A UC/GUC-Secure Protocol for Set-Intersection Computation[1]

TIAN Yuan[1] and WANG Ying[2]

[1] Software School of Dalian University of Technology, Dalian, Liaoning, 116620,
tianyuan_ca@sina.com

[2] Department of Mathematics, Dalian University of Technology, Dalian, Liaoning, 116620,
wangying@dlut.edu.cn

Abstract *Secure set-intersection computation is one of important problems in the field of secure multiparty computation with valuable applications. We propose a general construction for 2-party set-intersection computation based-on anonymous IBE* (*identity-based encryption*) *scheme and its user private-keys blind generation techniques. Compared with recently-proposed set-intersection computation protocols, e.g., those of Freedman-Nissim-Pinkas*, *Kissner-Song and Hazay-Lindell, this construction is provably UC-secure in standard model with acceptable efficiency. After proving the general construction's UC-security, an efficient instantiation based-on the anonymous Boyen-Watrers IBE scheme is presented. We further enhance the UC-secure construction to be GUC-secure*(*in ACRS model*)*, for this goal a new notion of non-malleable zero-knowledge proofs of knowledge and its general construction is presented.*

Key words: Computer Security; Secure Set-Intersection Computation; Anonymous Identity-based Encryption; Universally Compossable Security; Generalized Universally Compossable Security.

1 Introduction

Secure set-intersection computation is one of important problems in the field of secure multiparty computation, with valuable applications in, e.g., secure keyword searching, pattern matching, private database processing, etc. In secure set-intersection computation, participants with their own private data sets get the intersection of all their private sets and nothing more(except for each private set's cardinality). In this paper, like most recent works, we are only focused on the 2-party case and make an efficient UC-secure, standard model protocol for it.

[1] *An extended abstract*(*with only main notions and consequences*) *is submitted to* TCC'09.

Much work has been done in designing solutions to secure computation for different cryptographic functions[1-2], but only few are about this special problem among which [3,6-7] are most relevant to our paper. They are heuristic and valuable works on secure set-intersection computation published most recently, each using different techniques and security concepts and most of them(except [7]) mainly dealing with the 2-party case. However, none reaches Canneti's UC/GUC security[14-15]. In [6] Freedman et al present provably-secure and efficient protocols for this problem against semi-honst and malicious adversaries respectively based-on polynomial interpolation and homomorphic encryption schemes. The solution against malicious adversaries assumes the random oracle model. [7] solves this problem (and more, e.g., union and element reduction operations) via smartly exploiting mathematical properties of polynormials and has fully-simulatable security so that their solution is securely compossable(the concept of fully-simulatable security can be refered in [2], however, this security is still weak than Canetti's concept of UC/GUC security proposed in [14-15]). In addition, as indicated by [3], [7] executes lots of zero-knowledge proofs of knowledge most of which are known how to efficiently realize but not all. Most recently [3] proposes solutions to this problem via oblivious pseudorandom function evaluation techniques. More interestingly, they work in two relaxed adversary models to achieve security of "half-simulatability" and full-simulatability against covert adversaries[4]. At the price of relaxation in security, the protocols presented in [3] are highly efficient, so these solutions can be considered as practical and reasonable compromise between security and efficiency.

1.1 **Our Contributions**

In this paper we construct a protocol for secure set-intersection computation in standard model which is efficient and secure under the concept of Canneti's UC/GUC security. Like most previous works, we are mainly focused on the 2-party case, however, there are substancial differences between our solution and the others. At first, our construction is based-on anonymous IBE scheme and it's user private-keys blind generation techniques(i.e., to generate the correct user private-key usk(a)=UKG(msk,a) for the user-id a but without knowing anything about a). Our protocol is constant-round in communications and linear-size in message-complexity (close to [3,6]). In computation-complexity, one party is $O(N_1+N_2)$ (close to [3,6]) and the other is $O(N_1N_2)$(close to [7]) where N_1, N_2 are each party's private set's cardinality. The construction is well-modularized, only executing few zero-knowledge proofs of knowledge which can be efficiently realized(we present examples in this paper).

