Journal of Information Security and Applications manuscript No.
(will be inserted by the editor)

A cryptographic cloud-based approach for the mitigation of the airline cargo cancellation problem

Gabriele Gianini · Stelvio Cimato · Maryam Sepehri · Rasool Asal · Ernesto Damiani

Received: DD Month YEAR / Accepted: DD Month YEAR

Abstract In order to keep in good long-term relationships with their main customers, Airline Cargo companies do not impose any fee for last minute cancellations of shipments. As a result, customers can book the same shipment on several cargo companies. Cargo companies try to balance cancellations by a corresponding volume of overbooking. However, the considerable uncertainty in the number of cancellations does not allow to fine-tune the optimal overbooking level, causing losses. In this work, we show how the deployment of cryptographic techniques, enabling the computation on private information of customers and companies data can improve the overall service chain, allowing for striking and enforcing better agreements. We propose a query system based on proxy re-encryption and show how the relevant information can be extracted, still preserving the privacy of customers' data. Furthermore, we provide a Game Theoretic model of the use case scenario and show that it allows a more accurate estimate of the cancellation rates. This supports the reduction of the uncertainty and allows to better tune the overbooking level.

Keywords Airline Cargo · Overbooking and cancellation · Proxy re-encryption · Inspection Games

G. Gianini (Corresponding Author)
EBTIC - Khalifa University of Science and Technology, PO Box 127788, Abu Dhabi, UAE and Computer Science Department, Universita degli Studi di Milano, via Bramante 65, Crema (CR), 26013, Italy – gabriele.gianini@unimi.it
S. Cimato
Computer Science Department, Universita degli Studi di Milano, via Bramante 65, Crema (CR), 26013, Italy – stelvio.cimato@unimi.it
M. Sepehri
Computer Science Department, Universita degli Studi di Milano, via Celoria 18, Milano (MI), 20133, Italy – maryam.sepehri@unimi.it
R. Asal
EBTIC - Khalifa University of Science and Technology, PO Box 127788, Abu Dhabi, UAE – rasool.asa@bt.com
E. Damiani
EBTIC - Khalifa University of Science and Technology, PO Box 127788, Abu Dhabi, UAE and Computer Science Department, Universita degli Studi di Milano, via Bramante 65, Crema (CR), 26013, Italy – ernesto.damiani@unimi.it

1 Introduction

One of the main problems in air cargo revenue management is the modeling of overbooking and cancellation in the operation of the service chain, which involves air-cargo carrier companies (ACCs) and their clients, the freight forwarder companies (FFs). Together with shippers, they are the main players in the air-cargo chain. ACCs operate flights on which cargo loads are transported; FFs buy in advance bulk cargo capacity and sell it to individual shippers, consolidate smaller shipment into larger units and deliver them to ACCs by agreed dates (only few shippers are direct customers of the ACCs [9]).

There are two mechanisms used by ACCs for selling capacity: *pre-allocation* sale and *ad-hoc* sale. In terms of financial market terminology, the pre-allocation sale corresponds to a long term *forward* contract between the FF and the ACC, where the former commits to buy in the future the agreed capacity (specific volume or weight on a specific flight and date). The ad-hoc sale corresponds to a *spot-market* sale, without prior commitments. The original purpose of the long term forward contracts is to grant to the ACC some paid capacity and compensate the FFs for their loss of flexibility by means of price breaks. Before the start of each season (a predefined time interval, typically of the order of few months) ACCs allocate cargo capacity to FFs on that season's flights. Capacity sold to a FF in this manner is called an *allotment* or *allotted capacity* [24]. FFs typically book cargo weight, or volume, on a specific flight several weeks in advance. However, for several reasons, the cargo load might fail to be delivered to the ACC by the scheduled date (this event is called *no-show*), or might undergo severe *volume or weight reduction*: those two contingencies cause a waste of capacity to the ACC (called *spoilage*). Globally the problem of spoilage reduction is referred to as the cargo airline *cancellation problem.*

Typically, ACCs try to balance this problem by overbooking the flight (i.e. in selling more capacity than the one is actually admissible on the flight); however an excess of overbooking w.r.t. the actual amount of cancellation can result in some loads not being carried (this problem is called *off-loading*), with consequent economic losses (contractual penalties, storage or re-routing charges). The problem of balancing cancellation and overbooking, and of reducing the associated risk, is studied in revenue management, and dealt with using a wide spectrum of approaches (see for instance [24], [22] and references therein).

Trying to reduce the cancellation rate is, naturally, at the core of most approaches. Some ACCs introduced a cancellation fee on those air cargo bookings that are cancelled within three days from departure, so as to deter cancellations; however, due to the forces acting within this specific market, this countermeasure is *difficult to enforce* with every customer: in order to keep in good long-term relationships with their main customers - typically large FFs - most ACCs do not impose fines for last minute cancellations. In practice, large FFs pay only for the capacity they actually use [5].

As a result, a large FF can: (1) book a departure time where the cargo might not be ready, or book an optimistically large volume, which is unlikely

to be filled (we refer to this behavior as *excess reservation*) or, even, (2) book on several cargo companies the transportation of the same cargo (we refer to this behaviour as *multiple reservation*).

Excess reservation is mainly motivated by the opportunity to exploit the possible *upward* demand fluctuations on a volatile shipment market; other less openly acknowledged motivations for this behavior by freight forwarders is the purpose to block out competitors [1]. Excess reservation is thus one of the main causes of weight/volume reduction.

Multiple reservation, instead, is a way for exploiting the *downward* price fluctuations on the cargo spot-market and it implies at least two bookings by a FF for the same cargo load: a booking with a first ACC through a long-term *forward* contract and a booking with a second ACC through a short-term contract on the spot market [27]. Multiple reservation is one of the causes of no-show. Notice, that we assume that a load can be uniquely identified by source, destination, dates, weight and volume and its other features declared at booking time: this information forms the *descriptor* of the load. We posit that even if on the spot market a smaller weight or volume is booked for the cargo by a FF, the occurrence of a multiple reservation can be assessed.

