GUC-Secure Join Operator in Distributed Relational Database

TIAN Yuan
Software School of Dalian University of Technology
tianyuan_ca!@sina.com

Abstract Privacy-preserving SQL computation in distributed relational database is one of important applications of secure multiparty computation. In contrast with comparatively more works on privacy-preserving data-query in database, only few works deal with provably-secure privacy-preserving data manipulations, among which the join operator is the most powerful in generating new data (relation). We propose a very general cryptographic protocol framework for secure 2-party join computation based on anonymous IBE (identity-based encryption) scheme and its user private-keys blind generation techniques. This construction is provably GUC (generalized universally composable) secure in standard model with acceptable efficiency. In addition, an efficient instantiation based-on the anonymous Boyen-Waters IBE scheme is presented which user private-key's blind generation protocol may be of independent values.

Keywords: Secure Multiparty Computation; Distributed Relational Database; Join; Anonymous Identity-based Encryption; Generalized Universally Composable Security.

1 INTRODUCTION

Relational database (RDBMS) is one of the most widely deployed and mission-critical information systems. In addition, distributed relational database technology has become one of the major recent developments in this area [15]. In distributed RDBMS, unlike a single centralized RDBMS server, information is distributed among multiple sites and some pieces of this information are highly private for its owners. Therefore a natural secure multiparty computation problem arises in this setting: how to generate the useful information while keeping all participiating sites' data privacy?

In RDBMS information is modeled as relations or tables. Such highly structured information is processed by the powerful relational algebra operators, e.g., (constrained) selection, projection, product and join operators, which are applied to one or multiple tables to generate new table [7]. Set operators, e.g., union, intersection, insert, etc., are also regarded as extended relational algebra operators. All these have been formally standardized as the RDBMS programming language SQL(Structured Query Language) which is widely used even in distributed RDBMS systems.

Motivations So far lots of works have been done to develop solutions to secure SQL computations in distributed RDBMS and the most majority comes from database rather than cryptography community. However, from the cryptography theorist's perspective, only few of them are reliable: most works, although highly heuristic, do not have precise security analysis or even without precise security definitions. As a (unfortunate) result, most of them are actually either insecure under the standard security definition currently accepted by cryptographists or unable-to-be-proved. In particular, since the SQL-operator is always called by high-level programs, its security must be preserved no matter how it is called and combined with any other (maybe malicious) running contexts. In other words, secure solutions to SQL-operators must be provably UC or GUC secure [4-5]. But with our knowledge

none of such solution exists in this area. For this reason, we won't further review these works from database community (interested readers can refer the survey [9] and its references) and only focus on provably secure solutions.

Among the provably secure solutions to distributed data processing (not limited to distributed RDBMS), most of them are focused on data query(i.e., read-only) rather than data manipulation operators. Although data query is one of the most important and frequently-used operations in distributed systems, lack of secure solutions to data manipulation operators will become a bottleneck in deploying highly security-critical applications. Roughly speaking, secure data query solutions are applied on encrypted data (e.g.,[1,2,11,16-17], many approaches are not limited to relational data model) to extract the desired information via smartly designed trapdoor machenisms. All these solutions can provide techniques for constructing secure SQL-query operators such as the most frequently-used constrained selection operators. Generally speaking, secure data manipulation is more technically involved than data query schemes and so far there are indeed few works on this problem, among which [10,12-13] provide the provably secure solutions. [10,13] can be also regarded as the secure solution to SQL set operators, but none of them reaches Canetti's UC/GUC-security [4-5] ([10] works in two relaxed adversary models to achieve security of "half-simulatability" and "full-simulatability against covert adversaries"; [13] is provably secure in terms of full-simulatability which is still strictly weaker than UC/GUC-security). With our knowledge, there're no satisfactory secure solutions to important relational data manipulation operators such as join and its variants.

In this paper we focus on secure join computation. Join is a powerful SQL operator to generate a new table from existing tables in RDBMS. For example, given two tables, *Incom* with attributes "*customer_id*"and "*income*", *Debt* with attributes "*customer_id*" and "*debt*", then the joint operator *Join*(*customer_id*: *Incom*,*Debt*) produces a new table with attributes "*customer_id*", "*income*" and "*debt*" which entries are derived from the product of the entries in table *Incom* and *Debt* which have the same values in *customer_id* field. For instance, suppose *Incom* has entries {(c1,2500),(c2,3000),(c5,1010),(c6,2000)}, *Debt* has entries {(c2,19000),(c4,7000),(c5,88),(c7,100)}, then the above join operator outputs the entries {(c2,3000,19000),(c5,1010,88)}. Join computation with equality constraints on multiple common attributes is also useful in practice.

Furthermore, if the table *Incom* and *Debt* are stored at different sites and the information about those uncommon customers must be kept secret to each other site, this join operator must be securely implemented in consistency with the cryptographic multiparty computation definition, i.e., each site knows nothing about the other table beyond the join operator's output.

Contributions In this paper we construct a GUC-secure [4-5] protocol for join computation in standard model with acceptable efficiency. Like many theoretical works, we focus on the 2-party case. Our approach is very general based-on the anonymous IBE scheme and it's user private-key's blind generation techniques (i.e., to generate the correct user private-key $usk(a)$=UKG(msk,a) for a user without leaking the user-id a to the key-generator). The protocol's high-level description is simple: let Π=(Setup,UKG,E,D) be an IBE scheme both anonymous and data-private, M_0 be some publicly-known plaintext. Site P_1 owns the table X_1 with attributes w and x and site P_2 owns table X_2

with attributes w and y. The goal is securely computing $Join(w{:}X_1,X_2)$ and outputing at P_2[1]. suppose X_1 is filled with entries $\{(w_1,x_1), (w_2,x_2), (w_3,x_3), (w_4,x_4)\}$ and X_2 is filled with $\{(w_2,y_2), (w_4,y_4), (w_5,y_5), (w_7,y_7)\}$ where w_i's, x_i's and y_i's are values of the attribute w, x and y respectively. P_1 generates IBE's master public/secret-key (mpk,msk), sends mpk and all $\xi_i=E(mpk,w_i,x_i||M_0)(i=1,2,3,4)$ to P_2. When P_2 tries to decipher each ξ_i by private-keys $usk(w_2)$, $usk(w_4)$, $usk(w_5)$ and $usk(w_7)$(obtained via Π's user private-keys blind generation protocol), only $usk(w_2)$ and $usk(w_4)$ can succeed in obtaining the plaintext with a suffix M_0. As a result, P_2 gets X_1's entries $\{(w_2,x_2),(w_4,x_4)\}$ and can now get the result of $Join(w{:}X_1,X_2)$, i.e., $\{(w_2,x_2,y_2),(w_4,x_4,y_4)\}$, by a local join computation. Note that Π's anonymity and data-privacy prevents P_2 from knowing anything about X_1 beyond $\{(w_2,x_2),(w_4,x_4)\}$ while the private-key generation protocol's blindness prevents P_1 from knowing anything about X_2.

This protocol's incorrectness probability is not 0, but by chosing M_0 lengthy enough, e.g., 128-bit, its incorrectness probability can be negligible in practice.

To be GUC-secure, the formal constrution is more involved (section 3). It is constant-round in communications and linear-size in message-complexity. In computation-complexity, one party is $O(N_1+N_2)$ and the other is $O(N_1N_2)$ encryptions/decriptions where N_1, N_2 are each party's private table's cardinality. Note that $O(N_1N_2)$ is also local join operator's computation complexity [7], i.e., neglecting a constant factor our construction's efficiency is asymptotically the same as that of conventional join operator.

The formal construction is also well-modularized, only executing few zero-knowledge proofs of knowledge which can be efficiently instantiated. Most importantly and distinctively, our construction is provably GUC-secure against malicious adversaries assuming static corruptions in the ACRS(augmented common reference string) model [5]. For this goal we introduce a notion of identity-augmented non-malleable zero-knowledge proofs of knowledge which may be of independent values. This construction can be also enhanced to be GUC-secure against malicious adversaries assuming adaptive corruptions in erasure model by some slight modifications.

Why IBE? Although there can be other approaches to the same solution, our approach has the good potential to deal with more complicated cases within a unified framework. In fact what we deal with in this paper is just the most simple (also most frequently-used) case: *equijoin*. A general join operator (called theta-join) involves some condition on attributes from each of its argument tables, i.e., $Join(\theta(\boldsymbol{a},\boldsymbol{b}){:}\ X_1,X_2)$ where θ is a predicate, $\boldsymbol{a}$, $\boldsymbol{b}$ are vectors of attributes of X_1 and X_2 respectively, e.g., $\boldsymbol{a}=(a^{(1)},a^{(2)})$, $\boldsymbol{b}=(b^{(1)},b^{(2)})$, for equijoin $\theta(\boldsymbol{a},\boldsymbol{b})\equiv a^{(1)}=b^{(1)} \wedge a^{(2)}=b^{(2)}$ but in general $\theta(\boldsymbol{a},\boldsymbol{b})$ can be any predicate, e.g., $a^{(1)}<b^{(1)}$, $a^{(1)}+a^{(2)}\geq b^{(1)}+b^{(2)}$, $a^{(1)}<b^{(1)} \wedge a^{(2)}=b^{(2)}$, etc. We believe that some further extensions to recently proposed ABE schemes(e.g.,[11,17]) which are powerful generalizations of IBE will be helpful to solve such secure SQL computation problems while keeping the general protocol framework in section 3 unchanged, which has obvious advantages in practice.

Organization Section 2 briefly presents all required important notions and facts, including an enhanced concept of zero-knowledge proofs of knowledge (def. 2.4). Section 3 presents the general construction and section 4 instantiates it via the anonymous Boyen-Waters IBE scheme. For space limitations, detailed proofs are presented in Appdenix B and C. Appendix D presents a general and

[1] In most cases a distributed SQL-transaction outputs its final result at some particular site.

efficient construction of the required zero-knowledge proof protocol.

2 NOTATIONS, DEFINITIONS AND TOOLS

P.P.T. means "*probabilistic polynomial-time*", $x||y$ means string x and y in concatenation, $|x|$ means string x's size(in bits) and $|X|$(X is a set) means X's cardinality, $x \leftarrow^{\$} X$ means randomly selecting x from the domain X. k denotes the complexity parameter. $\approx^{PPT}$ stands for *computational indistinguishability* and $\approx$ for *perfect indistinguishability*.

2.1 Secure Join Computation and Its GUC Security

Briefly speaking, GUC-security means that any adversary attacking the real-world protocol can be efficiently simulated by an adversary attacking the ideal-world functionality, both have the outputs indistinguishable by the (malicious) environment. For space limitations, we assume the reader's familiarity with the whole theory in [4-6] and only provide necessary descriptions with respect to the secure join computation problem here.

Similar to many theoretical works, we focus on the 2-party scenario. Furthermore, in this paper we only consider the horizontal distribution setting in which each table is at a site as a whole, none table is separated among different sites. Let X_1 and X_2 denote the tables with attributes w, $u^{(1)}$ and w, $u^{(2)}$ respectively, w is the common attribute and for equijoin we consider only single such common attribute without loss of generality (in case of multiple common attributes, e.g., v and w, we simply consider a imaginary single attribute $v||w$ which values are just the concatenation of values of v and w). The ideal cryptographic functionality to perform equijoin computation on X_1 and X_2 with equality constraint on w is defined as

$$F_{Join}: (X_1,X_2) \rightarrow (|w|_2, |X_1|||Join(w{:}X_1,X_2))$$

where $|w|_2$ means the number of different values of w in X_2 and $Join(w{:}X_1,X_2)$ means the reault of ideal relational join computation. More precisely, let P_1^*, P_2^* be parties in ideal model with private tables X_1 and X_2 respectively, $N_1=|X_1|$, $N_2=|X_2|$, S be the adversary in ideal model, the ideal model works as follows:

> *On receiving message* (sid,"input",P_1^*,X_1) *from* P_1^*, F_{Join} *records* X_1 *and sends message* (sid,"input",N_1) *to* P_2^* *and* S; *On receiving message* (sid,"input",P_2^*,X_2) *from* P_2^*, F_{Join} *records* X_2 *and sends* (sid,"input",$|w|_2$) *to* P_1^* *and S.*
>
> *On receiving message* (sid,"equijoin", P_2^*) *from* P^*_2, F_{Join} *responses* P_2^* *with message* (sid,"equijoin", $Join(w{:}X_1,X_2)$).
>
> *At last* P_1^* *outputs* $|w|_2$, P_2^* *outputs* $N_1||Join(w{:}X_1,X_2)$.