Second, our construction reaches Canneti's UC/GUC security so that it is securely compossable. More concretely, we propose two versions of our construction, one is in the CRS model and UC-secure, another in the ACRS(augmented common reference string) model and GUC-secure. Although in general the notion of GUC-security is strictly stronger than that of UC-security, the two versions have the same structure with only differences in their zero-knowledge proofs of knowledge subprotocols. We present the UC-secure version and prove its security first and then systematically enhance its security to the GUC notion, to make things simpler and clearer. More importantly, since lots of UC-secure protocols are proposed and proved since the publication of [14] and most of them are in CRS model, we are interested in what can make a UC-secure protocol in CRS model become GUC-secure in ACRS model. For this goal, we introduce the concept of identity-based non-malleable zero-knowledge proofs of knowledge, present a general and efficient realization for this new concept and apply it to enhance our UC-secure construction to be GUC-secure. We believe such a method is valuable and helpful beyond the special problem in this paper.

1.2 Paper Organization

Section 2 presents some necessary concepts, preliminaries and tools. Section 3 presents the general construction based-on anonymous IBE and its user private-keys blind generation protocol. Section 4 instantiates the general construction via Boyen-Waters IBE scheme[12](in [12] two anonymous IBE schemes are proposed, one is ordinary IBE another is HIBE. We only use the ordinary IBE scheme for efficiency) together with a efficient construction of Boyen-Waters IBE's user private-keys blind generation protocol. Now there are only few provably-anonymous IBE schemes and our work shows again such IBE's importance[11-13][2]. Section 5 presents a systematic enhancement to make our UC-secure construction GUC-secure in ACRS model.

1.3 Some Terminologies and Notations

P.P.T. means "*probabilistic polynomial-time*", x||y means string x and y in concatenation, |x| means string x's size(in bits), $a \xleftarrow{\$} X$ means random selection of a sample over the domain X. When X is explicit or not important to discussions, a notation *νa*("*new a*") is also used.

k represents the complexity parameter, *poly*(k) means a given polynomial in *k*. $\approx^{PPT}$ means *computationally indistinguishable*, $\approx^{PDF}$ means *perfectly indistinguishable*. IND_CPA means security against chosen plaintext attacks and ANO_CPA means anonymity against chosen plaintext attacks.

2 Definitions and Tools

2.1 Secure Set-Intersection Computation and Its UC/GUC Security

Briefly speaking, UC/GUC-security means that any attacker against the real-world protocol can be simulated by an adversary against the ideal-world functionality, both have the outputs indistinguishable by the (malicious) environment. For space limitation, we assume the reader familiar with the whole theory in [14-15] and only make the necessary descriptions with respect to the secure set-intersection computation problem here.

Similar to most previous work, we are focused on the unidirectional 2-party scenario. The ideal cryptographic functionality for set-intersection computation is defined as

[2] IBE's user private-keys blind generation techniques are also used in [8], however all realizations they present are for non-anomynous IBE schemes so cannot be applied to our work directly. Interestingly, our construction in section 4 can be applied to their general framework as an addition.

$$F_{\mathrm{INT}}: (X_1,X_2) \rightarrow (|X_2|, |X_1|||(X_1 \cap X_2))$$

The bi-directional functionality is defined as

$$F^*_{\mathrm{INT}}: (X_1,X_2) \rightarrow (|X_2|||(X_1 \cap X_2), |X_1|||(X_1 \cap X_2))$$

and can be constructed by combining two F_{INT}'s in both directions.

Let P^*_1, P^*_2 be parties in ideal model with private sets 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",X_1) *from* P^*_1, F_{INT} *records* X_1 *and sends message* (sid,"input",N_1) *to* P^*_2 *and* S; *On receiving message* (sid,"input",X_2) *from* P^*_2, F_{INT} *records* X_2 *and sends* (sid,"input",N_2) *to* P^*_1 *and* S.

On receiving message (sid,"intersection") *from* P^*_2, F_{INT} *responses* P^*_2 *with message* (sid,"intersection", $X_1 \cap X_2$).

At last P_1^* *outputs* N_2, P_2^* *outputs* $N_1||(X_1 \cap X_2)$.

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 [14-15], 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 interactions with A and S, e.g., to give special inputs or instructions to A/S, collects outputs from A/S to make some analysis, etc. In UC theory [14], Z cannot access parties' shared functionality (such shared functionality is specified in specific protocol) while in the improved GUC theory [15] Z is enhanced to do this, i.e., to provide inputs to and get outputs from it. As a result, in GUC theory Z is strictly stronger and more realistic than in UC theory.