Excess reservation and multiple reservation have thus, very different characters. However they have one element in common: hidden information plays an important role in the motivations. We will develop this point further. Before doing this, it is important to mention – as noticed, for instance, by Hellermann [22, 23] – that the right of a forwarder to cancel without penalty, is equivalent to "having signed a *forward contract*, but holding in fact a *call option* on the allotted capacity": this option gives the right, but not the obligation, to buy the allotted capacity. The approach proposed by Hellerman is to take that contract for what actually is: to formalize it as an *option contract* and to give it a suitable pricing, computed using *option theory*.

The approach followed in the present paper is different, in that we propose to leverage part of the hidden information by means of privacy preserving computing techniques, so as to remove some inefficiencies of the market, and move market players toward a different equilibrium.

1.1 Leveraging hidden information

Our point, indeed, is that part of the information unknown to the parties is just hidden information, i.e. information available to individual parties, but that cannot be disclosed, for *market confidentiality* reasons. If at least part of that information could be elicited, without compromising confidentiality, to obtain publicly sharable information, the fairness of the process could be considerably improved.

In other words, market forces, exploiting the principle of confidentiality are currently producing an unfair share of the risk, between FFs and ACC, even though the parties are working within the same supply chain. Letting some hidden information surface, could allow the invocation of strong principles such

as rightfulness, and make the share of the risk among the parties less unfair. Typically, if an indicator of the non-justifiability/rightfulness/legitimacy of a given behavior could be made available (without violating confidentiality) most dysfunctional behaviors could be assigned a penalty by an *enforceable* contract, thus discouraging that behavior.

With respect to *excess reservation*, the hidden information concerns the actual capacity demand by the suppliers: this information can be forecast, based on private information known to FFs and not to ACCs: making available this information in aggregated form to the ACCs could help the latter to tune the overbooking. We return to this point in the Discussion and Conclusions section: our focus here is on the multiple reservation problem.

With respect to *multiple reservation*, at shipment time, the hidden information, known by the FF, but not by the ACC, consists in whether the FF has actually sent the load through another ACC. This information, though, is present in the airlines company cargo records. An ACC could impose, by contract, to a FF that it will not book the same cargo over more ACCs (in exchange, the ACC could offer incentives in the form of moderate discounts in case the spot price falls below some threshold). This is a condition that a FF could agree to accept, even if it restricts its operational freedom: indeed, it is unlikely that a FF defends the right to no-shows motivated by the use of alternative ACC for the very same cargo. To support the enforcement of such condition one can set up a *privacy-preserving search engine system*. In case of no show, the system can be queried, to check whether the event results from using an alternative ACC: if that is the case, the FF incurs a penalty. The adoption of such solution is beneficial for cargo companies: protection against this type of dysfunctional behavior is mutual interest of all the airline cargo companies, even though they are competitors.

Such privacy preserving query system can be supported either by a trusted party managed ledger, or by a real time query computation mechanism over a distributed dataset. In general, techniques based on secure multi-party computation enable different parties to perform distributed computation on secret inputs: following this paradigm, it is possible to compute any public function and share the output among the parties, while preserving the privacy of the inputs: after the execution of the protocol each party does not learn anything more than the computed values.

In this paper we propose a privacy preserving query system that protects users' data, still allowing the detection of misbehavior from one of the participants. Synergies between ACCs and their customer FFs and synergies among ACCs motivate the adoption of the above described solutions: the cost of such audit system could be shared among the participants.

Hereafter we develop the design of the system to contrast the problem of multiple reservations: we plan to discuss in a future work the problem of excess reservation and the corresponding solution. Thus, the main contributions of the present work is the description of an audit system for *multiple reservation detection* based on cryptographic techniques.

1.2 Game Theoretic Modeling

Obviously, such an audit system has a cost, not only for its construction and deployment, but also for its operation. It is well known that some SMC queries can be rather expensive and time consuming. Some important elements to take into account are the following: in the business scenario described, the burden of the proof is upon the ACC, i.e. the ACC has to pay for the audit, so as to prove that the cancellation is illegitimate, in order to apply a fine; furthermore, cancellations happen rather frequently and most of the time they do not correspond to multiple booking. Consequently it is impractical and can be economically disadvantageous for the ACC to run a audit at each cancellation: the ACCs can afford, instead, the adoption of a random sampling schema (randomness is used to grant non-predictability). Thus, not all the violations will be detected. This fact is know to the FFs, which can count on some level of impunity, depending on the audit rate of by the ACC. In turn, the ACC is aware of this possibility for the FFs and might try to tune the audit rate consequently.

Such an interdependent decision landscape – where the system consists of selfish players with non-aligned interests – can be effectively modeled by using Game Theory (GT). The problem of building selfish-resilient collaborative systems is often approached using Game Theory [10, 11, 12, 14, 16], also in relation to the control of private information release in Supply Chains and the associated risk [2, 3, 6, 8, 13, 15, 20].

By means of GT, under suitable hypotheses about the rationality of the players, one is able to *predict*, at least statistically, the players' behaviour in specified circumstances: such joint behavior (called Nash equilibrium) consists in a collection of strategies (one for each player) from which no player has incentives to deviate unilaterally (since this would not increase its personal payoff). The above outlined misbehaviour/auditing scenario, can be mapped to a specific class of GT models: Inspection Games (IGs). In this type of games an *inspector* controls the correct behaviour of an *inspectee*, and applies a fine if a misbehaviour is detected during the inspection. GT modeling allows to find the rate of violation and the rate of inspection at equilibrium, as a function of the parameters of the problem.