Let ψ be the real-world protocol, each party P_i of ψ corresponds to an ideal-world party P_i^*. A is the real-world adversary attacking ψ, Z is the environment in which the real protocol/ideal functionality executes. According to [4-5], Z is a P.P.T. machine modeling all malicious behaviors against the protocol's execution. Z is empowered to provide inputs to parties and interacts with A and S, e.g., Z gives special inputs or instructions to A/S, collects outputs from A/S to make some analysis, etc. In UC theory [4], Z cannot access parties' shared functionality (such shared functionality is specified in specific protocol) while in the improved GUC theory [5] Z is enhanced to do this, i.e., to

provide inputs to and get outputs from the shared functionality. As a result, in GUC theory Z is strictly stronger and more realistic than in UC theory.

Let $\text{output}_Z(\psi,A)$ denote the outputs (as a joint stochastic variable)from ψ's parties P_1, P_2 under Z and A, $\text{output}_Z(F_{Join},S)$ denote the similar thing under Z and S. During the real/ideal protocol's execution, Z (as an active distinguisher) interacts with A/S and raises its final output, w.l.o.g., 0 or 1. Such output is denoted as $Z(\text{output}_Z(\psi,A),u)$ and $Z(\text{output}_Z(F_{Join},S),u)$ respectively, where u is the auxiliary information.

Definition 2.1(GUC security [5]) If for any P.P.T. adversary A in real-world, there exists a P.P.T. adversary S (called A's simulator) in ideal-world, both corrupt the same set of parties, such that for any environment Z the function $|P[Z(\text{output}_Z(\psi,A),u)=1]-P[Z(\text{output}_Z(F_{Join},S),u)=1]|$ is negligible in complexity parameter k (hereafter denote this fact as $\text{output}_Z(\psi,A)\approx^{PPT}\text{output}_Z(F_{Join},S)$), then we define that ψ GUC-emulates F_{Join} or say ψ is GUC-secure, denoted as $\psi\rightarrow^{GUC}F_{Join}$.

The most significant property of GUC-security is the universal composition theorem. Briefly speaking, given protocols φ_2, φ_1 and $\psi(\varphi_1)$ where $\psi(\varphi_1)$ is the so-called φ_1-hybrid protocol, if $\varphi_2\rightarrow^{GUC}\varphi_1$ then (under some technical conditions, e.g., subroutine-respecting) $\psi(\varphi_2/\varphi_1)\rightarrow^{GUC}\psi(\varphi_1)$ where $\psi(\varphi_2/\varphi_1)$ is a protocol in which every call to the subprotocol φ_1 is replaced with a call to φ_2. This guarantees that a GUC-secure protocol can be composed in any execution context while still preserving its proved security. A similar consequence is also ture in UC theory but with some serious constraints. All details are presented in [4-5]. ACRS model is defined in [5]'s sec.4 and repeated in Appendix A in our paper.

2.2 IBE Scheme, Its Anonymity and Blind User-Private Key Generation Functionality

In addition to data-privacy, anonymity(key-privacy) is another valuable property for public-key encryption schemes [1]. An IBE scheme Π=(Setup, UKG, E, D) is a group of P.P.T. algorithms, where Setup takes as input the complexity parameter k to generate master public/secret-key pair (mpk, msk), UKG takes as input msk and user's id a to generate a's user private-key $usk(a)$; E takes (mpk,a,M) as input where M is the message plaintext to generate ciphertext y, D takes (mpk,$usk(a)$,y) as input to do decryption. Altogether these algorithms satisfy the consistency property: for any k, a and M

$$P[(mpk,msk)\leftarrow\text{Setup}(k);\ usk(a)\leftarrow\text{UKG}(msk,a);\ y\leftarrow\text{E}(mpk,a,M):\ \text{D}(mpk,usk(a),y)=M]=1$$

Definition 2.2(IBE Scheme's Chosen Plaintext Anonymity [1]) Given an IBE scheme Π=(Setup, UKG,E,D), for any P.P.T. attacker $A=(A_1,A_2)$ consider the following experiment $Exp_{\Pi,A}^{ANO-CPA}(k)$:

$(mpk, msk)\leftarrow\text{Setup}(k)$;
$(M^*, a_0^*, a_1^*, St)\leftarrow A_1^{\text{UKG}(msk,.)}(mpk)$, $a_0^*\neq a_1^*$;
$b\leftarrow^{\$}\{0,1\}$;
$y^*\leftarrow\text{E}(mpk, a_b^*, M^*)$;
$d\leftarrow A_2^{\text{UKG}(msk,.)}(St,y^*)$;
$\text{output}(d\oplus b)$;

A is contrained not to query its oracle UKG(msk,.) with a_0^* and a_1^*. Define $Adv_{\Pi,A}^{ANO-CPA}$ as $|2P[Exp_{\Pi,A}^{ANO-CPA}(k)=1]-1|$. If $Adv_{\Pi,A}^{ANO-CPA}$ is negligible in k for any P.P.T. A then Π is defined as

anonymous against chosen plaintext attack or ANO_CPA for short. In the above, if M^*, a_0^*, a_1^* are generated independent of *mpk* then Π is called id-*selective* ANO_CPA .

Denote $\max_{A \in P.P.T.} Adv_{\Pi,A}^{ANO\_CPA}(k)$ as $Adv_{\Pi}^{ANO\_CPA}(k)$ or $Adv_{\Pi}^{ANO\_CPA}(t,q)$ where t is the adversary's maximum time-complexity and q is the maximum number of queries for the UKG-oracle.

Now we present the ideal functionality $F^{\Pi}_{\text{Blind-UKG}}$ for an IBE scheme Π's user private-key blind generation(note: even IBE scheme is not anonymous such functionality still makes sense. However, in this paper only anonymous IBE's such protocol is needed). In the ideal model, one party generates(just one time) Π's master public/secret-key pair (*mpk*,*msk*) and submits it to $F^{\Pi}_{\text{Blind-UKG}}$; $F^{\Pi}_{\text{Blind-UKG}}$ generates *usk*(*a*)=UKG(*msk*,*a*) for another party who submits its private input *a*(this computation can take place any times and each time for a new *a*), revealing nothing about *a* to the party who provides (*mpk*,*msk*) except how many private-keys are generated. Formally, let *S* be the ideal adversary, P_1^*, P_2^* the ideal party, *sid* and *ssid* the session-id and subsession-id respectively, the ideal model works as follows:

P_1^* *selects randomness* ρ *and computes* (*mpk*,*msk*)←Setup(ρ), *sends the message* (sid, *mpk*||*msk*||ρ) *to* $F^{\Pi}_{\text{Blind-UKG}}$; $F^{\Pi}_{\text{Blind-UKG}}$ *sends message* (sid, *mpk*) *to* P_2^* *and S*;

On receiving a message (sid||ssid,*a*) *from* P_2^* (*ssid and a are fresh everytime*), *in response* $F^{\Pi}_{\text{Blind-UKG}}$ *computes usk*(*a*)←UKG(*msk*,*a*), *sends the message* (sid||ssid, *usk*(*a*)) *to* P_2^* *and the message* (sid||ssid, *n*) *to* P_1^* *and S*, *where n is initialized to be 0 and increased by 1 everytime the computation takes place*.

At last, P_1^* *outputs its last n*, P_2^* *outputs all its obtained usk*(*a*)*'s*.

2.3 (Identity-Augmented) Non-Malleable Zero-Knowledge Proofs of Knowledge

This subsection presents the concept of zero-knowledge proofs of knowledge following [8,14] with slight symbolic modifications. Let *L* be a NP language, *R* is its associated P-class binary relation. i.e., $x \in L$ *iff* there exists *w* such that $R(x,w)=1$. Let *A*, *B* be two machines, then $A(x;B)_{[\sigma]}$ represents *A*'s output due to its interactions with *B* under a public common input *x* and common reference string (c.r.s.) σ, $\text{tr}_{A,B}(x)_{[\sigma]}$ represents the transcripts due to interactions between *A* and *B* under a common input *x* and c.r.s. σ. When we emphasize *A*'s private input, say *y*, we also use the expression $A_y(x;B)_{[\sigma]}$ and $\text{tr}_{A(y),B}(x)_{[\sigma]}$ respectively. Let $A=(A_1,A_2)$, *B* and *C* be machines where A_1 can coordinate with A_2 by transferring status information to it, then $(<B,A_1>,<A_2,C>)$ represents the interaction between A_1 and *B*, (maybe concurrently) A_2 and *C*. Due to such interactions, let *tr* be the transcripts between A_2 and *C*, *u* be the final output from A_2 and *v* be the final output form *C*, then $(<B,A_1>,<A_2,C>)$'s output is denoted as (*u*,*tr*,*v*).

Two transcripts tr_1 and tr_2 are *matched* each other, if tr_1 and tr_2 are the same message sequence(consisted of the same messages in the same order) and the only difference is that any corresponding messages are in the opposite directions.

Let *A* be a machine, the symbol $\boxed{A}$ represents such a machine which accepts two kinds of instructions: the first one is in the form of ("start", *i*,*x*,*w*) and $\boxed{A}$ in response starts a new instance of *A*, associates it with a unique name *i* and provides it with public input *x* and private input *w*; the second is in form of ("message",*i*,*m*) and $\boxed{A}$ in response sends message *m* to instance A_i and then returns A_i's

response to m.

Definition 2.3(Zero-Knowldeg Proof and Non-Malleable Zero-Knowledge Proof Protocol [14]) $ZPoK_R=(D_{crs},P,V,Sim)$ where $Sim=(Sim_1,Sim_2)$ is a group of P.P.T. algorithms, k is complexity parameter, D_{crs} takes k as input and generates c.r.s. σ; P is called *prover*, takes (σ,x,w) as input where $R(x,w)=1$ and generates a proof π; V is called *verifier*, takes (σ,x) as input and generates 0 or 1; $Sim_1(k)$ generates (σ,s), Sim_2 takes $x\in L$ and (σ,s) as input and generates the simulation. All algorithms except D_{crs} and Sim_1 take the c.r.s. σ as one of their inputs, so σ is no longer explicitly included in all the following expressions unless for emphasis. Now $ZPoK_R$ is defined as a *zero-knowledge proof protocol for relation R*, if the following properties are all satisfied:

(1) For any $x\in L$ and $\sigma\leftarrow D_{crs}$, it's always true that $P[V(x;P)_{[\sigma]}=1]=1$;

(2) For any P.P.T. algorithm A, $x\notin L$ and $\sigma\leftarrow D_{crs}$, it's always true that $P[V(x;A)_{[\sigma]}=1]=0$[2];

(3) For any P.P.T. algorithm A which outputs 0 or 1, let ε be empty string, the function

$$|P[\sigma\leftarrow D_{crs};\ b\leftarrow A(\varepsilon;\boxed{P})_{[\sigma]}:\ b=1]-P[(\sigma,s)\leftarrow Sim_1(k);\ b\leftarrow A(\varepsilon;\boxed{Sim_2(s)})_{[\sigma]}:\ b=1]|$$

is always negligible in k, where we emphasize the fact by symbol $Sim_2(s)$ that all Sim_2 instances have the same s as one of their inputs.

The *non-malleable zero-knowledge proof protocol* for relation R is defined as $NMZPoK_R=(D_{crs},P,V,Sim,Ext)$ where $Sim=(Sim_1,Sim_2)$, $Ext=(Ext_1,Ext_2)$ and (D_{crs},P,V,Sim) is a zero-knowledge proof protocol for relation R as above, P.P.T. algorithm $Ext_1(k)$ generates (σ,s,τ) and the interactive P.P.T. machine Ext_2(named as witness extractor) takes (σ,τ) and protocol's transcripts as its input and extracts w, and all the following properties hold:

(4) The distribution of the first output of Sim_1 is identical to that of Ext_1;

(5) For any τ, the distribution of the output of V is identical to that of Ext_2's restricted output which does not include the extracted value (w);

(6) There exists a negligible function $\eta(k)$ (named as knowledge-error function) such that for any P.P.T. algorithm $A=(A_1,A_2)$ it's true that

$P[(\sigma,s,\tau)\leftarrow Ext_1(k);\ (x,tr,(b,w))\leftarrow(<\boxed{Sim_2(s)},A_1>,<A_2,Ext_2(\tau)>)_{[\sigma]}:\ b=1\wedge R(x,w)=1\wedge tr$ doesn't match any transcript generated by $\boxed{Sim_2(s)}$]

$>P[(\sigma,s)\leftarrow Sim_1(k);\ (x,tr,b)\leftarrow(<\boxed{Sim_2(s)},A_1>,<A_2,V>)_{[\sigma]}:\ b=1\wedge tr$ doesn't match any transcript generated by $\boxed{Sim_2(s)}$] $-\eta(k)$.

