of a group element in groups $\mathbb{G}$ and $\mathbb{G}_T$, respectively; "n, ℓ," respectively, represent the maximum input size and the row's number in the sharing matrix **M**.

	Description	Sizes
pk_{F}	Public key	$4\lvert\mathbb{G}\rvert + 2\lvert\mathbb{G}_T\rvert$
ek_{F}	Evaluation key	$(4\ell + 2n + 6)\lvert\mathbb{G}\rvert$
$\sigma_{\omega,\mathsf{G}}$	Problem description	$(2n + 4)\lvert\mathbb{G}\rvert + 2\lvert\mathbb{G}_T\rvert$
$\mathrm{vk}_{\omega,\mathsf{G}}$	Verification key	$2\lvert\mathbb{G}\rvert$

(4) Compute Phase. In this case, this process computes as follows:

$$\begin{aligned} m' &\longleftarrow \frac{C \cdot e(C_0, K)}{\prod_{j\in\{i\mid\pi(i)\in\omega\}}\left(e\left(K_j', C'\right)\cdot e\left(K_j'', C_{\pi(j)}''\right)\right)^{\nu_j}}, \\ \overline{m}' &\longleftarrow \frac{C \cdot e\left(\overline{C}_0, \overline{K}\right)}{\prod_{j\in\{i\mid\overline{\pi}(i)\in\omega\}}\left(e\left(\overline{K}_j', \overline{C}'\right)\cdot e\left(\overline{K}_j'', \overline{C}_{\overline{\pi}(j)}''\right)\right)^{\nu_j}}. \end{aligned} \tag{18}$$

Finally, output the problem solution $\sigma_{\mathrm{ot}} := (m', \overline{m}')$.

(5) Verification Phase. This step is exactly same as that of $\mathcal{SV}$-$\mathcal{OC}$.

This concludes the construction description.

5.2. Analysis on the Single-Mode $\mathcal{SV}$-$\mathcal{OC}$ for Just Outsourcing Computing Functions.

In this subsection, we still give a correctness, efficiency, and security analysis on the $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ protocol.

5.2.1. Correctness Analysis.

The correctness holds when F accepts the data sets ω, where the secret shares' reconstruction follows

$$\sum_{j\in\{i\mid\pi(i)\in\omega\}} \nu_i\eta_i = \gamma + \alpha r. \quad \text{(Definition 2)}. \tag{19}$$

5.2.2. Efficiency Analysis.

In general, the size and time efficiency of the single-mode $\mathcal{SV}$-$\mathcal{OC}$ protocol for only outsourcing computing functions are comparable to those of $\mathcal{SV}$-$\mathcal{OC}$ one. Next, we present the time and size efficiency analysis for $\mathcal{SV}$-$\mathcal{OC}$ in concrete way; Table 4 gives the size

Table 5: Group operations analysis in each phase of our single-mode $\mathcal{SV}$-$\mathcal{OC}$ protocol. In the table, "Pairing" represents a paring operation in the protocol; $\mathsf{Exp}_{\mathbb{G}}$ and $\mathsf{Exp}_{\mathbb{G}_T}$ denote an exponentiation operation in group $\mathbb{G}$ or $\mathbb{G}_T$, respectively; similarly, $\mathsf{Mul}_{\mathbb{G}}$ and $\mathsf{Mul}_{\mathbb{G}_T}$ denote a multiplication operation in groups $\mathbb{G}$ and $\mathbb{G}_T$, respectively; "n, ℓ," respectively, represent the maximum input size and the row's number in the sharing matrix $\mathbf{M}$.