At their core, the multiple reservation by the FF and the audit by the ACC can be modeled as a (non-coalitional) two-player inspection game. By using such a model, we show that FFs and ACCs, if acting rationally, would adopt, respectively, a specific rate of violation and a specific rate of inspection: the two rates depend on the parameters of the problem. Those parameters are: cost of individual inspections, quantitative damage inflicted by a violation, benefit to the violator and value of the fine. The computation of the rate of violation by the FFs allows the ACCs to reduce the relative uncertainty in the estimate of the overall cancellation rate, thus improving the estimate of the necessary overbooking rate to be used as a countermeasure. Notice, in passing, that in this case, the players' rationality is a sound assumption: whereas individuals,

forced to take decisions under condition of uncertainty, often act irrationally, profit oriented organizations tend in general to act rationally.

The paper is organized as follows: in Section 2 the scenario, the system solution and the corresponding protocol are formally defined; in Section 3 the Game Theoretic Model of the use case is developed and the equilibrium solution is given; there, we also point out how the system can reduce the relative error cancellation rate estimate; a brief discussion of the future work, in Section 4 concludes the paper.

2 The multiple reservation detection model

We model the multiple reservation detection as a query returning a boolean value, which represents the presence of a given *descriptor* in a database. This can be modeled as querying the database resulting from the union of the databases owned by each participant. Note that those databases hold sensitive data that cannot be shared in public, since the disclosure could affect the business of the involved parties (competitors could take advantage of the information offering lower prices and increasing their market shares).

The solution we propose is based on *proxy re-encryption* in a *cloud-based scenario*. In the next subsection we give an introduction to this cryptographic technique.

2.1 Cloud-based Proxy Re-encryption Schemes

In the last years, the provision of a secure and efficient data-sharing system on the cloud has been challenging several researchers, who want to complement the reliability and availability of cloud-based storage systems, with the privacy requirements that must be satisfied when the shared data contain sensible information [32]. One of the possible solutions to the data sharing problem comes from the deployment of proxy re-encryption (PRE) schemes, where a semi-trusted proxy holding a re-encryption key translates a message encrypted under a public key into the encryption of the same message under a different public key. In this setting, firstly introduced by Blaze et al. in 1988 [7], the proxy is not able to learn anything about the encrypted message. The cloud provider can in some cases act as the proxy agent that runs the re-encryption algorithm to translate the cipher-texts of the sender to the cipher-texts encrypted using the public key of the receiver, so that the receiver can read the data using his own decryption key. The security of the underlying PREs provides the guarantee that anyone else (including the cloud) cannot access private data. In literature several examples of PREs have been provided to securely share data on public clouds [25, 31]. Proxy re-encryption has been used also to construct keyword search technique [17] where users can re-encrypt an encrypted message using different keys held by the other participants to the scheme. The scheme provided by Dong *et al.* generates a trapdoor for the user

keyword that is used by the proxy server to find a match in the encrypted data. More recently, Sepehri *et al.* [28, 29] addressed the problem of privacy-preserving equality queries over horizontally partitioned data among multiple owners adopting a proxy re-encryption scheme. They experimentally implemented the key translation process, and computed the time needed to bring data encrypted with different keys under the same key, utilizing El-Gamal encryption system. In this work we re-adapt this scheme to build a cloud-based solution for the air cargo cancellation problem.

2.2 Problem Definition

We consider the *air cargo service chain* scenario, where the freight forwarders (FFs) can book cargo weight or volume on a specific flight from air carrier companies (ACCs). We recall that usually, freight forwarders have long term contracts with ACCs, which for this reason do not impose any fee for last minute cancellations of forwarders' shipments. This benefit for FFs may lead to two main events: *no-show* event, when the cargo is not delivered to the airline by the scheduled date by a FF (cancellation); *reduction* event, in which a strong reduction of volume or weight is operated by a FF. Both those contingencies cause a waste of capacity to the ACCs. In this study we focus on the former event, and in particular on the case in which no-show events occur: for instance, given two ACCs, 1 and 2, some FFs may move their shipments from 1 to 2 at the last minute (for the sake of simplicity we call this behavior *contractual violation* or simply *violation*), e.g. because 2 has lower shipping costs. In this case, we propose a scheme to detect violations from FFs.

Let $ACC = \{ACC_1, \ldots, ACC_m\}$ denote a group of air carrier companies with $m > 1$. Each carrier ACC_i has a table T_i from a collection of horizontally partitioned data $T = \{T_1, \ldots, T_m\}$ for $1 \leq i \leq m$. For the sake of conciseness and clarity, we suppose each T_i contains one searchable attribute $T_{i,A} = Cargo\ Id$ and w extra attributes $T_{i,B1,\ldots,B_w}$; say $T_{i,B_1} = Origin$, $T_{i,B_2} = Destination$, $T_{i,B_3} = Volume/Weight$, and $T_{i,B_4} = Flight\ Id$, respectively.

Given a query $v = CargoId$ of a *no-show* cargo, the output of this equality test is a set of all tuples with extra attributes $T_{i,B1,\ldots,B_4}$ whose searchable attribute value is equal to the v. If the result of the query is not empty, a *violation event* has occurred.

2.3 System Model

Here, we consider a cloud based search system with multiple data owners as shown in Figure 1. There are five types of entities in our system:

1. *Data owners* (e.g., *ACCs*), each *ACC* locally encrypts its data with its private key and uploads it to the proxy server,

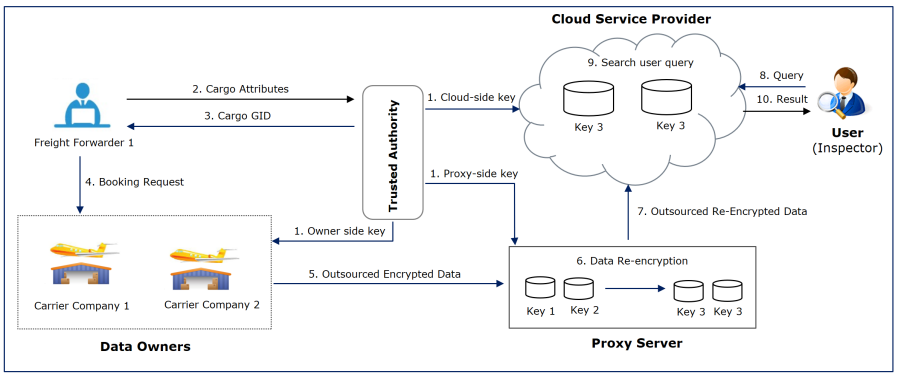

Fig. 1 Overall view of our system model with 2 air cargo carrier and 1 freight forwarder