It's easy to see that $NMZPoK_R$ is a zero-knowledge proof of knowledge. [8,14] developed an efficient method to derive non-malleable zero-knowledge proof protocols based-on simulation-sound tag-based commitment schemes and the so-called Ω-protocols (proposed in [14]). In order to achieve GUC-security in our construction, we need to further enhance NMZPoK to the concept of identity-augmented non-malleable zero-knowldege proof protocol(IA-NMZPoK) as follows.

Definition 2.4(IA-NMZPoK Protocol for Relation R) The IA-NMZPoK Protocol for relation R, $IA\text{-}NMZPoK_R=(D,Setup,UKG,P,V,Sim,Ext)$ where $Sim=(Sim_1,Sim_2)$ and $Ext=(Ext_1,Ext_2)$, is a group of P.P.T. algorithms. Setup(k) generates master public/secret-key pair (mpk,msk), UKG(msk,id) generates id's private-key $usk(id)$ where $id\in\{P,V\}$(the prover's and verifier's identity). Sim_1 takes $usk(V)$ as input, Ext_1 takes $usk(P)$ as input. All algorithms except Setup take (mpk,σ) as one of its

[2] Strictly this protocol should be called "*zero-knowledge argument*", however, such difference is not essential in this paper so we harmlessly abuse the terminology.

inputs(so it no longer explicitly appears). The protocol has the same properties as *R*'s NMZPoK protocol in definition 2.3.

Note that by this definition an IA-NMZPoK protocol works in ACRS model [5] which ACRS is its *mpk*. In addition, only the corrupt verifier can run Sim (Sim_1 taking *usk*(*V*) as input) and only the corrupt prover can run Ext (Ext_1 taking *usk*(*P*) as input). This is exactly what is required in ACRS model. Given a relation *R*, a general and efficient construction of IA-NMZPoK protocol for *R* is presented in Appendix D.

2.4 Commitment Scheme

We need the non-interactive identity-based trapdoor commitment sheme [5](IBTC for short) as another important tool in our construction.

Definition 2.5(IBTC scheme [5]) Let *k* be complexity parameter, the non-interactive identity-based trapdoor commitment sheme IBTC=(D, Setup, UKG, Cmt, Vf, FakeCmt, FakeDmt) is a group of P.P.T. algorithms, where D(*k*) generates *id*, Setup(*k*) generates master public/secret-key pair (*mpk*, *msk*), UKG(*msk*,*id*) generates *id*'s user private-key *usk*(*id*), Cmt(*mpk*,*id*,*M*) generates message *M*'s commitment/decommitment pair (*cmt*,*dmt*), Vf(*mpk*, *id*, *M*, *cmt*, *dmt*) outputs 0 or 1, verifying whether *cmt* is *M*'s commitment with respect to *id*. These algorithms are consistant, i.e., for any *M*:

P[(*mpk*, *msk*)←Setup(*k*); (*cmt*,*dmt*)←Cmt(*mpk*,*id*, *M*): Vf(*mpk*, *id*, *M*, *cmt*, *dmt*)=1]=1

FakeCmt(*mpk*, *id*, *usk*(*id*)) generates ($\overline{cmt}$,λ), FakeDmt(*mpk*, *M*, λ, $\overline{cmt}$) generates $\overline{d}$ (w.l.o.g. λ contains *id*||*usk*(*id*) as one of its components so FakeDmt doesn't explicitly take *id* and *usk*(*id*) as its input). A secure IBTC scheme has the following properties:

(1)Hiding: for any *id* and M_0, M_1, (cmt_i,dmt_i)←Cmt(*mpk*,*id*, M_i), i=0,1, then $cmt_0 \approx^{P.P.T.} cmt_1$;

(2)Binding: for any P.P.T. algorithm *A*, the function $Adv^{binding}_{IBTC,A}(k)$≡P[(*mpk*, *msk*)←Setup(*k*); (*id**, *cmt**, M_0*, d_0*, M_1*, d_1*)←$A^{UKG(msk,.)}$(*mpk*): *A* doesn't query oracle-U(*msk*,.) with $id^* \wedge M_0^* \neq M_1^* \wedge Vf(mpk, id^*, M_0^*, cmt^*, d_0^*)=Vf(mpk, id^*, M_1^*, cmt^*, d_1^*)=1$] is always negligible in *k*.

(3)Equivocability: For any P.P.T. algorithm $A=(A_1,A_2)$ the following experiment always has |P[*b**=*b*]-1/2| upper-bounded by a negligible function in *k*:

(*mpk*, *msk*)←Setup(*k*);
(*St, id**, *M**)←A_1(*mpk*,*msk*);
usk(*id**)←UKG(*msk*,*id**); ($\overline{cmt}$,λ)←FakeCmt(*mpk*,*id**,*usk*(*id**));
d_1←FakeDmt(*mpk*, *M**, λ, $\overline{cmt}$); $d_0 \leftarrow^{\$} \{0,1\}^{|d_1|}$;
b←$^{\$}${0,1};
*b**←A_2(*St*, d_b);

Note that equivocability implies P[Vf(*mpk*, *id**, *M**, $\overline{cmt}$, d_1*)=1]>1-γ(*k*) where γ(*k*) is a negligible function in *k*. [5] presented an efficient IBTC construction and proved its security.

3 GENERAL CONSTRUCTION

Now we present the formal consctrution of the real-world private equijoin protocol Ψ. P_1 and P_2 denote two real-world parties with private tables X_1 and X_2, each with attributes *w*, *x* and *w*, *y* respectively (more generally *x* is the vector of all attributes in X_1 othe than *w*, similar for *y*, but this is

immaterial in our approach), $X_1=\{(u_1,x_1),\ldots,(u_{N_1},x_{N_1})\}$, $X_2=\{(v_1,y_1),\ldots,(v_{N_2},y_{N_1})\}$ where u_i's, v_j's are values of w, x_i's are values of x and y_j's are values of y. Π=(ESetup,UKG,E,D) is a id-selective ANO_CPA anonymous and id-selective IND_CPA data-private IBE scheme, $\Delta^{\Pi}_{\text{Blind-UKG}}$ is the real-world protocol for Π's user private-keys blind generation. In addition, we suppose a predefined bijective coding function H maping strings or values (e.g., x_i, y_j) to Π's plaintexts, but for simplicity we always write $E(mpk,a,x||y)$ instead of $E(mpk,a,H(x||y))$. IA-NMZPoK(w:$R(x,w)$=1) denotes an IA-NMZPoK protocol for relation R where w is x's witness. TC=(D,TSetup,UKG,Cmt,Vf,FakeCmt, FakeDmt) is an IBTC scheme. M_0 is a (fixed) public common string and $|M_0|$=poly(k). Ψ's ACRS is $mpk_{\text{TC}}||mpk_{\Delta}||mpk_{\text{ZK}}||M_0$ where mpk_{TC}, mpk_{Δ}, mpk_{ZK} are respectively TC's, $\Delta^{\Pi}_{\text{Blind-UKG}}$'s and an IA-NMZPoK protocol (see below)'s master public key. Ψ works as follows. For intuition it is also presented in a figure where the IA-NMZPoK protocol's arrow points from the prover to its verifier.

Equijoin Protocol Ψ: General Construction

(1) P_1 computes Π's master public/secret-key (mpk,msk)←ESetup(k), for each $(u_i,x_i)\in X_1$(i=1,…,N_1) computes ciphertext ξ_i←E(mpk,u_i,$x_i||M_0$; r_i) where r_i is the independent randomness in each encryption, then computes (cmt,dmt)←Cmt(mpk_{TC}, P_2, $\xi_1||\ldots||\xi_{N1}$) and sends $mpk||cmt$ to P_2.

(2) P_1 and P_2 run the protocol $\Delta^{\Pi}_{\text{Blind-UKG}}$ where P_1 (as the key-generater) inputs (mpk,msk) and P_2 (as the key-receiver) inputs all distinct $v_1,\ldots,v_N$ to $\Delta^{\Pi}_{\text{Blind-UKG}}$. On $\Delta^{\Pi}_{\text{Blind-UKG}}$'s completion, P_1 obtains N and P_2 obtains $usk(v_1),\ldots, usk(v_N)$ as the output.

(3) P_1 sends $\xi_1||\ldots||\xi_{N1}||dmt$ to P_2.

(4) P_2 verifies Vf(mpk_{TC}, P_2, $\xi_1||\ldots||\xi_{N1}$, cmt, dmt)=1.

(5) P_1 runs the protocol IA-NMZPoK((u_i,x_i,r_i): ξ_i=E(mpk, u_i, $x_i||M_0$; r_i), i=1,…,N_1) as a prover with P_2 as a verifier. On this IA-NMZPoK 's completion, P_2 tries to decrypt each ξ_i by all $usk(v_j)$'s it obtained in step 2 and everytime the output has suffix M_0, i.e., D(mpk,$usk(v_j)$,ξ_i)=$x_i||M_0$, it generates an entry (v_j, x_i). All such entries constitute a temporary table X_0, then P_2 performs a local join Y_0←Join(w:X_0,X_2).

(6) P_1 outputs N and P_2 outputs Y_0.

Note that $X_0=\{(v_j, x_i)\in X_1$: there exists ξ_i s.t. D(mpk, $usk(v_j)$, ξ_i)=$x_i||M_0\}$ and D(mpk,$usk(v_j)$,ξ_i)= $x_i||M_0$ implies $u_i=v_j$ with negligible exception, so Y_0=Join(w:X_1,X_2), i.e., Ψ's output is correct with only negligible exception probability.

Ψ is atually a $\Delta^{\Pi}_{\text{Blind-UKG}}$-hybrid protocol and we require $\Delta^{\Pi}_{\text{Blind-UKG}}\rightarrow^{\text{GUC}}F^{\Pi}_{\text{Blind-UKG}}$(definition 2.1). However, merely requiring $\Delta^{\Pi}_{\text{Blind-UKG}}\rightarrow^{\text{GUC}}F^{\Pi}_{\text{Blind-UKG}}$ cannot guarantee Ψ's GUC-security but only "half GUC-security" instead (i.e., the real adversary A corrupting P_1 can be completely simulated by an ideal adversary S but this is not true when A corrupts P_2. Only data-privacy can be proved in the latter case). In order to make the real adversary completely simulatable in ideal-world, some additional property is required for $\Delta^{\Pi}_{\text{Blind-UKG}}$. This leads to definition 3.1 and it is not hard to verify that our concrete construction of $\Delta^{\Pi}_{\text{Blind-UKG}}$ in next section really satisfies it.

Definition 3.1(IBE's User Private-keys Blind Generation Protocol with Extractor) Given IBE scheme Π=(ESetup,UKG,E,D) and $\Delta^{\Pi}_{\text{Blind-UKG}}\rightarrow^{\text{GUC}}F^{\Pi}_{\text{Blind-UKG}}$, let P_1, P_2 be $\Delta^{\Pi}_{\text{Blind-UKG}}$'s parties where P_2 provides user-id a and obtains $usk(a)$, P_1 owns msk and (blindly) gernates $usk(a)$ for P_2. This $\Delta^{\Pi}_{\text{Blind-UKG}}$ is defined as *extractable*, if there exists P.P.T. algorithm Setup_{Δ}, UKG_{Δ}, $\text{Ext}_{\Delta}=(\text{Ext}_1,\text{Ext}_2)$ and a negligible function $\delta(k)$, called the *error function*, such that

(1) $\text{Setup}_{\Delta}(k)$ generates the master public/secret-key pair (mpk_{Δ},msk_{Δ}).