Step	Description	Performer	Operations
(1)	System initialization	Pilot vessel	$2\mathsf{Pairing} + 2\mathsf{Exp}_{\mathbb{G}_T} + 2\mathsf{Exp}_{\mathbb{G}}$
(2)	Evaluation key generation	Pilot vessel	$(6\ell + 2n + 6)\mathsf{Exp}_{\mathbb{G}} + (2n + 2\ell + 2)\mathsf{H} + (2\ell + 2)\mathsf{Mul}_{\mathbb{G}}$
(3)	Problem generation	Any vessel	$(2n + 4)\mathsf{Exp}_{\mathbb{G}} + (2n)\mathsf{H} + 2\mathsf{Mul}_{\mathbb{G}}$
(4)	Compute	Cloud	$4n\mathsf{Pairing} + (2n + 2)\mathsf{Exp}_{\mathbb{G}_T} + 4\mathsf{Mul}_{\mathbb{G}_T}$
(5)	Verification	Anyone	$2\mathsf{H}$

calculations and moreover Table 5 lists the dominant time operations (i.e., pairing, exponentiation, and multiplication) in group which performed in each step of single-mode $\mathcal{SV}$-$\mathcal{OC}$.

In concrete way, the problem generation and verification overheads enjoy better efficiency than that in $\mathcal{SV}$-$\mathcal{OC}$, but its overhead on generating evaluation key is a little expensive (including the size of EK_{F}) compared to $\mathcal{SV}$-$\mathcal{OC}$, since "on-the-fly" data set is involved to handle the construction. A tradeoff between Steps (1), (2), and (3) and Steps (4) and (5) over the above three steps does inevitably exist. Apart from this, the overall time and time overhead are almost same as that of $\mathcal{SV}$-$\mathcal{OC}$.

As a result, the (non)pilot vessel or anyone can efficiently run the single-mode $\mathcal{SV}$-$\mathcal{OC}$ service, and moreover the cloud's running cost on computing the problem also turns out to be short. In this way, we can directly extract a highly efficient $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ protocol from $\mathcal{SV}$-$\mathcal{OC}$.

5.2.3. Security Analysis

Theorem 9 (main theorem). *Let $\mathcal{F}$ be a class of Boolean functions (implemented by a family of circuits $\mathcal{C}$), and let $\overline{\mathcal{F}} = \{\overline{\mathsf{F}} \mid \mathsf{F} \in \mathcal{F}\}$ be a class of the complement function $\overline{\mathsf{F}}$ of each function F and H be any one-way function. Suppose Definition 4 holds; then the single-mode $\mathcal{SV}$-$\mathcal{OC}$ protocol for only outsourcing computing functions achieves selective soundness according to the security definition in Section 2.3.*

The proposed single-mode verifiable outsourcing computation protocol $\mathcal{SV}$-$\mathcal{OC}$ can be seen as a special variant of $\mathcal{SV}$-$\mathcal{OC}$ in fact, whereas their functionalities are merely not the same. Based on the security analysis on Theorem 8, Theorem 9 can be proved easily as well.

5.2.4. Privacy Analysis. The privacy analysis on the $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ protocol is same as that of the $\mathcal{SV}$-$\mathcal{OC}$ protocol in Section 4.4.

6. Performance Evaluation

In this section, we give a performance evaluation on our $\mathcal{SV}$-$\mathcal{OC}$ and its extracted single-mode outsourcing computation protocol $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$. Applying a certain implementation technique on realizing bilinear maps, we choose using an asymmetric bilinear group $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ to implement the symmetric bilinear group $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ for $\mathcal{SV}$-$\mathcal{OC}$ and $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ in the actual experiment as in [25].

Table 6: Element lengths of "SS512" elliptic curve.

	$\mathbb{G}_1$	$\mathbb{G}_2$	$\mathbb{G}_T$
Element length	512 bits	512 bits	1024 bits

Standing by the standard NIST recommendation [26] and general remarks [25, 27] based on the Python language's realizations along with its provided Charm-crypto Benchmark, we note that the charm tool [25] is an extensible Python-based framework under Pairing-Based Cryptography (PBC) library for rapidly prototyping cryptographic schemes and protocols, which is widely used in conducting functional encryption-based primitives. We remark that this is instantiated in an Ubuntu 12.04 operating system with 1 GB RAM (established in a MACBOOK Air Intel i5@1.8 GHz and 4 GB RAM equipped with a VMWare software). Next, we decide to employ the "SS512" elliptic curve for our performance evaluation. Finally, Table 7 shows the "SS512" curve's element length; and moreover Table 6 gives a list of the "SS512" curve's average running-times for each protocol step.