2. *Trusted Authority (TA)*, a fully trusted server that is responsible for generating random keys for data owners and authorized users. It also assigns a global identifier GId to each cargo booking requested by a certain FF,
3. *Proxy server*, an honest-but-curious server that converts owners' record encrypted with different keys to the ones under the same key,
4. *Authorized users*, an ACC inspector as an authorized user submits query over relations stored on the cloud server,
5. *Cloud service provider*, an honest-but-curious server who stores data contributed by the owners and executes search queries

In the next section, we will propose a fast equality test for multi-owner search problem adopting multi owner equality test queries [30] while satisfying data confidentiality and query privacy

- *Data Confidentiality*: users learn the information authorized them to learn but nothing else, and the cloud server does not learn about the owners' data.

- *Query Privacy*: queried value is not disclosed neither to the cloud server nor to the data owners.

2.4 The Protocol Description

In this section, we provide a complete description of our proxy re-encryption scheme underlying ElGamal cryptosystem [19]. The proposed scheme consists in several phases, described hereafter:

Setup. On input a security parameter λ, a randomized algorithm is run by the Trusted Authority TA to output system public parameters and master key.

– **Setup** (1^{λ}): TA takes as input a security parameter λ and picks two prime numbers p, q with $p - 1 = 2q$. It generates a cyclic group $\mathbb{G}$ with generator g such that $\mathbb{G}$ is the unique order q subgroup of $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$, and then picks a random key K_M uniformly from $\mathbb{Z}_q^*$ to outputs the system master key MK and the corresponding system public parameters $\mathsf{Param} = \langle \mathbb{G}, g, q \rangle$.

Global Identity Generation (GId). TA takes as input a set of extra attributes for each cargo requested by a certain FF and outputs a global identifier (GId) that could be $Cargo\ Id$ for which a cargo has provable credential. Two cargoes with the same extra attributes must receive the same $GIds$.

Key Generation. On input the master key, the TA first runs a randomize algorithm to pick random keys for the data owners and the users and correspondingly computes keys for the proxy and the cloud server.

– **KeyGen** (MK, i, j): For each data owner i and user j, the TA does the following:

1. Generates a random value r_c and distributes it to the data owners and the authorized users.

2. For each ACC_i, the TA generates uniformly a random key $k_i \xleftarrow{R} \mathbb{Z}_q$ and computes its corresponding proxy side key $k_i^{'} \leftarrow \mathsf{MK} - k_i$. It then securely distributes k_i and $(i, k_i^{'})$ to the ACC_i and proxy server, respectively.

3. For each user j, the TA generates uniformly a random key $k_j \xleftarrow{R} \mathbb{Z}_q$ and divides it into two shares k_{j_1} and k_{j_2} such that $k_j \leftarrow k_{j_1} + k_{j_2}$. The TA computes the user's cloud side key $k_j^{'} \leftarrow \mathsf{MK} - k_j$ and securely returns keys (j, k_{j_1}), k_{j_2} and $(j, k_j^{'})$ to the proxy, the user and the cloud service provider, respectively.

Data Encryption. On input a booking request from a FF including a searchable attribute $GId = CargoId$, each ACC locally encrypts the values of searchable attribute using ElGamal encryption and applies a symmetric encryption on the values of extra attributes.

– **Enc** (k_i, T_i): For each tuple $t \in T_i$, the ACC_i does the following:

1. Picks a random r_t and encrypts the value of serachable attribute of each tuple $t \in T_i$ namely $t.A$ using ElGaml encryption to output $C_0 = (g^{r_t}, g^{r_t k_i} g^{t.A})$.

2. Creates a metadata consisting of two encrypted values using ElGamal to obtain

$$C_1 = (g^{r_t}, g^{r_t k_i} g^{r_t}) \qquad C_2 = (g^{r_c}, g^{r_c k_i} g^{r_t})$$

3. Sets $C(t.A) = (C_0, C_1, C_2)$

4. Picks a random $\bar{k_i}$ and encrypts the value of extra attribute l of tuple $t \in T_i$ say $t.B_l$, $1 \leq l \leq w$ to get $C(t.B_l) = f(t.B_l)$

5. Encrypts $\bar{k_i}$ as $I_i = (g^{r_c}, g^{r_c k_i} \bar{k_i})$

6. Outsources to the proxy server $C(T_i) =< C(t.A), C(t.B_l), I_i >$, $1 \leq l \leq w$.

Data Re-Encryption. On input the encrypted data received from ACC_i, the proxy re-encrypts data using ACC_i's proxy side key $k_i^{'}$. The data re-encryption brings all data encrypted with different keys under the same key.