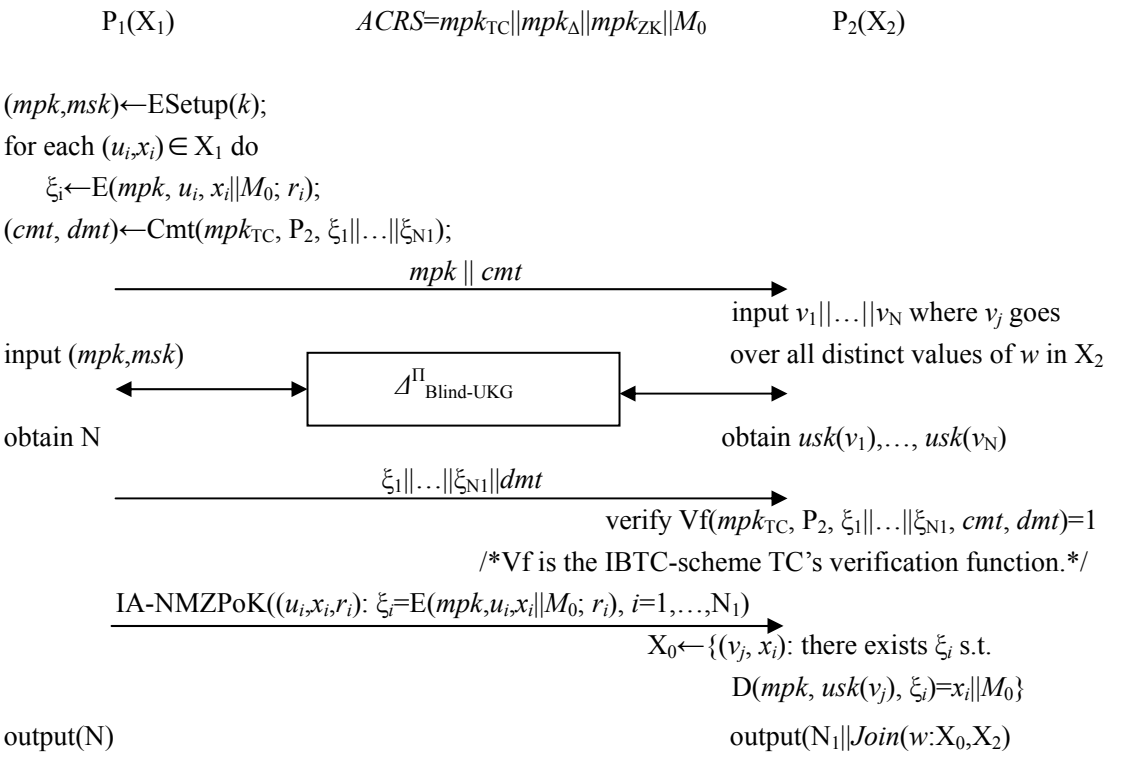
(2) $UKG_{\Delta}(msk_{\Delta},id)$ outputs a trapdoor $usk_{\Delta}(P_2)$ when id=P_2(key-receiver's identity) and outputs nothing otherwise.

(3) for any user-id a, honest P_1 and any P.P.T. algorithm A, it is true that(via notations in subsection 2.3) $Ext_1(usk(P_2))$ outputs (σ,τ) such that

$$P[Ext_2(mpk||\tau; A(a))_{[\sigma]}=a]>P[A_a(mpk; P_1(mpk,msk))_{[\sigma]}=UKG(msk,a)]-\delta(k)$$

where (mpk,msk) is Π's master public/secret-key owned by P_1(mpk is published).

We stress that all extractors in definition 2.3 and definition 3.1 are non-rewinding.



Combining all the instantiations of subprotocols in this general construction (some presented in next section and Appendix D), it's easy to see that we can get a O(1) and $O(N_1+N_2)$ message-complexity solution. The exact computation efficiency analysis can be only done for specific instantiations (e.g., that presented in next section) which is provided in the full version paper.

Theorem 3.1 *Suppose that IBE scheme* Π = (ESetup,UKG,E,D) *is both id-selective* ANO_CPA *anonymous and id-selective* IND_CPA *data-private*, $\Delta^{\Pi}_{\text{Blind-UKG}} \rightarrow^{GUC} F^{\Pi}_{\text{Blind-UKG}}$ *with extractor* $Ext_{\Pi}=(Ext_{\Pi,1},Ext_{\Pi,2})$ *and error function* δ *as in def.3.1*, *IA-NMZPoK*$((u_i,x_i,r_i)$: $\xi_i=E(mpk, u_i, x_i||M_0; r_i)$, $i=1,\ldots,N_1)$ *is an* IA-NMZPoK *protocol*, TC=(D,TSetup, UKG,Cmt,Vf,FakeCmt,FakeDmt) *is an* IBTC *scheme*, *then* $\Psi \rightarrow^{GUC} F_{join}$ *assuming static corruptions*.

The proof is in Appendix B.

4 AN INSTANTIATION VIA BOYEN-WATERS IBE SCHEME

Theorem 3.1 presents security conditions for the general construction Ψ, among which some are

available in existing works, e.g., the commitment scheme can be directly borrowed from [5]. The subprotocols which require new efficient constructions are only IBE scheme's user private-keys generation protocol and the protocol IA-NMZPoK((a,r): ξ=E(mpk,a,M_0;r)). In this section we present an efficient instantiation of Ψ via *Boyen-Waters* IBE scheme. The related zero-knowledge protocol's construction is presented in Appendix D.

4.1 Boyen-Waters IBE[3]

Given an bilinear group pairing ensemble $J=\{(p,G_1,G_2,e)\}_k$ where $|G_1|=|G_2|=p$, p is k-bit prime number, $P\in G_1$, e:$G_1\times G_1\rightarrow G_2$ is a non-degenerate pairing, *Boyen-Waters* IBE consists of

ESetup(k):

$g, g_0, g_1\leftarrow^{\$}G_1$; $\omega, t_1, t_2, t_3, t_4\leftarrow^{\$}Z_p$; $\Omega\leftarrow e(g,g)^{t_1t_2\omega}$;
$v_1\leftarrow g^{t1}$; $v_2\leftarrow g^{t2}$; $v_3\leftarrow g^{t3}$; $v_4\leftarrow g^{t4}$;
$mpk\leftarrow(G_1,G_2,p,e,\Omega,g,g_0,g_1,v_1,v_2,v_3,v_4)$;
$msk\leftarrow(\omega,t_1,t_2,t_3,t_4)$;
return(mpk,msk);

UKG(msk, a), $a\in Z_p$:

$r_1, r_2\leftarrow^{\$}Z_p$;
$usk(a)\leftarrow(g^{r_1t_1t_2+r_2t_3t_4}, g^{-\omega t_2}(g_0g_1^{\,a})^{-r_1t_2}, g^{-\omega t_1}(g_0g_1^{\,a})^{-r_1t_1}, (g_0g_1^{\,a})^{-r_2t_4}, (g_0g_1^{\,a})^{-r_2t_3})$;
return($usk(a)$);

E(mpk, a, M), $M\in G_2$:

$s, s_1, s_2\leftarrow^{\$}Z_p$; $\xi\leftarrow(\Omega^sM, (g_0g_1^{\,a})^s, v_1^{s-s1}, v_2^{s1}, v_3^{s-s2}, v_4^{s2})$; return($\xi$);

D(mpk, $usk(a)$, ($\xi_{00}, \xi_0, \xi_1, \xi_2, \xi_3, \xi_4$)) where $usk(a)\equiv(d_0, d_1, d_2, d_3, d_4)$:

$T\leftarrow e(d_0,\xi_0)e(d_1,\xi_1)e(d_2,\xi_2)e(d_3,\xi_3)e(d_4,\xi_4)$; return($\xi_{00}T$);

[3] has proven that assuming the decisional bilinear Diffie-Hellman problem(D-BDHP)'s hardness on J, this scheme is id-selective IND_CPA data-private; assuming the decisional linear problem(D-LP)'s hardness, this scheme is id-selective ANO_CPA anonymous. Notice that D-BDHP hardness implies D-LP's hardness, all the above consequences can be also obtained only under D-BDHP's hardness.

4.2 User Private-Keys Blind Generation Protocol $\Delta_{Blind-UKG}^{Boyen-Waters}$ and Its GUC-Security

For simplicity we only present how to blindly generate $usk(a)$ for a single user-id a. The generalization to blindly generating $usk(a_1)||\ldots||usk(a_N)$ for multiple user-id's $a_1||\ldots||a_N$ is trival and still constant-round, though the total message-complexity is linearly increased.

The two parties are P_1 (with private input msk) and P_2 (with private input a). Both parties have the common input mpk where (mpk,msk) are generated by IBE scheme's ESetup(k) (usually msk per se is the randomness in ESetup so we use a simplified notation $mpk\leftarrow$ESetup(msk) hereafter). $\Delta_{Blind-UKG}^{Boyen-Waters}$ has two IA-NMZPoK subprotocols (see below) which ACRS's are denoted as $mpk_{ZK,II}$ and $mpk_{ZK,III}$. $\Delta_{Blind-UKG}^{Boyen-Waters}$ is in ACRS model which ACRS is $mpk_{ZK,II}||mpk_{ZK,III}$. $\Delta_{Blind-UKG}^{Boyen-Waters}$ works as follows:

(1) P_1 runs a protocol IA-NMZPoK(*msk*: *mpk*=ESetup(*msk*)) as a prover with P_2 as a verifier, where the meaning of the notation IA-NMZPoK(*msk*: *mpk*=ESetup(*msk*)) follows section 3. Denote this protocol as IA-NMZPoK$_{II}$.

(2) P_2 selects r_1, r_2, y_1, y_2, y_3, y_4 at random, computes $U_i \leftarrow g^{r_i}$, $V_i \leftarrow (g_0 g_1^a)^{-r_i}$ for i=1,2 and $h_j \leftarrow g^{y_j} g_1^a$ for j=1,2,3,4, sends $U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4$ to P_1. Then P_2 runs the protocol

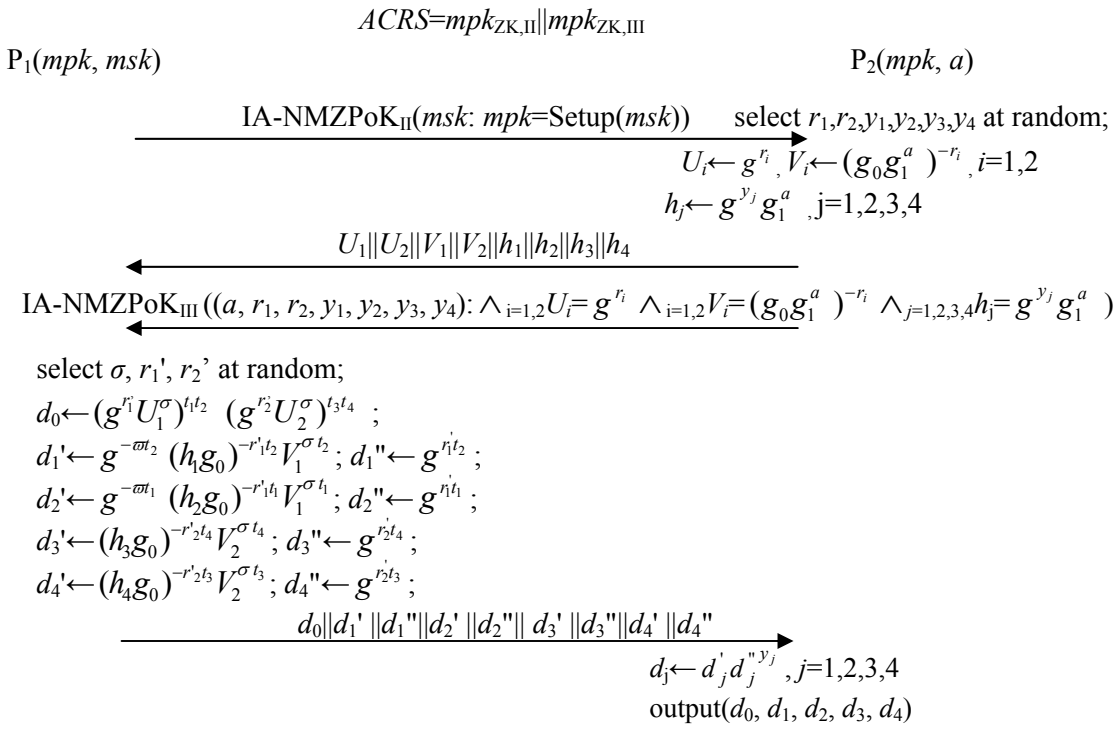
IA-NMZPoK($(a, r_1, r_2, y_1, y_2, y_3, y_4)$: $\wedge_{i=1,2} U_i = g^{r_i} \wedge_{i=1,2} V_i = (g_0 g_1^a)^{-r_i} \wedge_{j=1,2,3,4} h_j = g^{y_j} g_1^a$)

as a prover with P_1 as a verifier. Denote this protocol as IA-NMZPoK$_{III}$.