Suppose that the size of the data set ψ, ω is n and the value of ℓ, ℓ' is 10. Based on the employed "SS512" elliptic curve [28], the actual size evaluation in Figure 2 and the time efficiency simulation in Figure 3 are both given. In addition, we use "+" to denote the *dual-mode* $\mathcal{SV}$-$\mathcal{OC}$ protocol and "$\diamond$" to denote the *extracted single-model* $\mathcal{SV}$-$\mathcal{OC}$: $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ protocol in both Figures 2 and 3.

From Figures 2 and 3, we can deduce the fact that both $\mathcal{SV}$-$\mathcal{OC}$ and $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ achieve high space and time efficiency. Our $\mathcal{SV}$-$\mathcal{OC}$ protocol's efficiency is comparable to the extracted $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ one's efficiency. Particularly, the overload that belongs to the weak clients' sides is actual satisfactory.

Table 7: Average running-time of "SS512" elliptic curve. In the table, the symbol "ms" denotes running-time in millisecond.

	$\mathsf{Mul}_{\mathbb{G}_1}$	$\mathsf{Exp}_{\mathbb{G}_1}$	$\mathsf{Mul}_{\mathbb{G}_2}$	$\mathsf{Exp}_{\mathbb{G}_2}$	$\mathsf{Mul}_{\mathbb{G}_T}$	$\mathsf{Exp}_{\mathbb{G}_T}$	$\mathsf{Pairing}$
Running-time	0.024 ms	3.7503 ms	0.0201 ms	3.7833 ms	0.0055 ms	0.4844 ms	3.9723 ms