– **Re-Enc** $(i, k_i^{'}, C(T_i))$: For each tuple of $C(T_i)$, the proxy server does the following:

1. Finds the proxy side key $k_i^{'}$ of ACC_i

2. Re-encrypts each component of $C(t.A)$ with proxy side key $k_i^{'}$ using ElGamal encryption to obtain

$$C_0^* = (g^{r_t}, (g^{r_t})^{k_i^{'}} \cdot g^{r_t k_i} g^{t.A}) = (g^{r_t}, g^{r_t \mathsf{MK}} g^{t.A})$$
$$C_1^* = (g^{r_t}, (g^{r_t})^{k_i^{'}} \cdot g^{r_t k_i} g^{r_t}) = (g^{r_t}, g^{r_t \mathsf{MK}} g^{r_t})$$
$$C_2^* = (g^{r_c}, (g^{r_c})^{k_i^{'}} \cdot g^{r_c k_i} g^{r_t}) = (g^{r_c}, g^{r_c \mathsf{MK}} g^{r_t})$$

3. Re-encrypts I_i with proxy side key as

$$I_i^{'} = (g^{r_c}, (g^{r_c})^{k_i^{'}} \cdot g^{r_c k_i} \bar{k_i}) = (g^{r_c}, g^{r_c \mathsf{MK}} \bar{k_i})$$

4. Set $C^*(t.A) = (C_0^*, C_1^*, C_2^*)$
5. Keeps $I_i^{'}$ and outsources to the cloud server $C^*(T_i) =< C^*(t.A), C(t.B_l) >$, $1 \leq l \leq w$

Query Search. An ACC inspector with key k_{j_2} submits its query q_v with $v = Cargo\ Id$ in an encrypted form using ElGamal encryption as

$$q_v = (g^{-r_c}, g^{-r_c k_{j_2}} g^{-v})$$

– **Q-Search** $(j, k_j^{'}, C^*(T_i), q_v)$: On input the user query q_v, the cloud server does the following:

1. Sends user query to the proxy server who re-encrypts the query with user's proxy side key k_{j_1} to get

$$q_v^{'} = (g^{-r_c}, (g^{-r_c})^{k_{j_1}} \cdot g^{-r_c k_{j_2}} g^{-v})$$
$$q_v^{'} = (g^{-r_c}, (g^{-r_c})^{\mathsf{MK}-k_j^{'}} g^{-v})$$

2. Re-encrypts $q_v^{'}$ with the user's cloud side key to output

$$q_v^{*} = (g^{-r_c}, (g^{-r_c})^{k_j^{'}} \cdot (g^{-r_c})^{\mathsf{MK}-k_j^{'}} g^{-v})$$
$$q_v^{*} = (g^{-r_c}, (g^{-r_c})^{\mathsf{MK}} g^{-v})$$

3. Upon receiving each $C^*(T_i)$ from the proxy server, the cloud service provider finds equality match with the user query value using *multiplicative homomorphic encryption* property of ElGamal cryptosystem:

   – Multiplies C_0^* by C_2^* to get $R = (g^{r_t+r_c}, g^{r_t+r_c\mathsf{MK}} g^{r_t+t.A-v})$

   – Multiplies R by q_v^* to get $R^* = (g^{r_t}, g^{r_t\mathsf{MK}} g^{r_t+t.A-v})$

   – Compares R^* with C_1^* and the match is found if and only if $t.A = v$

4. If the match results set of T_i is not empty, the cloud doses the following:
   – Sends a request to the proxy server who partially decrypts I_i to get
$$I_i^{''} = (g^{r_c}, (g^{r_c})^{-k_{j_1}} \cdot g^{r_c\mathsf{MK}} \bar{k_i})$$
   – Pre-encrypts $I_i^{''}$ with user's cloud side key to obtain
$$I_i^{*} = ((g^{r_c})^{-k_j^{'}} \cdot (g^{r_c})^{\mathsf{MK}-k_{j_1}} \bar{k_i}) = (g^{r_c}, g^{r_c k_{j_2}} \bar{k_i})$$
   – Sends to user side all extra attributes $\{C(t.B_l),\ 1 \leq l \leq w\}$ of $C^*(T_i)$ related to each tuple in the match results set along with I_i^*

Data Decryption. An *ACC* inspector fully decrypts the received I^* with its own key k_{j_2} to recover the key $\bar{k}$ corresponding to extra attributes as $\bar{k} = (g^{r_c}, (g^{r_c})^{-k_{j_2}} \cdot g^{r_c k_{j_2}} \bar{k_i})$

3 Game Theoretic model: Inspection Games

As anticipated in the introduction, the operation of the query system described in the previous section is far from being costless: the system implies a series of economic costs, not only for construction, deployment, maintenance and ordinary information update, but also per query computation. We focus on

the cost per query, and assume that the cost of a query is incurred by the querying agent, in our case an ACC.

There is a wide literature on the cost of queries in cryptographic distributed systems (indeed one of the main assessment metrics for cryptographic protocols is efficiency) however, the analysis of such costs is out of the scope of the present work: here it is important to know that they consist both in communication and computation costs and that in some cases the cost of a query is considerable. We assume that the expected cost of a query can be estimated with reasonable accuracy and refer to such a cost by c. For the sake of simplicity, we also assume that such cost is essentially the same for every query.

The point is the following: if $c > 0$, then, depending on the rate at which a cancellation corresponds to a multiple reservation, it may or may not be economically advantageous for the ACC to adopt an exhaustive audit strategy. We develop this point further below.

3.1 Definitions and assumptions

For this purpose, let us recall that we use the term *contract agreement violation* or simply *violation* to indicate a cancellation that results from a multiple reservation, i.e. from cheating. We call *non-violation* a cancellation resulting from other causes (we do not enter in to the detail of the legitimacy of those other causes, since we are interested only in detecting multiple reservations). For brevity, here, the contractual agreement that excludes multiple reservations will be called *the rule*. Let us indicate by p the rate at which a cancellation operated by the FF correspond to a violation of the rule, and by q the rate at which the ACC runs a query, given a cancellation.

For the sake of simplicity, we can put aside elements that are inessential for the reasoning, such as the fact that every cancellation corresponds to lots of different sizes (in weight and volume) and thus has a different economical value: we assume that every time a FF cancels, it saves an amount b and it brings a damage d to the ACC (those assumptions can be lifted subsequently with a minor increase in our model complexity). Ideally d (for *damage* received) represents the pre-agreed *forward price* of the capacity corresponding to the cancelled load – for which the ACC will not receive compensation, if the multiple reservation is not proved. On the other hand, b (for *benefit* received) represents the difference between that forward price and the "spot market" price for that capacity: b is the saving that the FF obtains through cheating.