(3) P_1 selects σ, r_1', r_2' at random, computes $d_0 \leftarrow (g^{r_1'} U_1^{\sigma})^{t_1 t_2} (g^{r_2'} U_2^{\sigma})^{t_3 t_4}$; $d_1' \leftarrow g^{-\varpi t_2} (h_1 g_0)^{-r_1' t_2} V_1^{\sigma t_2}$; $d_1'' \leftarrow g^{r_1' t_2}$; $d_2' \leftarrow g^{-\varpi t_1} (h_2 g_0)^{-r_1' t_1} V_1^{\sigma t_1}$; $d_2'' \leftarrow g^{r_1' t_1}$; $d_3' \leftarrow (h_3 g_0)^{-r_2' t_4} V_2^{\sigma t_4}$; $d_3'' \leftarrow g^{r_2' t_4}$; $d_4' \leftarrow (h_4 g_0)^{-r_2' t_3} V_2^{\sigma t_3}$; $d_4'' \leftarrow g^{r_2' t_3}$ and sends $d_0||d_1'||d_1''||d_2'||d_2''||d_3'||d_3''||d_4'||d_4''$ to P_2.

(4) P_2 computes $d_j \leftarrow d_j' d_j''^{y_j}$, j=1,2,3,4 and outputs (d_0, d_1, d_2, d_3, d_4).

For intuition the protocol is presented in the figure below where IA-NMZPoKs' arrows point from zero-knowledge's prover to its verifier.



It's easy to show by direct calculation that P_2 outputs the correct $usk(a)=(d_0,d_1,d_2,d_3,d_4)$ where $d_0 = g^{(r_1'+r_1\sigma)t_1t_2+(r_2'+r_2\sigma)t_3t_4}$, $d_1 = g^{-\varpi t_2}(g_0 g_1^{\,a})^{-(r_1'+r_1\sigma)t_2}$, $d_2 = g^{-\varpi t_1}(g_0 g_1^{\,a})^{-(r_1'+r_1\sigma)t_1}$, $d_3 = (g_0 g_1^{\,a})^{-(r_2'+r_2\sigma)t_4}$, $d_4 = (g_0 g_1^{\,a})^{-(r_2'+r_2\sigma)t_3}$. Regarding security, we have

Theorem 4.1 *Suppose the bilinear group pairing* J has *D-BDHP hardness*, *both IA-NMZPoK*$_{II}$ *and IA-NMZPoK*$_{III}$ *are identity-augmented non-malleable zero-knowledge proof protocols for specific relations described in the above*, *then* $\Delta_{Blind-UKG}^{Boyen-Waters} \rightarrow^{GUC} F_{Blind-UKG}^{Boyen-Waters}$ *assuming static corruptions and* $\Delta_{Blind-UKG}^{Boyen-Waters}$ *satisfies def.* 3.1.

Appendix C gives the proof and Appendix D has the related IA-NMZPoK protocol's construction.

REFERENCES

[1] M.Abdalla, M.Bellare, D.Catalano et al. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE and Extensions. In Crypto'05, LNCS Vol. 3621, 205-222, 2005.

[2] I.Blake, V.Kolenilkov. Conditional Encrypted Mapping and Comparing Encrypted Numbers, In Proc. Financial Cryptography, 2006.

[3] X.Boyen, B.Waters. Anonymous Hierarchical Identity-based Encryption without Random Oracles, Crypto'06, LNCS Vol.4117, 290-307, 2006.

[4] R.Canetti, Universally Composable Security: a New Paradgim for Cryptographic Protocols, 42nd Annual Symposium on foundations of computer Science, IEEE Computer Society, 136-145, 2001.Updated in 2005, eArchive Cryptology 2001/067.

[5] R.Canetti, Y.Dodis, R.Pass et al Universally Composable Security with Global-Setup, TCC'07, LCNS Vol.4392, 61-85, 2007. Full version available at eArchive Cryptology 2007.

[6] Y.Dodis, V.Shoup, S.Walfish Efficient Construction of Composable Commitment and Zero-Knowledge Proofs, Proc. Crypt'08, 2008.

[7] H.Garacia-Molina, J.Ullman, J.Widom Database System Implementation, Prentice-Hall, Inc.,2001.

[8] J.Garay, P..MacKenzie, K.Yang Strengthening Zero-Knowledge Protocols using Signatures, Proc. Eurocrypt, LNCS Vol.2656, 177-194, 2003.

[9] H. Hacigumus, B.Iyer, C.Li, S.Mehrotra Executing SQL over Encrypted Data in the Database-Service-Provider Model. In *ACM SIGMOD*'06, 96-107, 2006.

[10] C.Hazay, Y.Lindell Efficient Protocols for Set Intersection and Pattern Matching with Security against Malicious and Covert Adversaries, Proc. CT-RSA'08, 2008.

[11] J.Katz, A.Sahai, B.Waters. Predicate Encrypted Supporting Disjunctions, Polynomial Equations and Inner Products, In Proc. Eurocrypt'08, 146-162, 2008

[12] Y.Lindell, B.Pinkas. Privacy Preserving Data Mining. Journal of Cryptography 15,3(2002), 177-206.

[13] L.Kissner, D.Song Private-Preserving Set Operations, Crypto'05, LNCS Vol.3621, 241-257, 2005.

[14] P.MacKenzie, K.Yang On Simulation-Sound Trapdoor Commitments, In Proc. Eurocrypt'04, LNCS Vol.3027, 382-400, 2004.

[15] M.T.Ozsu, P.Valduriez. Principles of Distributed Database Systems, Prentice-Hall, Inc., 1999

[16] E.Shi, J.Bethencourt, H.Chen, D.X.Song, A.Perrig. Multi-dimensional Range Quries over Encrypted Data. In IEEE Symposium on Security and Privacy, 2007.

[17] B.Waters. Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient and Provabley Secure Realization, eArchive 2008/304.

APPENDIX.A ACRS MODEL

Recently [5] improves and generalizes the early UC-theory proposed in [4] to make a more general, realistic and strictly stronger security notion. The universal composition theorem is still true in this paradigm, however, the pre-setup needs to be strictly enhanced. In GUC paradigm the CRS model is insufficient to implement general cryptographic functionalities, instead we need a new pre-setup model called ACRS(augmented common reference string) model. This pre-setup can be performed via a shared functionality $\overline{G}_{acrs}^{Setup,UKG}$ with two parameter functions Setup and UKG similar to IBE scheme's master public/secret-key generator and its user private-key generator. $\overline{G}_{acrs}^{Setup,UKG}$'s program is [5]:

Initialization Phase: *compute* (*mpk*, *msk*)←Setup(*k*) *and store* (*mpk*, *msk*);

Running Phase: *on receiving message* ("*CRS request*",P_i) *from any party* P_i, *response* ("*ACRS*", *mpk*) *to* P_i *and the adversary S*;

On receiving message ("*Retrieve*",*sid*,P_i) *from a corrupt party* P_i, *compute* *usk*(P_i)←UKG(*msk*,P_i) *and return the message* ("*Private-key*", *sid*, *usk*(P_i)) *to* P_i; *if* P_i *is not a corrupt party*, *response nothing*.

APPENDIX.B PROOF OF THEOREM 3.1

We prove the GUC-security in two cases that the real-world adversary *A* corrupts P_1 or P_2 respectively. Below P_1* and P_2* stand for P_1 and P_2's respective counterparts in ideal-world.

All parties are assumed to be initialized with a copy of the common reference string *ACRS*, i.e., the concatenation of TC's master public-key mpk_{TC}, $\Delta^{\Pi}_{\text{Blind-UKG}}$'s mpk_{Δ}, the IA-NMZPoK protocol's mpk_{ZK} and M_0, generated by the pre-setup G_{ACRS}. For this *ACRS*, its $msk=msk_{TC}||msk_{\Delta}||msk_{ZK}$ and UKG(*msk*,*id*) responses with $usk(id)=usk_{TC}(id)||usk_{\Delta}(id)||usk_{ZK}(id)$ where $usk_{TC}(id)$, $usk_{\Delta}(id)$ and $usk_{ZK}(id)$ are respectively TC's, $\Delta^{\Pi}_{\text{Blind-UKG}}$'s and the IA-NMZPoK protocol's user private-keys corresponding to $id\in\{P_1,P_2\}$.

(1) *A* corrupts P_1: for simplicity we first make the proof in $F^{\Pi}_{\text{Blind-UKG}}$–hybrid model and then complete the proof by generalized universal composition theorem. Let $X_1=\{(u_1^*,x_1^*),\ldots,(u_{N1}^*,x_{N1}^*)\}$ be *A*'s(i.e., P_1's) own table, $X_2=\{(v_1^*, y_1^*),\ldots,(v_{N1}^*, y_{N2}^*)\}$ be P_2*'s own table. We need to construct an ideal adversary S_1 who corrupts P_1*, runs *A* as a black-box and simulates the real-world honest party P_2 to interact with *A*:

On receiving the message (sid,"input",N_2) from F_{Join}, S_1 gets $usk(P_1)$ by querying the shared functionality G_{ACRS} with ("retrieve",sid,P_1) where $usk(P_1)=usk_{TC}(P_1)||usk_{\Delta}(P_1)||usk_{ZK}(P_1))$, computes (σ,s,τ)←IA-NMZPoK::$Ext_1(usk_{ZK}(P_1))$(to avoid ambiguity, we use $\Gamma::f$ to represent a protocol Γ's algorithm *f*), generates N_2 entries $(v_1, y_1),\ldots, (v_{N2}, y_{N2})$ at random and then starts *A*;

After *A* sends the first message $mpk||cmt$, S_1 interacts with *A* as an honest key-receiver in model of $F^{\Pi}_{\text{Blind-UKG}}$ and obtains $usk(v_1),\ldots,usk(v_{N2})$;

S_1 intercepts the message $\xi_1||\ldots||\xi_{N1}||dmt$ sent from *A*,verifies whether Vf(mpk_{TC}, P_2, $\xi_1||\ldots||\xi_{N1}$, *cmt*,*dmt*)=1 and then participates in protocol IA-NMZPoK((u^*_i, x^*_i, r_i): ξ_i=E(mpk, u_i*, $x^*_i||M_0$; r_i), $i=1,\ldots,N_1$ as a verifier calling the knowledge extractor IA-NMZPoK::$Ext_2(\tau)$ to extract the witness (u^*_i,x_i^*,r_i), $i=1,\ldots,N_1$ (in fact only u^*_i's and x_i*'s are needed in this proof);

S_1 sends the message (sid,"input",$\{(u^*_1,x_1^*),\ldots,(u_{N1}^*,x^*_{N1})\}$ to F_{Join}, then outputs whatever A outputs to the environment.

Let tr(A,S_1) denote the transcripts due to the interaction between S_1 and A, $\text{tr}^{\psi}(A, P_2(X_2))$ denote the transcripts due to the interaction between A and $P_2(X_2)$ in the real-world protocol Ψ($P_2(X_2)$ means the real-world party possessing the same private table X_2 as P_2*). From A's perspective, the difference between tr(A,S_1) and $\text{tr}^{\psi}(A, P_2(X_2))$ is that the former provides $F^{\Pi}_{\text{Blind-UKG}}$ with $\{v_1,\ldots,v_{N2}\}$ as the input, the latter provides $F^{\Pi}_{\text{Blind-UKG}}$ with $\{v_1^*,\ldots,v_{N2}^*\}$, but according to $F^{\Pi}_{\text{Blind-UKG}}$'s specification A knows nothing about what data entries are provided to $F^{\Pi}_{\text{Blind-UKG}}$ by the other party except the number N_2, as a result, $\text{tr}(A,S_1)\approx \text{tr}^{\psi}(A,P_2(X_2))$(perfectly indistinguishable) from A's perspective. In particular, the distribution of A's output due to interactions with S_1 is the same as that (in real-world protocol Ψ) due to interactions with $P_2(X_2)$. Let η be IA-NMZPoK protocol's error function, $Adv_{TC}^{binding}$ be attacker's advantage against TC's binding property, all are negligible functions in k. It's not hard to show (by contradiction) that the probability with which S_1 correctly extracts all A's data-entries $(u^*_1,x_1^*),\ldots,(u^*_{N1},x_{N1}^*)$ is greater than the probability $P[P_2(mpk\|\xi_1\|\ldots\|\xi_{N1};A)=1] - N_1(\eta + Adv_{TC}^{binding}) \geq P[P_2$ outputs $Join(w{:}X_1,X_2)] - N_1(\eta + Adv_{TC}^{binding})$, therefore, the difference between the probability with which $P_2^*(X_2)$ outputs $Join(w{:}X_1,X_2)$ under the ideal-world adversary S_1's attacks and the probability with which $P_2(X_2)$ outputs $Join(w{:}X_1,X_2)$ under the real-world adversay A's attacks against Ψ is upper-bounded by $N_1(\eta+ Adv_{TC}^{binding})$, also a negligible function in k. Combining all the above facts, for any P.P.T. environment Z we have $\text{output}_Z(\psi, A)\approx^{PPT} \text{output}_Z(F_{Join}, S_1)$, i.e., $\Psi\rightarrow^{GUC} F_{Join}$, in $F^{\Pi}_{\text{Blind-UKG}}$–hybrid model.