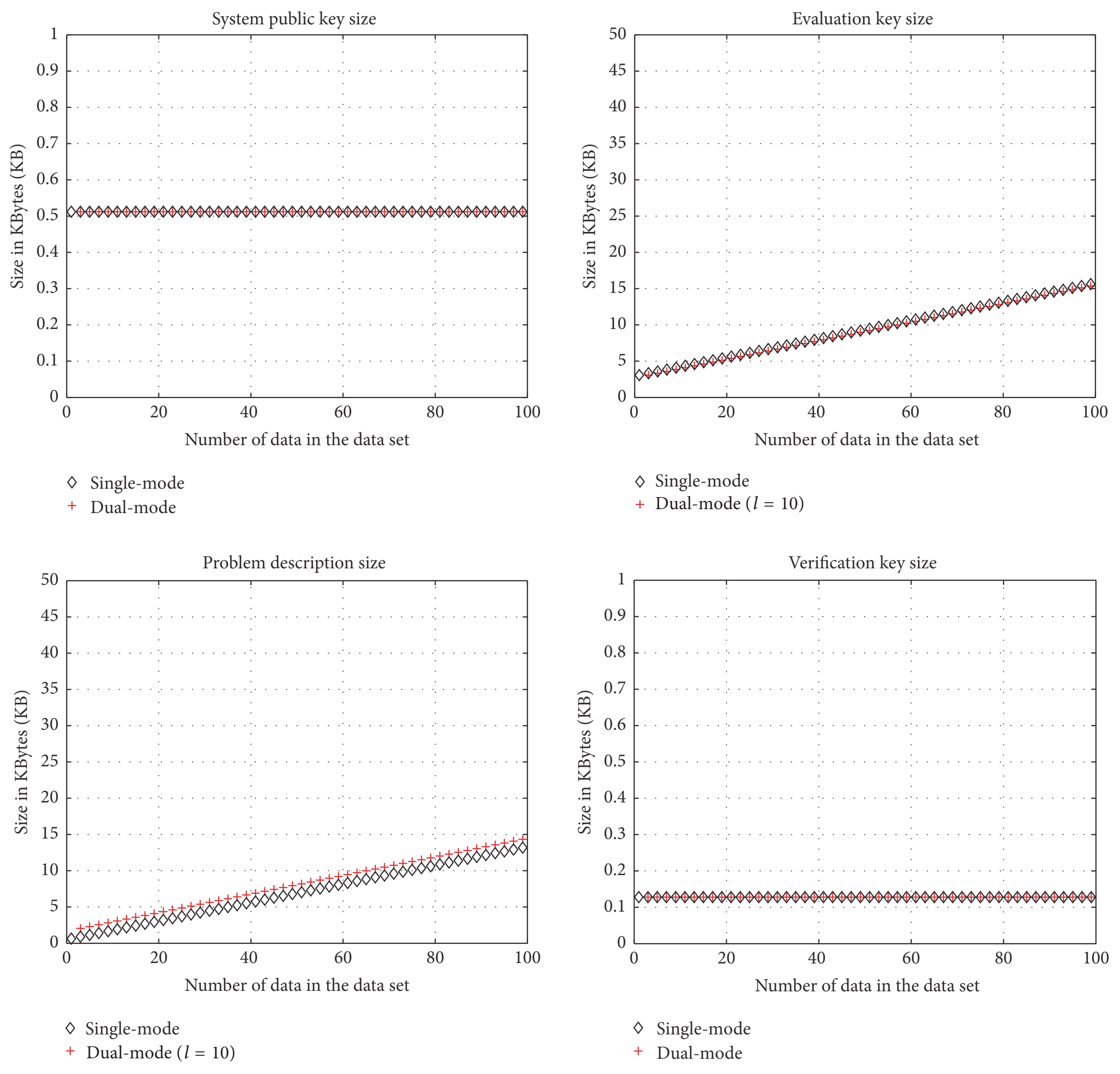

Figure 2: Size efficiency of the $\mathcal{SV}$-$\mathcal{OC}$ protocol.

7. Concluding Remarks

This paper presented a scalable and soundness verifiable outsourcing computation protocol in marine mobile cloud computing. Our $\mathcal{SV}$-$\mathcal{OC}$ protocol enabled any client to delegate a computation task to the server and was also able to designate anyone to verify the result. In addition, an extracted single-mode outsourcing computation protocol $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ from $\mathcal{SV}$-$\mathcal{OC}$ was presented, which led to a fact that the client can adapt $\mathcal{SV}$-$\mathcal{OC}$ based on the inputs' option in terms of its interest or its own needs. However, we found that our $\mathcal{SM}$-$\mathcal{SV}$-$\mathcal{OC}$ protocol could just handle the outsourcing function as the single mode; hence a design of a verifiable outsourcing computation protocol towards outsourced function may be an open problem.