We assume that, if the violation by the FF is discovered, the FF has to pay to the ACC a compensation at least equal to the forward price of the capacity. This represents a penalty to FF. This amount is specified in the contract. We indicate this amount by a (for *amends*, in the sense of penalty/fine).

Let us note, in passing, that $b < d \leq a$, this fact however (as the amount by which a is greater that d) are inessential for the following discussion: as we will see, the parameters a and b alone determine the behavior of the ACC (and

$b < a$ is granted by definition), while the parameters c and d alone determine the behavior of the FF.

3.2 Interactive decision landscape

From the point of view of the ACC, if the expected return from a query (which depends on the violation rate) is higher than the damage received, then, performing a query on every cancellation, is economically convenient. If $c < dp$ (and if p is fixed), the strategy adopted by a rational ACC would be deterministic: it would consist in auditing always, i.e. to choose $q = 1$. Should this be the case, there would be no interest from the FF in violating the rule: certainty in the detection and in the consequent reparation would discourage any attempt and would suppress the multiple reservation phenomenon.

In practical cases, however, the cost c of a query is high enough and the order of magnitude of the violation rate p is low enough for dp being less than c. This is mainly due to the fact that cancellations can happen for many reasons, most of them legitimate, many of them related to the intrinsic inefficiency of a complex system such as the air cargo service supply chain. Since $c > dp$, auditing all the cancellations would not represent a paying strategy for the ACC. Thus, the deterministic strategy does not to apply.

The ACC has to resort to some form of random-sampling based auditing: it should audit with probability $0 < q < 1$: its problem becomes choosing the optimal q. A rational FF, knowing this, would have room for violating sometimes the rule and could do so at random (again for granting non-predictability). The number of rule breaking cancellations, compared to the total number of cancellations, would determine the rate p: the problem of the FF consists in choosing its moves so that the value of p is optimal in some sense. Again for the sake of simplicity, we assume that the spot market offers enough opportunities to the FF to let her set the rate p with no restrictions.

Clearly, the choice of p by FF and the choice of q by ACC influence not only each actor's own payoff, but also the other actor's payoff. This interdependent decision landscape can be modeled by Game Theory, so as to find the behavior that the agents would adopt: under the assumption of full rationality of the players, such solution has predictive value. The form of reasoning of rational agents that can be applied to the present model is the one studied by John Nash in the context of strategic, non-coalitional games. The solution put forward by Nash [26] (and later called Nash Equilibrium) stipulates that rational agents would adopt a strategy profile (a strategy – here a choice – for each player), such that no player could improve its expected payoff by deviating from that choice unilaterally.

3.3 Nash Equilibrium of the two-player Inspection Game

Being, in our case, the game based on randomization, the solution consists in a suitable mix of the two choices by each actor (FF chooses between violating

or not, ACC choses between auditing or not). It can be shown that, for mixed strategies, the Nash equilibrium always exists and is unique [26]: the equilibrium strategy profile corresponds to the joint choice of the pair (p, q) (p chosen by FF and q chosen by ACC), such that the other player is not encouraged in modifying the mix unilaterally. This means that one or the other strategy does not bring improvement to the actor. This is equivalent to say that the right choice of rate by one player is the one that makes the other player *indifferent* between its own two choices: FF should choose p so as to make ACC indifferent between auditing or not, ACC should choose q so as to make FF indifferent between violating or not.

As mentioned in the Introduction, this is a characteristic trait of a class of randomization games known as Inspection Games, whose original formulation was introduced by Dresher in 1962 [18] in the context of arm proliferation control (for an account see [4], for a generalization to several inspectee and several interdependent inspectors see [21]). The case under discussion corresponds to an Inspection Game in strategic form (each player takes its decision about the rate without knowing the decision of the other player).

Furthermore, it is a two player game – despite the fact that there are several FFs and several ACCs – because each violation by a FF affects only the contract with a specific ACC and damages that ACC only: in the whole ecosystem of FFs and ACCs, many parallel and unrelated two-player games can be played concurrently. It is true that the choice of violating the forward contract with one ACC allows another ACC (we call here third party ACC) to sell its spare capacity on the spot market, however in this case the third party is not a player, in the sense that it has no choice between strategies (urthermore not necessarily this third party would receive a benefit from the violation: if the market is so active that it would absorb its capacity anyway, the third party is indifferent to the choice by the FF).

In the present two-player case, the equilibrium (p_*, q_*) pair can be found by solving algebraically a simple linear system [21]. The solution is

$$p_* = \frac{c}{d} \qquad\qquad q_* = \frac{b}{a} \tag{1}$$

Notice that, by construction, q_*, which represents the behavioral choice of the inspector, is determined by the quantities defining the payoffs of the inspectee, whereas p_*, which represents the behavioral choice of the inspectee, is determined by the quantities defining the payoffs of the inspector.

This solution holds under the hypothesis that the players know the parameters of the game (i.e. that this is a game of complete information). This assumption can be made confidently: the damage d and the amends a are known by both parties by contract; b is known by FF and can be discovered by ACC using public spot market information; c is known by ACC and can be learned by FF with good accuracy consulting domain experts.

3.4 Reduction of the Uncertainty of the variance rate

The most relevant quantity yield by the above Game Theoretic discussion is the predicted rate of violation p_* (computed based on the costs of inspection c and of the the damage d received by the ACC). Knowing this rate, one can reduce the relative error in the overall estimate of the cancellation rate. This happens because it establishes a constraint between two otherwise unlinked quantities, thus reducing the degrees of freedom of the problem, which simplifies the estimate: the quantities now linked are the *number of cancellations* due to multiple reservations and the *total number of cancellations* due to other causes.