Now replace the ideal functionality $F^{\Pi}_{\text{Blind-UKG}}$ with $\Delta^{\Pi}_{\text{Blind-UKG}}$ in Ψ. By what is just proved, the assumption $\Delta^{\Pi}_{\text{Blind-UKG}}\rightarrow^{GUC} F^{\Pi}_{\text{Blind-UKG}}$ and the GUC-theorem, we still have the GUC-emulation consequence. In addition, it's not hard to estimate S_1's time complexity $T_{S1}=T_A+O(N_2+N_1T_e)$ where T_A and T_e are A's and the knowledge extractor's computation time.

(2) A corrupts P_2: Denote A's(i.e., P_2's) own table as $X_2=\{(v^*_1, y^*_1),\ldots,(v^*_{N2}, y^*_{N2})\}$, P_1*'s own table as $X_1=\{(u^*_1, x^*_1),\ldots,(u^*_{N1},x^*_{N1})\}$, we need to construct an ideal adversary S_2. S_2 corrupts P_2*, gets $usk(P_2)$ by querying the pre-setup G_{ACRS} with ("retrieve",sid,P_2) where $usk(P_2)=usk_{TC}(P_2)\|usk_{\Delta}(P_2)\|usk_{ZK}(P_2)$, generates $(\sigma, s)\leftarrow$IA-NMZPoK::$\text{Sim}_1(usk_{ZK}(P_2))$, runs A as a black-box and simulates the real-world honest party P_1 to interact with A:

On receiving message (sid,"input",N_1) from F_{Join}, S_2 generates $(u_1, x_1),\ldots,(u_{N1}, x_{N1})$ at random, computes $(mpk,msk)\leftarrow\text{Setup}(k)$ and $\xi_i\leftarrow E(mpk,u_i,x_i\|M_0; r_i)$ for each (u_i,x_i) where r_i is the independent randomness in each encryption, computes $(cmt^0,\lambda)\leftarrow\text{FakeCmt}(mpk_{TC},P_2,usk_{TC}(P_2))$, starts A and sends the message $mpk\|cmt^0$ to A;

S_2 interacts with A as the user private-key generator in $\Delta^{\Pi}_{\text{Blind-UKG}}$ and calls the extractor $\Delta^{\Pi}_{\text{Blind-UKG}}$::$\text{Ext}_{\Delta}(usk_{\Delta}(P_2))$ to extract $v^*_1,\ldots,v^*_N$ (N is the number of distinct v^*_i's), generates $y_1,\ldots, y_N$ at random, sends the message (sid,"input", P_2*, $\{(v^*_1, y_1),\ldots,(v^*_N, y_N)\}$ to F_{Join};

S_2 sends the message (sid,"join",P_2*) to F_{Join} and gets the response $\{(u^*_{j1}, x^*_{j1}, y_{j1}),\ldots, (u^*_{jt}, x^*_{jt}, y_{jt})\}$ (i.e., the equijoin of X_1 and $\{(v^*_1, y_1),\ldots,(v^*_N, y_N)\}$, in particular this result implies that there are t-v_j's such that $u^*_{j1}=v_{j1},\ldots, u^*_{jt}=v^*_{jt}$). To simplify the notation, denote this response table as $\{(u^*_1, x^*_1, y_1),\ldots, (u^*_t, x^*_t, y_t)\}$.

S_2 computes $\xi^*_i\leftarrow E(mpk, u^*_i, x^*_i\|M_0; r^*_i)$ where r^*_i's are independent randomness for i=1,…,t,

replaces arbitrary t ξ_i's with ξ_i*'s and keeps other N_1-t ξ_i's unchanged, making a new sequence denoted as $\xi'_1||\ldots||\xi'_{N1}$, computes $dmt^0 \leftarrow$FakeDmt(mpk_{TC}, $\xi'_1||\ldots||\xi'_{N1}$, λ, cmt^0). S_2 sends the message $\xi'_1||\ldots||\xi'_{N1}||dmt^0$ to A, interacts with A by calling IA-NMZPoK::Sim$_2$($\xi'_1||\ldots||\xi'_{N1}$, s) where ξ'_i=E(mpk, u^0_i, $x^0_i||M_0$; r'_i), i=1,…,N_1, $u^0_i=u^*_i=v^*_i$ and $x^0_i=x^*_i$ for i=1,…,t and $u^0_i=u_i$, $x^0_i=x_i$ for other i's.

Finally S_2 outputs whatever A outputs to the environment.

Let tr(S_2,A) denote the transcripts due to the interaction between A and S_2, $tr^{\Psi}(P_1(X_1),A)$ denote the transcripts due to the interaction between A and the real-world party $P_1(X_1)$(possessing the same table $X_1=\{(u^*_1,x_1^*),\ldots,(u_{N1}^*,x^*_{N1})\}$ as the ideal-world party P_1*). From A's perspective, the differences between these two transcripts are: *a*)*cmt* in these two transcripts are respectively cmt^0 output by FakeCmt and *cmt* output by Cmt(mpk_{TC}, P_2, E(mpk, u_1*, x_1*$||M_0$; r_1)||…||E(mpk, u^*_{N1}, $x^*_{N1}||M_0$; r_{N1})); *b*)*dmt* in these two transcripts are dmt^0 output by FakeDmt and *dmt* output by Cmt(mpk_{TC}, P_2, E(mpk, u_1*, x_1*$||M_0$; r_1)||…||E(mpk, u^*_{N1}, $x^*_{N1}||M_0$; r_{N1})) respectively *c*)Among the ciphertext sequence $\xi_1||\ldots||\xi_{N1}$ in these two transcripts, there are t ciphertexts ξ_i having the same identity public-key u^*_i and the same plaintext $x^*_i||M_0$ but the remaining N_1-t ciphertexts having different identity public-keys and plaintexts; *d*)there are t IA-NMZPoK-witness' with the same u^0_i and x^0_i.

Because of TC's equivocation property, (*cmt*,*dmt*)'s are P.P.T.-indistinguishable in both cases; because of IBE scheme's selective ANO_CPA anonymity and IND_CPA data-privacy, $\xi_1||\ldots||\xi_{N1}||dmt$ in both cases are P.P.T.-indistinguishable (otherwise suppose they are P.P.T.-distinguishable with the difference $\delta \geq 1/poly(k)$, it's not hard to construct either a selective ANO_CPA or IND_CPA attacker against Π with an advantage at least δ/N_1, contradicting with Π's either selective ANO_CPA anonymity or data-privacy). Now denote the ciphertext sequence $\xi_1||\ldots||\xi_{N1}$ in two cases as $\xi_1^{(1)}||\ldots||\xi_{N1}^{(1)}$ and $\xi_1^{(2)}||\ldots||\xi_{N1}^{(2)}$ respectively, denote the transcripts in session of IA-NMZPoK as IA-NMZPoK$^{(1)}$(=tr$_{S2,A}$($mpk||M_0||\ \xi_1^{(1)}||\ldots||\xi_{N1}^{(1)}$)) and IA-NMZPoK$^{(2)}$(=tr$_{P1,A}$($mpk||M_0\ ||\xi_1^{(2)}||\ldots||\xi_{N1}^{(2)}$))) respectively, by the above analysis we have $\xi_1^{(1)}||\ldots||\xi_{N1}^{(1)} \approx^{PPT} \xi_1^{(2)}||\ldots||\xi_{N1}^{(2)}$; furthermore, by IA-NMZPoK's zero-knowledge property we have

$$\text{IA-NMZPoK}^{(2)} \approx^{PPT} \text{IA-NMZPoK::Sim}_2(\xi_1^{(2)}||\ldots||\xi_{N1}^{(2)}, s)$$

and by S_2's construction we have

$$\text{IA-NMZPoK}^{(1)} = \text{IA-NMZPoK::Sim}_2(\xi_1^{(1)}||\ldots||\xi_{N1}^{(1)}, s)$$

so IA-NMZPoK$^{(1)} \approx^{PPT}$ IA-NMZPoK$^{(2)}$.

As a result, the transcripts received by A in both cases are P.P.T.-indistinguishable.

Let δ be $\Delta^{\Pi}_{\text{Blind-UKG}}$'s extractor's error function(negligible in k), then the probability with which S_2 correctly extracts A's one data-item y^*_i is at least P[A(mpk;P_1(mpk,msk))=UKG(msk,y^*_i)]-δ, so the probability with which S_2 correctly extracts A's all values $v^*_1,\ldots,v^*_N$ is at least P[A(mpk;P_1(mpk, msk))=UKG(msk,v^*_i): i=1,…,N]-$N_2\delta \geq$ P[P_2 outputs Join(w:X_1,X_2)]-Nδ. As a result, S_2's output is P.P.T.-indistinguishable from A's output in Ψ with respect to the GUC-environment Z with an error upper-bounded by $N_1(k)(\ Adv_{\Pi}^{ANO\_CPA}(k) + Adv_{\Pi}^{IND\_CPA}(k)\ )$+$N_2\delta$ ($N \leq N_2$) which is negligible in k. Note that in both cases the other party P_1*(X_1) and P_1(X_1) always output the same N, so we have the consequence that output$_Z$(ψ,A) $\approx^{PPT}$ output$_Z$(F_{Join},S_2) and it's easy to estimate S_2's time-complexity $T_{S2}=T_A+O(N_1+N_2T_{ext})$ where T_A and T_{ext} are A's and the extractor's computation-time.

By all the facts, we have $\Psi \rightarrow^{GUC} F_{Join}$.

APPENDIX.C PROOF OF THEOREM 4.1

All parties are assumed to be initialized with a copy of the common reference string *ACRS*, i.e., the concatenation of the two IA-NMZPoK protocol's $mpk_{ZK,II}$ and $mpk_{ZK,III}$. For this *ACRS*, $msk=msk_{ZK,II}||msk_{ZK,III}$ and UKG(*msk*,*id*) outputs $usk(id)=usk_{ZK,II}(id)||usk_{ZK,III}(id)$ where $usk_{ZK,II}(id)$ and $usk_{ZK,III}(id)$ are respectively two IA-NMZPoK protocol's user private-keys corresponding to $id \in \{P_1, P_2\}$.

At first it's easy to show there exists an identity extractor for $\Delta_{Blind-UKG}^{Boyen-Waters}$ to satisfy definition 3.1. In fact it is IA-NMZPoK$_{III}$((a, r_1, r_2, y_1, y_2, y_3, y_4): $\wedge_{i=1,2} U_i = g^{r_i}$ $\wedge_{i=1,2} V_i = (g_0 g_1^a)^{-r_i}$ $\wedge_{j=1,2,3,4} h_j = g^{y_j} g_1^a$)'s knowledge extractor for which the to-be-extracted witness is a.

Now we prove $\Delta_{Blind-UKG}^{Boyen-Waters}$'s GUC-security in two cases that the real-world adversary A corrupts P_1 or P_2 respectively. Below P_1* and P_2* stand for P_1 and P_2's respective counterparts in ideal-world.

(1) A corrupts P_1: Suppose A's(i.e., P_1's) private input is (*mpk*,*msk*), P_2*'s private input is a*. we need to construct an ideal adversary S_1. S_1 corrupts the ideal-world party P_1*, gets $usk(P_1)$ by querying G_{ACRS} with the message ("retrieve",sid,P_1) where $usk(P_1)=usk_{ZK,II}(P_1)||usk_{ZK,III}(P_1)$, computes $(\sigma_{II},s_{II},\tau) \leftarrow$ IA-NMZPoK$_{II}$::Ext$_1$($usk_{ZK,II}(P_1)$)(notice that P_1 is the prover in protocol IA-NMZPoK$_{II}$), runs A as a black-box. S_1 simulates the real-world honest party P_2 to interact with A:

In session of IA-NMZPoK$_{II}$(*msk*: *mpk*=ESetup(*msk*)), S_1 interacts with A as a verifier extracting *msk* via running IA-NMZPoK$_{II}$::Ext$_2$(τ), sends message (sid, *mpk*||*msk*) to $F_{Blind-UKG}^{Boyen-Waters}$;

S_1 generates an user-id a at random, follows P_2's specification in section 4.2 to compute U_1,U_2,V_1, V_2,h_1,h_2,h_3,h_4, sends $U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4$ to A, computes $(\sigma_{III},s_{III}) \leftarrow$ IA-NMZPoK$_{III}$:: Sim$_1$($usk_{ZK,III}(P_1)$)(notice that P_1 is the verifier in protocol IA-NMZPoK$_{III}$) and sends IA-NMZPoK$_{III}$:: Sim$_2$($U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4$,$s_{III}$) to A.