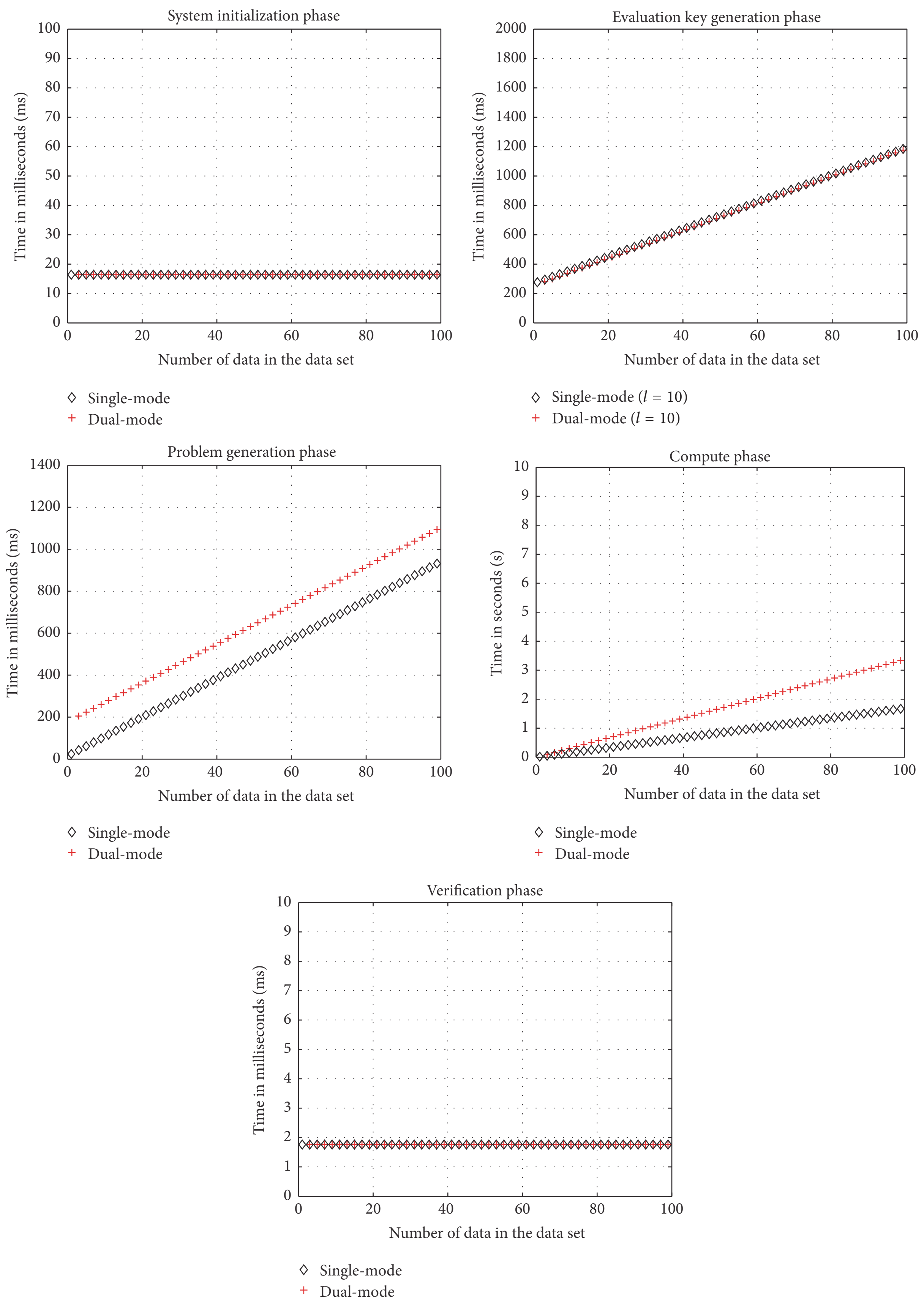

Figure 3: Time efficiency of the $\mathcal{SV}$-$\mathcal{OC}$ protocol.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors want to acknowledge the WASA 2017 anonymous reviewers' suggestions. This work was supported by the National Natural Science Foundation of China (61571191, 61572192, 61472249, 61472142, and 61402282), the "Dawn" Program of Shanghai Education Commission (no. 16SG21), and the Open Foundation of State Key Laboratory of Integrated Services Networks (ISN17-11).

References

[1] P. M. Mell and T. Grance, *Draft nist working definition of cloud computing*, vol. 15, 2009.

[2] C. D'Este, P. de Souza, C. Sharman, and S. Allen, "Relocatable, automated cost-benefit analysis for marine sensor network design," *Sensors*, vol. 12, no. 3, pp. 2874–2898, 2012.

[3] L. Yu, H. Shen, K. Sapra, L. Ye, and Z. Cai, "CoRE: Cooperative End-to-End Traffic Redundancy Elimination for Reducing Cloud Bandwidth Cost," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 446–461, 2017.

[4] D. Huang, "Mobile cloud computing," in *Proceedings of the IEEE COMSOC Multimedia Communications Technical Committee (MMTC) E-Letter*, vol. 6, pp. 27–31, 2011.

[5] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang, and Y. Pan, "Stochastic Load Balancing for Virtual Resource Management in Datacenters," *IEEE Transactions on Cloud Computing*, 2016.

[6] S. Zhang, J. Yu, A. Zhang, L. Yang, and Y. Shu, "Marine vehicle sensor network architecture and protocol designs for ocean observation," *Sensors*, vol. 12, no. 1, pp. 373–390, 2012.

[7] P. Hu, K. Xing, X. Cheng, H. Wei, and H. Zhu, "Information leaks out: Attacks and countermeasures on compressive data gathering in wireless sensor networks," in *Proceedings of the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014*, pp. 1258–1266, Ontario, Canada, May 2014.

[8] D. Huang, D. Zhao, L. Wei, Z. Wang, and Y. Du, "Modeling and analysis in marine big data: Advances and challenges," *Mathematical Problems in Engineering*, vol. 2015, Article ID 384742, 2015.