Indeed, more specifically, a revenue manager would normally try to estimate *independently* the part of the cancellation rate due to multiple reservation and the part of the cancellation rate due to other causes, on the base of the fact that the two classes of phenomena are originated by distinct mechanisms.

The total cancellation rate is defined as

$$z = \frac{X+Y}{R} = x + y$$

where R is a known constant representing the total number of reservations to an ACC form a FF, X is the total number of cancellations due to multiple reservations, while Y is the total number of cancellations due to other causes, whereas $x = X/R$ and $y = Y/R$. In practice x and y are not known.

The revenue manager, normally tries to find an estimate $\hat{x}$ of x and an estimate $\hat{y}$ of y: the two estimates will be affected by uncertainties, expressed by the variances $\sigma^2(\hat{x})$ and $\sigma^2(\hat{y})$, so that the variance of the overall estimate $\hat{z}$ of the cancellation rate, $\sigma^2(\hat{z}) = \sigma^2(\hat{x}) + \sigma^2(\hat{y}) - cov(x, y)$, under the *hypothesis of independence*, will be $\sigma^2(\hat{z}) = \sigma^2(\hat{x}) + \sigma^2(\hat{y})$. The relative error is defined as

$$\frac{\sigma(\hat{z})}{z} = \frac{\sqrt{\sigma^2(\hat{x}) + \sigma^2(\hat{y})}}{x+y}$$

If, as we did using GT, we find that the quantity X is tied to the quantity Y by a fixed ratio p_*,

$$p_* = \frac{X}{X+Y} \quad \text{i.e.} \quad X = \frac{p_*}{1-p_*} Y \ , \quad \text{or} \quad x = \frac{p_*}{1-p_*} y = ry \ ,$$

with $r = p_*/(1-p_*)$, then the overall estimate reduces to the estimate of y:

$$z = y\,(1+r)$$

and the relative error on the estimate reduces to the only relative error on y

$$\frac{\sigma(\hat{z})}{z} = \frac{\sigma(\hat{y})}{y}$$

This represents a considerable improvement in the estimate, which allows the ACC to fine-tune the overbooking rate, thus saving economical resources.

4 Discussion and Conclusions

In this work we have addressed the issue of the air cargo cancellation due to *multiple reservation* by proposing the use of a query system based on a privacy preserving cryptographic technique.

The audit method can be used within a randomized inspection schema, which modeled by Game Theory, allows to predict the optimal rate of inspection and of cancellation, respectively.

We show that the prediction of the rate of cancellation due to multiple reservations reduces the uncertainty on the overall cancellation rate and allows the revenue management of Cargo companies to better tune the overbooking level.

In the future, we plan to develop further this work by a more detailed specification of the system based on realistic data from the application domain; furthermore, we plan to refine the Game Theoretic model – for the prediction of the cancellation rate originated by multiple reservation – by lifting several simplifying assumptions adopted in the present paper.

Finally, we plan to extend the approach also to leverage the private information within the collaborative forecasting of demand, to deter excess reservation. In the *excess reservation* problem, the globally unknown information concerns the actual capacity demand by the suppliers: this information can be however forecast, from information known to freight forwarders, but not to carriers (this hidden information consists in the filling of pre-orders and orders by the suppliers to the freight forwarder and on the information, through order tracking, about actual the shipment evolution). *Without* this kind of data the cargo carrier can only rely on historical no-show record of the forwarder, to establish the overbooking rate; *with* these data the carrier would considerably improve the accuracy of the forecast. It is true that a detailed view by the carrier of the data of a single forwarder would violate confidentiality, but an aggregated view of the data (possibly with a partial obfuscation) would represent a lesser information disclosure; furthermore the resulting forecast improvement could be rather profitable. From the profits of this improvement, the carrier can draw incentives, and reward the forwarder companies for their collaboration. Those incentives could be proportional to the impact of the provided information on the improvement of the forecast. The carrier and its forwarder company customers would fairly benefit from the adoption of this forecast system. In general, although forwarders compete against one another and carriers compete against one another, each carrier collaborates with its own customers: despite the fact that they have contrasting interests for what concerns service levels and prices, they share the interest that the supply chain works efficiently. In a future work we will describe one such privacy preserving collaborative forecast system.

Acknowledgements

The authors acknowledge the support of the Information and Communication Technology Fund (ICT Fund) at EBTIC/Khalifa University of Science and Technology, Abu Dhabi, UAE (Project number 88434000029). The work was partially founded also by the EU Horizon 2020 research and innovation programme, within the projects Toreador (grant agreement No. 688797), Evotion (grant agreement No. 727521) and Threat- Arrest (Project-ID No. 786890).

References

1. http://www.joc.com/air-cargo/qantas-moves-impose-stiff-fees-late-cancellations-air-cargo_19881011.html.
2. Marco Anisetti, Valerio Bellandi, Ernesto Damiani, Fulvio Frati, Gabriele Gianini, Gwanggil Jeon, and Jechang Jeong. Supply chain risk analysis: open source simulator. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2009 Fifth International Conference on*, pages 443–450. IEEE, 2009.
3. Marco Anisetti, Ernesto Damiani, Fulvio Frati, Stelvio Cimato, and Gabriele Gianini. Using incentive schemes to alleviate supply chain risks. In *Proceedings of the International Conference on Management of Emergent Digital EcoSystems*, pages 221–228. ACM, 2010.
4. Rudolf Avenhaus, Bernhard Von Stengel, and Shmuel Zamir. Inspection games. *Handbook of game theory with economic applications*, 3:1947–1987, 2002.
5. Mokhtar Bazaraa, Joseph D Hurley, Ellis L Johnson, George L Nemhauser, Joel S Sokol, I-Lin Wang, Ek Peng Chew, Huei Chuen Huang, Ivy Mok, Kok Choon Tan, et al. The asia pacific air cargo system. *The Logistics Institute–Asia Pacific, Research paper no. TLI-AP/00/01, http://www.tliap. nus. edu. sg/TliapOpeningWebsite/research/white_papers_document/Air-Cargo-Report-16012001. pdf*, 2000.
6. Valerio Bellandi, Stelvio Cimato, Ernesto Damiani, Gabriele Gianini, and Antonio Zilli. Toward economic-aware risk assessment on the cloud. *IEEE Security & Privacy*, 13(6):30–37, 2015.
7. Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *In EUROCRYPT*, pages 127–144. Springer-Verlag, 1998.
8. P Ceravolo, S Cimato, E Damiani, G Gianini, C Fugazza, and S Marrara. Risk management and information disclosure in supply chain analysis. In *Conference on Advanced Information Technologies for Management (AITM)*. Publishing house of the Wrocław University of economics, 2008.
9. Chris Coppersmith. Airlines, forwarders must work together. *Journal of Commerce*, 4(11):38, 2003.
10. G. Lena Cota, S. Ben Mokhtar, G. Gianini, E. Damiani, J. Lawall, G. Muller, and L. Brunie. Racoon++: A semi-automatic framework for