S_1 outputs whatever A outputs to the environment.

Denote the second-round message in $\Delta_{Blind-UKG}^{Boyen-Waters}$'s specification (i.e., $U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4$) as W. From A's perspective, the transcripts due to its interactions with S_1 and the transcripts due to its interactions with the real-world party $P_2(a^*)$($P_2(a^*)$ stands for party P_2 possessing a*, the same private input as the ideal-world party P_2*) differs in: *a*)W depends on a in the former case, denoted as $W(a)$, while it depends on a* in the latter case and denoted as $W(a^*)$; *b*)IA-NMZPoK$_{III}$'s witness depends on a in the former case while it depends on a* in the latter. The messages of subprotocol IA-NMZPoK$_{III}$ in these two cases are respectively denoted as IA-NMZPoK$_{III}$(a) and IA-NMZPoK$_{III}$(a*).

Let $g_0 \equiv g^{\alpha}$, $g_1 \equiv g^{\alpha^*}$. Explicitly expand $W(a)$'s expression to $g^{r_1} || g^{r_2} || g^{-(\alpha+a\beta)r_1} || g^{-(\alpha+a\beta)r_2} || g^{y_1+a\beta} || \ldots || g^{y_4+a\beta}$ and $W(a^*)$ to a similar expression where a, r_1, r_2, y_1, y_2, y_3, y_4, α and a*, r_1*, r_2*, y_1*, y_2*, y_3*, y_4*, α* are probabilistically independent and all are unknown to A, so $W(a) \approx W(a^*)$ (perfectly indistinguishable). Furthuremore, by IA-NMZPoK$_{III}$'s zero-knowledge property we have

$$\text{IA-NMZPoK}_{III}::\text{Sim}_2(W(a^*), s_{III}) \approx^{PPT} \text{IA-NMZPoK}_{III}(a^*)$$

and by S_1's construction we also have

$$\text{IA-NMZPoK}_{III}::\text{Sim}_2(W(a), s_{III}) = \text{IA-NMZPo}K_{III}(a)$$

so IA-NMZPoK$_{III}$(a) = IA-NMZPoK$_{III}$::Sim$_2$($W(a)$,s_{III}) $\approx$ IA-NMZPoK$_{III}$::Sim$_2$($W(a^*)$,s_{III}) $\approx^{PPT}$ IA-NMZPoK$_{III}$(a*). As a result, from A's perspective the transcripts due to its interactions with S_1 is

P.P.T.-indistinguishable from that due to its interactions with $P_2(a^*)$, in particular, the output of A due to its interactions with S_1 is P.P.T.-indistinguishable from its output due to its interactions with $P_2(a^*)$ in $\Delta^{Boyen-Waters}_{Blind-UKG}$.

Let η_{II} denote IA-NMZPoK$_{II}$'s knowledge extractor's error function(a negligible function in k), then the probability with which $P^*_2(a^*)$ outputs Π::UKG(msk,a^*) under S_1's attacks is at least P[P_2 accepts mpk as a valid master public-key]-η_{II}, i.e., except for an probability upper-bounded by η_{II}, $P^*_2(a^*)$'s output under S_1's attacks is the same as $P_2(a^*)$'s output under A's attacks, in other words, for any P.P.T. environment Z we have output$_Z(\Delta^{Boyen-Waters}_{Blind-UKG}, A_1) \approx^{PPT}$ output$_Z(F^{Boyen-Waters}_{Blind-UKG}, S_1)$ and it's easy to estimate S_1's time-complexity $T_{S1}=T_A+T_{eII}+O(1)$ where T_A and T_{eII} are A's and Ext$_{II,2}$'s computation-time.

(2)A corrupts P_2: Let a denote A's (i.e., P_2's) private input, (mpk^*,msk^*) denote the ideal-world party P_1^*'s input where $mpk^*=(G_1,G_2,p,e,\Omega^*,g,g_0,g_1,v_1^*,v_2^*,v_3^*,v_4^*)$ and $msk^*=(\omega^*,t_1^*,t_2^*,t_3^*,t_4^*)$. We need to construct an ideal-world adversary S_2 which corrupts P_2^*, gets $usk(P_2)$ by querying G_{ACRS} with the message ("retrieve",sid,P_2) where $usk(P_2)=usk_{ZK,II}(P_2)||usk_{ZK,III}(P_2)$, runs A as a black-box and simulates the honest real-world party P_1 to interact with A:

On receiving the message (sid,mpk^*) from $F^{Boyen-Waters}_{Blind-UKG}$, S_2 generates ω, t_1, t_2, t_3, t_4 at random and computes

$\Omega \leftarrow e(g,g)^{t_1 t_2 \omega}$; $v_1 \leftarrow g^{t1}$; $v_2 \leftarrow g^{t2}$; $v_3 \leftarrow g^{t3}$; $v_4 \leftarrow g^{t4}$;

$mpk \leftarrow (G_1, G_2, p, e, \Omega, g, g_0, g_1, v_1, v_2, v_3, v_4)$;

$msk \leftarrow (\omega, t_1, t_2, t_3, t_4)$;

$(\sigma_{II}, s_{II}) \leftarrow$ IA-NMZPoK$_{II}$:Sim$_1$($usk_{ZK,II}(P_2)$);

$(\sigma_{III}, s_{III}, \tau) \leftarrow$ IA-NMZPoK$_{III}$::Ext$_1$($usk_{ZK,III}(P_2)$);

Note that P_2 is the verifier in protocol IA-NMZPoK$_{II}$ and prover in IA-NMZPoK$_{III}$.

S_2 starts A and interacts with it by running IA-NMZPoK$_{II}$::Sim$_2$(mpk,s_{II});

When A sends $U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4$ and then launches IA-NMZPoK$_{III}$ ((a, r_1, r_2, y_1, y_2, y_3, y_4):…), S_2 participates the session as an verifier by running IA-NMZPoK$_{III}$::Ext$_2$(τ) to extract (a, r_1, r_2, y_1, y_2, y_3, y_4)(in fact only a is used below);

S_2 sends the message (sid||1,a) to $F^{Boyen-Waters}_{Blind-UKG}$ and gets the response (sid||1,UKG(msk^*,a)) where UKG(msk^*,a)≡(d_0^*, d_1^*, d_2^*, d_3^*, d_4^*);

S_2 generates d_j'' at random, computes $d_j' \leftarrow d_j^* / d_j''^{y_j}$, j=1,2,3,4, sends $d^*_0||d_1'||d_1''||d_2'||d_2''||d_3'||d_3''||d_4'||d_4''$ to A.

Now we prove that from A's perspective the transcripts due to its interactions with S_2 and that due to its interactions with P_1(mpk^*,msk^*)(a real-world party possessing the same input as the ideal-world party P_1^*) are P.P.T.-indistinguishable.

At first, consider the transcripts in IA-NMZPoK$_{II}$'s session. Let IA-NMZPoK$_{II}$(*) and IA-NMZPoK$_{II}$() denote the messages generated by P_1(mpk^*, msk^*) and S_2 in this session respectively. By IA-NMZPoK$_{II}$'s zero-knowledge property we have

IA-NMZPoK$_{II}$::Sim$_2$(mpk^*, s_{II}) $\approx^{PPT}$ IA-NM ZPoK$_{II}$(*)

and by S_2's construction we have

IA-NMZPoK$_{II}$::Sim$_2$(mpk, s_{II}) = IA-NMZPoK$_{II}$()

Let Ω_R denote a random element on group G_2. Since ω^*, ω, t_i^*, t_i (i=1,2,3,4) are probabilistically independent and all are unknown to A, from A's perspectiove we have

$mpk^* \equiv (G_1, G_2, p, e, \Omega^*, g, g_0, g_1, v^*_1, v^*_2, v^*_3, v^*_4)$

$\approx^{PPT}$ $(G_1, G_2, p, e, \Omega_R, g, g_0, g_1, v^*_1, v^*_2, v^*_3, v^*_4)$ (D-BDHP's hardness)

$\approx (G_1, G_2, p, e, \Omega_R, g, g_0, g_1, v_1, v_2, v_3, v_4)$ (trivial)

$\approx^{PPT}$ $(G_1, G_2, p, e, \Omega, g, g_0, g_1, v_1, v_2, v_3, v_4)$ (D-BDHP's hardness)

$\equiv mpk$

So IA-NMZPoK$_{II}$(*) $\approx^{PPT}$ IA-NMZPoK$_{II}$::Sim$_2$(mpk^*,s_{II}) $\approx^{PPT}$ IA-NMZPoK$_{II}$::Sim$_2$(mpk,s_{II}) = IA-NMZPoK$_{II}$().

Now consider the last-round message, which are $d^*_0||d_1'||d_1''||d_2'||d_2''||d_3'||d_3''||d_4'||d_4''$ and $d^*_0||d^*_1'||d^*_1''||d^*_2'||d^*_2''||d^*_3'||d^*_3''||d^*_4'||d^*_4''$ in these two cases(interacting with S_2 and with $P_1(mpk^*, msk^*)$) respectively. Both messages have the same component d^*_0, all other components are denoted as D and D^* respectively. Expanding D we get

$$D \equiv d_1^* / d_1''^{y_1} \| d_1'' \| d_2^* / d_2''^{y_2} \| d_2'' \| d_3^* / d_3''^{y_3} \| d_3'' \| d_4^* / d_4''^{y_4} \| d_4''$$

where d^*_1, d^*_2, d^*_3, d^*_4 come from UKG(msk^*,a), i.e., $d^*_1 = g^{-\varpi^* t_2^*} (g_0 g_1^{\ a})^{-\tilde{r}_1 t_2^*}$, $d^*_2 = g^{-\varpi^* t_1^*} (g_0 g_1^{\ a})^{-\tilde{r}_1 t_1^*}$, $d^*_3 = (g_0 g_1^{\ a})^{-\tilde{r}_2 t_4^*}$, $d^*_4 = (g_0 g_1^{\ a})^{-\tilde{r}_2 t_3^*}$.

Expanding D^* we get

$$D^* \equiv g^{-\varpi^* t_2^*} (h_1 g_0)^{-r'_1 t_2^*} V_1^{\sigma t_2^*} \| g^{r'_1 t_2^*} \| g^{-\varpi t_1^*} (h_2 g_0)^{-r'_1 t_1^*} V_1^{\sigma t_1^*} \| g^{r'_1 t_1^*} \| (h_3 g_0)^{-r'_2 t_4^*} V_2^{\sigma t_4^*} \| \| g^{r'_2 t_4^*} \| (h_4 g_0)^{-r'_2 t_3^*} V_2^{\sigma t_3^*} \| g^{r'_2 t_3^*}$$

where σ, $\tilde{r}_i$, r_i' and d_j" are probabilistically independent each other and unkown to A, σ, r_i' are generated by P_1, d_j" by S_2, $\tilde{r}_i$ by $F^{Boyen-Waters}_{Blind-UKG}$.