[9] B. Parno, M. Raykova, and V. Vaikuntanathan, "How to delegate and verify in public: verifiable computation from attribute-based encryption," in *Theory of Cryptography*, vol. 7194 of *Lecture Notes in Computer Science*, pp. 422–439, Springer, Berlin, Germany, 2012.

[10] N. Attrapadung and H. Imai, "Dual-policy attribute based encryption," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, vol. 5536, pp. 168–185, 2009.

[11] K. Zhang, L. Wei, X. Li, and H. Qian, "Provably secure dual-mode publicly verifiable computation protocol in marine wireless sensor networks," in *Proceedings of the 12th International Conference on Wireless Algorithms, Systems, and Applications, WASA 2017*, vol. 10251, pp. 210–219, Springer International Publishing, Guilin, China, 2017.

[12] B. Applebaum, Y. Ishai, and E. Kushilevitz, "From secrecy to soundness: Efficient verification via secure computation," in *Proceedings of the International colloquium on automata, languages and programming (ICALP)*, vol. 6198, pp. 152–163, 2010.

[13] R. Gennaro, C. Gentry, and B. Parno, "Non-interactive verifiable computing: outsourcing computation to untrusted workers," in *Advances in Cryptology-CRYPTO 2010*, vol. 6223, pp. 465–482, Springer, Berlin, Germany, 2010.

[14] S. Benabbas, R. Gennaro, and Y. Vahlis, "Verifiable delegation of computation over large datasets," in *Advances in Cryptology-CRYPTO 2011*, vol. 6841, pp. 111–131, 2011.

[15] M. Backes, D. Fiore, and R. M. Reischuk, "Verifiable delegation of computation on outsourced data," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS 2013*, pp. 863–874, Berlin, Germany, November 2013.

[16] D. Fiore and R. Gennaro, "Publicly verifiable delegation of large polynomials and matrix computations, with applications," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 501–512, ACM, October 2012.

[17] K. Zhang, J. Gong, S. Tang et al., "Practical and efficient attribute-based encryption with constant-size ciphertexts in outsourced verifiable computation," in *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security, ASIA CCS 2016*, pp. 269–279, Xi'an, China, June 2016.

[18] Y. Sun, Y. Yu, X. Li, K. Zhang, H. Qian, and Y. Zhou, "Batch verifiable computation with public verifiability for outsourcing polynomials and matrix computations," in *Proceedings of the Part I of Information Security and Privacy - 21st Australasian Conference, ACISP 2016*, vol. 9722, pp. 293–309, Springer International Publishing, Melbourne, VIC, Australia, 2016.

[19] K. Chung, Y. T. Kalai, and S. P. Vadhan, "Improved delegation of computation using fully homomorphic encryption," in *in Proceedings of CRYPTO, 2010*, pp. 483–501, Springer, Santa Barbara, CA, USA, 2010.

[20] J. Alderman, C. Janson, C. Cid, and J. Crampton, "Access control in Publicly Verifiable Outsourced Computation," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security, ASIACCS 2015*, pp. 657–662, Singapore, April 2015.

[21] J. Alderman, C. Janson, C. Cid, and J. Crampton, "Hybrid Publicly Verifiable Computation," in *Topics in Cryptology - CT-RSA 2016*, vol. 9610 of *Lecture Notes in Computer Science*, pp. 147–163, Springer International Publishing, Cham, 2016.

[22] A. Beimel, *Secure Schemes for Secret Sharing And Key Distribution [Ph.D. Thesis]*, Technion-Israel Institute of technology, Faculty of computer science, 1996.

[23] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06)*, pp. 89–98, November 2006.

[24] L. Yu and Z. Cai, "Dynamic scaling of virtual clusters with bandwidth guarantee in cloud datacenters," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, IEEE INFOCOM 2016*, pp. 1–9, San Francisco, CA, USA, April 2016.

[25] J. A. Akinyele, C. Garman, I. Miers et al., "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[26] D. Giry, Bluekrypt, https://www.keylength.com/en/.

[27] F. Zhang, Tech. Rep., http://student.seas.gwu.edu/~zfwise/crypto.

[28] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 84, pp. 1234–1243, 2001.