the selfishness-aware design of cooperative systems. *IEEE Transactions on Dependable and Secure Computing*, PP(99):1–1, 2017.
11. G. Lena Cota, S. Ben Mokhtar, G. Gianini, E. Damiani, J. Lawall, G. Muller, and L. Brunie. Analysing selfishness flooding with seine. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 603–614, June 2017.
12. Guido Lena Cota, Sonia Ben Mokhtar, Julia Lawall, Gilles Muller, Gabriele Gianini, Ernesto Damiani, and Lionel Brunie. A framework for the design configuration of accountable selfish-resilient peer-to-peer systems. In *Reliable Distributed Systems (SRDS), 2015 IEEE 34th Symposium on*, pages 276–285. IEEE, 2015.
13. Ernesto Damiani, Paolo Ceravolo, Stelvio Cimato, and Gabriele Gianini. Obfuscation for the common good. In *Conference on Security in Network Architectures and Information Systems (SAR-SSI)*, pages 15–35. Publibook, 2008.
14. Ernesto Damiani, Stelvio Cimato, and Gabriele Gianini. A risk model for cloud processes. *The ISC International Journal of Information Security*, 6(2):99–123, 2014.
15. Ernesto Damiani, Gabriele Gianini, Florian Kerschbaum, and Richard Pibernik. Toward value-based control of knowledge sharing in networked services design. *Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu*, (85 Advanced Information Technologies for Management-AITM 2009):51–65, 2009.
16. Ernesto Damiani, Gabriele Gianini, and Marcello Leida. Toward behavioral business process analysis. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 2347–2353. IEEE, 2015.
17. Changyu Dong, Giovanni Russello, and Naranker Dulay. Shared and searchable encrypted data for untrusted servers. *Journal of Computer Security*, 19(3):367–397, 2011.
18. Melvin Dresher. A sampling inspection problem in arms control agreements: A game-theoretic analysis. Technical report, DTIC Document, 1962.
19. Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.
20. F Frati, E Damiani, P Ceravolo, S Cimato, C Fugazza, G Gianini, S Marrara, and O Scotti. Hazards in full-disclosure supply chains. In *Conference on Advanced Information Technologies for Management (AITM)*. Publishing house of the Wrocław University of economics, 2008.
21. Gabriele Gianini, Ernesto Damiani, Tobias R Mayer, David Coquil, Harald Kosch, and Lionel Brunie. Many-player inspection games in networked environments. In *Digital Ecosystems and Technologies (DEST), 2013 7th IEEE International Conference on*, pages 1–6. IEEE, 2013.
22. Rolf Hellermann. *Capacity options for revenue management: theory and applications in the air cargo industry*, volume 575. Springer Science &

Business Media, 2006.
23. Rolf Hellermann, Arnd Huchzermeier, and Stefan Spinler. Options contracts with overbooking in the air cargo industry. *Decision Sciences*, 44(2):297–327, 2013.
24. Raja G Kasilingam. Air cargo revenue management: Characteristics and complexities. *European Journal of Operational Research*, 96(1):36–44, 1997.
25. Qin Liu, Guojun Wang, and Jie Wu. Clock-based proxy re-encryption scheme in unreliable clouds. In *Parallel Processing Workshops (ICPPW), 2012 41st International Conference on*, pages 304–305. IEEE, 2012.
26. John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.
27. Lucio Pompeo and Ted Sapountzis. Freight expectations. *The McKinsey Quarterly*, 2:90–99, 2002.
28. Maryam Sepehri, Stelvio Cimato, and Ernesto Damiani. Privacy-preserving query processing by multi-party computation. *The Computer Journal*, 2014.
29. Maryam Sepehri, Stelvio Cimato, Ernesto Damiani, and Chan Yeob Yeuny. Data sharing on the cloud: A scalable proxy-based protocol for privacy-preserving queries. In *2015 IEEE TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*, pages 1357–1362, 2015.
30. Maryam Sepehri, Stelvio Cimato, Ernesto Damiani, and Chan Yeob Yeuny. Data sharing on the cloud: A scalable proxy-based protocol for privacy-preserving queries. In *Proceedings of the 7th IEEE International Symposium on Ubisafe Computing in conjunction with 14th IEEE Conference on Trust, Security and Privacy in Computing and Communications,TrustCom/BigDataSE/ISPA, Helsinki, Finland, August 20-22, 2015, Volume 1*, pages 1357–1362, 2015.
31. D. H. Tran, H. L. Nguyen, W. Zha, and W. K. Ng. Towards security in sharing data on cloud-based social networks. In *2011 8th International Conference on Information, Communications Signal Processing*, pages 1–5, Dec 2011.
32. Jiang Zhang and Zhenfeng Zhang. Secure and efficient data-sharing in clouds. *Concurrency and Computation: Practice and Experience*, 27(8):2125–2143, 2015.