Since r_1' and r_2' are probabilistically independent each other, D^*'s 4 leftmost-components are probabilistically independent of those 4 rightmost-ones; note that t^*_1, t^*_2, t^*_3, t^*_4 are also probabilistically independent each other, we finally partition D^* into 4 independent components D_i^* as:

$D_1^* \equiv g^{-\varpi^* t_2^*} (h_1 g_0)^{-r'_1 t_2^*} V_1^{\sigma t_2^*} \| g^{r'_1 t_2^*}$ $\quad$ $D_2^* \equiv g^{-\varpi t_1^*} (h_2 g_0)^{-r'_1 t_1^*} V_1^{\sigma t_1^*} \| g^{r'_1 t_1^*}$

$D_3^* \equiv (h_3 g_0)^{-r'_2 t_4^*} V_2^{\sigma t_4^*} \| g^{r_2 t_4^*}$ $\quad$ $D_4^* \equiv (h_4 g_0)^{-r'_2 t_3^*} V_2^{\sigma t_3^*} \| g^{r'_2 t_3^*}$

Similarly partition D into 4 independent components D_i as:

$D_1 \equiv d_1^* / d_1''^{y_1} \| d_1''$ $\quad$ $D_2 \equiv d_2^* / d_2''^{y_2} \| d_2''$ $\quad$ $D_3 \equiv d_3^* / d_3''^{y_3} \| d_3''$ $\quad$ $D_4 \equiv d_4^* / d_4''^{y_4} \| d_4''$

The problem is now reduced to analysis on relationship between D_i and D^*_i. Consider $D_3^* \equiv (h_3 g_0)^{-r'_2 t_4^*} V_2^{\sigma t_4^*} \| g^{r'_2 t_4^*}$ and $D_3 \equiv d_3^* / d_3''^{y_3} \| d_3''$: obviously $D_3 \approx (h_3 g_0)^{-\tilde{r}_2 t_4^*} / g^{y_3 r'_2 t_4^*} \| g^{r'_2 t_4^*}$ so it's adequate to analyze the relationship between $(h_3 g_0)^{-r'_2 t_4^*} V_2^{\sigma t_4^*}$ and $(g_0 g_1^{\ a})^{-\tilde{r}_2 t_4^*} / g^{y_3 r'_2 t_4^*}$. Further note that $(h_3 g_0)^{-r'_2 t_4^*} \approx (h_3 g_0)^{-\tilde{r}_2 t_4^*}$, $V_2^{\sigma t_4^*} \approx g^{-y_3 r'_2 t_4^*}$, $(h_3 g_0)^{-\tilde{r}_2 t_4^*}$ and $g^{r'_2 t_4^*}$ are independent each other, so $D_3^* \approx D_3$. For the same reason $D_4^* \approx D_4$.

Consider $D_1^* \equiv g^{-\varpi^* t_2^*} (h_1 g_0)^{-r'_1 t_2^*} V_1^{\sigma t_2^*} \| g^{r'_1 t_2^*}$ and $D_1 \equiv d_1^* / d_1''^{y_1} \| d_1''$: obviously $D_1 \approx g^{-\varpi^* t_2^*} (g_0 g_1^{\ a})^{-\tilde{r}_1 t_2^*} / g^{r'_1 t_2^* y_1} \| g^{r'_1 t_2^*}$, by similar analysis as before we have $D_1^* \approx D_1$. For the same reason $D_2^* \approx D_2$. Therefore:

$$d^*_0||d_1'||d_1''||d_2'||d_2''||d_3'||d_3''||d_4'||d_4'' \approx d^*_0||d^*_1'||d^*_1''||d^*_2'||d^*_2''||d^*_3'||d^*_3''||d^*_4'||d^*_4''$$

In consequence, under the assumption of D-BDHP's hardness on J, from A's perspective the transcripts due to its interactions with S_2 and that due to its interactions with $P_1(mpk^*, msk^*)$ are P.P.T.-indistinguishable. In particular, A's output in the former case is P.P.T.-indistinguishable from its

output in the latter, the error is (by some straightforward calculation) upper-bounded by η_{III} +2 $Adv_J^{D-BDHP}(k)$ where η_{III} is IA-NMZPoK$_{III}$'s knowledge extractor's error function. As a result, for any P.P.T. environment Z we have output$_Z$($\Delta_{Blind-UKG}^{Boyen-Waters}$,A) $\approx^{PPT}$ output$_Z$($F_{Blind-UKG}^{Boyen-Waters}$,S_2) and it's easy to estimate S_2's time-complexity $T_{S2}=T_A+T_{eIII}+O(1)$ where T_A and T_{eIII} are A's and IA-NMZPoK$_{III}$'s extractor's computation-time.

Combining all consequences in the above, the theorem is finally proved.

APPENDIX.D IA-NMZPoK PROTOCOL'S CONSTRUCTION

D.1 (Dense) Ω-Protocol[8,14]

A Ω-protocol for a given relation R is a 3-move protocol in CRS model consisted of P.P.T. algorithms D, A, Z, Φ, Sim and Ext=(Ext$_1$,Ext$_2$). D is the CRS generating algorithm. All algorithms except D takes a CRS ω as one of its inputs. For some (x,w) s.t.R(x,w)=1 the common input for both the prover P and the verifier V is x and witness w is P's private input. In the first move P generates a randomness r, computes a←A(ω,x,w,r) and sends a to V; in the second move, V selects a challenge c at random and sends it back to P; then P computes z←Z(ω,x,w,r,c) and sends z to V in the last move; on receiving z, V outputs "*accept*" or "*refuse*" depending on whether Φ(ω,x,a,c,z)=1 or 0. In addition, a Ω-protocol has the following properties [14]:

(1) For the honest P which behaves under the above specification, Φ(ω,x,a,c,z)=1 is always true.

(2) Given c and $x \in L_R$ the simulator Sim(ω,x,c) can generate accepting transcripts with a distribution that is P.P.T.-indistinguishable from those when P and V execute the protocol on common input x while V selects c as the challenge.

(3) (σ,τ)←Ext$_1$(k) where σ is P.P.T.-indistinguishable from ω←D(k); in addition, if there exists two accepting transcripts (a, c, z) and (a, c', z') where $c \neq c$' for some given $x \in L_R$, then Ext$_2$(x, τ, (a, c, z)) outputs w such that R(x,w)=1.

A dense Ω-protocol has the additional property as follows [6]:

(4) The CRS-domain D is a subset of a larger domain, D*(named extended CRS-domain), which is an Abelian group and its group operations are all efficient. Furthermore, the element of D and D* is P.P.T.-indistinguishable from each other.

D.2 A General Construction for IA-NMZPoK Protocol

Now we present a general construction for IA-NMZPoK protocol(definition 2.3-2.4) for given relation R. It uses a secure (strong existential-unforgeable) one-time signature scheme, a secure IBTC sheme(definition 2.5) and a dense Ω-protocol as its components. Note that among these components the secure one-time signature scheme and IBTC scheme can all be efficiently constructed, only the Ω-protocol is related with the specific relation R, therefore the construction can be regarded as a general transformation from the (comparatively weak but easy to construct) Ω-protocol to the

IA-NMZPoK protocol.

This construction is similar as that in [14] and borrows the coin-tossing technique used in [5-6]. Given a binary relation R and its dense Ω-protocol Ω_R=(D,A,Z,Φ,Sim, Ext=(Ext_1,Ext_2)) with its CRS denoted as ω; SIG=(KGen,Sign,Vf) is a strong existential-unforgeable one-time signature scheme; IBTC=(D_{TC},Setup,UKG,Cmt,Vf,FakeCmt, FakeDmt) is a secure IBTC scheme with its master public/secret-key pair denoted as (mpk_{TC},msk_{TC}). The constructed protocol IA-NMZPoK$_R$ (see Figure D.1) is in the ACRS model and its ACRS is the IBTC scheme's master public-key mpk_{TC}.

For clearity, we use IBTC::Cmt to stand for IBTC scheme's commitment algorithm Cmt, SIG::Sign to stand for SIG scheme's signing algorithm Sign, etc. *P* and *V* denote the prover(P)'s and verifier(V)'s identities respectively. ξ denotes the protocol's transcripts excluding the signature, i.e., $\xi \equiv k_1||\omega_2||\omega_1||d_1||sig\_vk||cmt||c||a||dmt||z$. Actually the first 3-move session is an IBTC-based coin-tossing [5-6] to generate a CRS ω for the following protocol Ω_R and the second 3-move session is similar as the construction of NMZPoK protocol in [14].

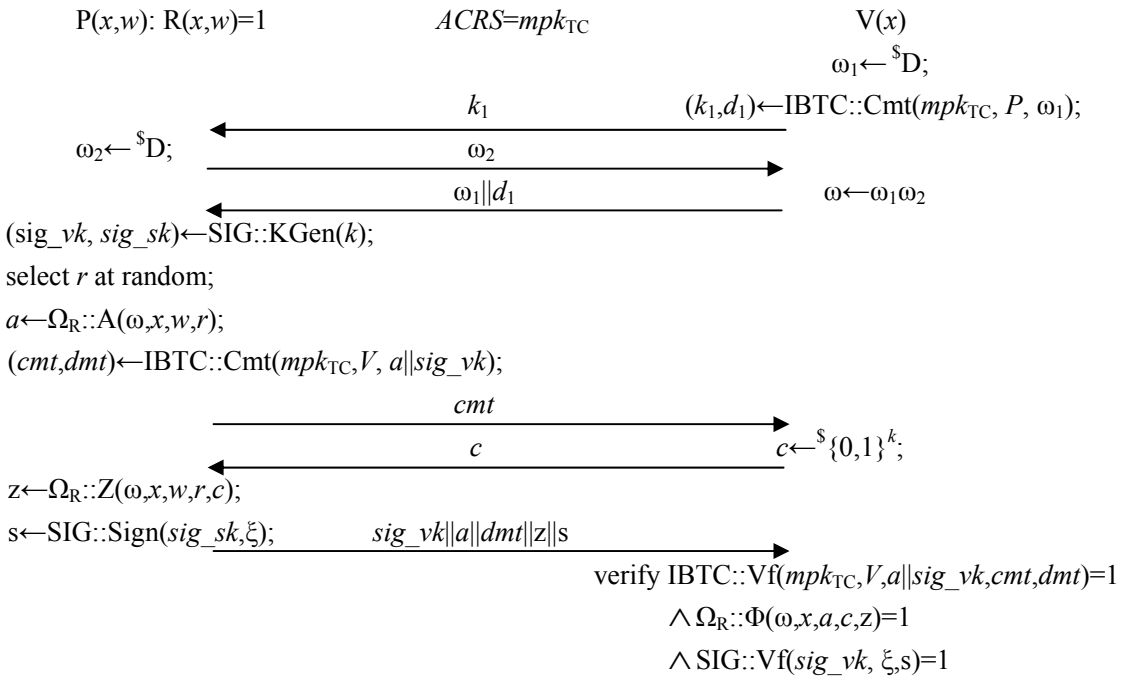


Figure D.1 IA-NMZPoK protocol IA-NMZPoK$_R$ for relation R.

Theorem D.1 IA-NMZPoK$_R$ is an IA-NMZPoK protocol for relation R.

Proof sketch The proof is similar as that of [14]'s theorem 4.1-4.2, the most difference is the simulation algorithm Sim=(Sim_1,Sim_2) and the extraction algorithm Ext=(Ext_1,Ext_2) which are presented here.

Let $usk(P) \equiv$IBTC::UKG(msk_{TC},*P*), $usk(V) \equiv$IBTC::UKG(msk_{TC},*V*). $Sim_1(usk(V))$ normally simulates the coin-tossing (the first 3-move session in IA-NMZPoK$_R$) as specified in the constrcution and its simulated transcript is denoted as $k_1||\omega_2||\omega_1||d_1$, then it outputs $k_1||\omega_2||\omega_1||d_1||usk(V)$. $Sim_2(mpk_{TC}, x, \omega, k_1||\omega_2||\omega_1||d_1||usk(V))$ (where $\omega=\omega_1\omega_2$) computes ($\overline{cmt}$,λ)←IBTC::FakeCmt(mpk_{TC},

V, $usk(V)$) and (sig_vk,sig_sk)←SIG::KGen(k), selects c at random, computes (a,z)←Ω_R::Sim(ω,x,c), $\overline{d}$ ←FakeDmt(mpk_{TC}, $a||sig\_vk$, λ,$\overline{cmt}$), s←SIG::Sign(sig_sk,ξ) where ξ is the whole transcript (as specified in the construction) excluding the signature s. Finally Sim_2 outputs

$$k_1||\omega_2||\omega_1||d_1||\overline{cmt}\,||c||sig\_vk||a||\overline{d}\,||z||s$$

For the extractor Ext=(Ext_1,Ext_2), Ext_1($usk(P)$) computes (ω,τ)←Ω_R::Ext_1(k) and outputs (ω, $usk(P)$||τ). Ext_2(mpk_{TC}, ω, $usk(P)$||τ) computes ($\overline{k_1}$,λ_1)←IBTC::FakeCmt(mpk_{TC},P, $usk(P)$) and sends $\overline{k_1}$ out; on receiving ω_2, it computes ω_1←ω/ω_2, $\overline{d_1}$ ←FakeDmt(mpk_{TC}, ω_1, λ_1,$\overline{k_1}$) and responses with $\omega_1||\overline{d_1}$; then it randomly generates a challenge c on receiving cmt. When it gets the last message $sig\_vk||a||dmt||z||s$, it checks all the required conditions and call Ω_R::Ext_2(ω, x, τ, $a||c||z$).

Now it can be shown that Sim=(Sim_1,Sim_2) and Ext=(Ext_1,Ext_2) indeed satisfy the properties in definition 2.3-2.4, the analysis is almost the same as in the proof of [14]'s theorem 4.1-4.2.