Let $\mathrm{output}_Z(\psi,A)$ denote the outputs (as one joint stochastic variable)from ψ's parties P_1, P_2 under Z and A, $\mathrm{output}_Z(F_{\mathrm{INT}},S)$ 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 $Z(\mathrm{output}_Z(\psi,A),u)$ and $Z(\mathrm{output}_Z(F_{\mathrm{INT}},S),u)$ respectively, where u is the auxiliary information. In the following we present the GUC-security's definition, however, when the environment Z is replaced with that in UC theory, it naturally becomes the concept of UC-security.

Defition 2.1(*GUC security*[15]) If for any (active) P.P.T. adversary A in real-world, there exists a P.P.T. adversary S in ideal-world, both corrupt the same party, such that for any environment Z the function $|P[Z(\text{output}_Z(\psi,A),u)=1]-P[Z(\text{output}_Z(F_{\text{INT}},S),u)=1]|$ is negligible in complexity parameter k (Hereafter denote this fact as $\text{output}_Z(\psi,A)\approx^{\text{PPT}}\text{output}_Z(F_{\text{INT}},S)$), then we define that ψ *GUC*-emulates F_{INT} or simply call that ψ is *GUC*-secure, denoted as $\psi\rightarrow^{\text{GUC}}F_{\text{INT}}$.

S is called A's simulator. In case of UC-emulation, we use the notation $\psi\rightarrow^{\text{UC}}F_{\text{INT}}$.

The most important and valuable property of the concept of GUC-emulation 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 $\varphi_2\rightarrow^{\text{GUC}}\varphi_1$ then(under some natural technical conditions, e.g., subroutine-respecting) $\psi(\varphi_2/\varphi_1)\rightarrow^{\text{GUC}}\psi(\varphi_1)$ where $\psi(\varphi_2/\varphi_1)$ is a protocol in which every call to its subprotocol φ_1 is replaced with a call to φ_2. Intuitively speaking, this guarantees a GUC-secure protocol can be composed in any execution context while still preserving its proved security. Similar consequence is also ture in UC theory but with some serious constraints. All details are presented in [14-15]。

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

In addition to data-privacy, anonymity(key-privacy) is another valuable property for public-key encryption schemes[11-13]. 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 as input (mpk,a,M) where M is the message plaintext to generate the ciphertext y=E(mpk,a,M), D takes as input (mpk,usk(a),y) to do decryption. Altogether these algorithms satisfy the consistency property: for any k, a and M

P[(mpk,msk)←Setup(k); usk(a)←UKG(msk,a); y←E(mpk, a, M): D(mpk, usk(a), y)=M]=1

Definition 2.2(*IBE Scheme's chosen plaintext anonymity*[11]) 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)←Setup(k);
> (M*, a_0*, a_1*, St)←$A_1^{UKG(\mathrm{msk},.)}(mpk)$, $a_0^*\neq a_1$;
> b$\xleftarrow{\$}$\{0,1\};
> y*←E(mpk, a_b*, M*);
> d←$A_2^{UKG(\mathrm{msk},.)}$(St,$y$*);
> output($d\oplus b$);

A is contrained not to query its oracle U(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, briefly called ANO_CPA. Denote $Adv_{\Pi}^{ANO\_CPA}(k)\equiv\sup_{A\in P.P.T.}Adv_{\Pi,A}^{ANO\_CPA}(k)$. In the above, if M* is generated independent of mpk then Π is called *selective* ANO_CPA.

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 provide it to $F^{\Pi}_{\text{Blind-UKG}}$; $F^{\Pi}_{\text{Blind-UKG}}$ generates usk(a)=UKG(msk,a) for another party who provides 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 functionality works as follows:

> P^*_1 *selects a seed* ρ *at random*, *computes* (mpk,msk)←Setup(ρ), *sends the message* (sid,ρ,mpk||msk) *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 sends 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)。

2.3 Non-malleable Zero-Knowledge Proofs of Knowledge and GMY/MY Techniques

We need the non-malleable zero-knowledge proofs protocol in our construction. This subsection presents this concept following [16-17] with small symbol 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 interaction with B under a public common input x and common reference string (c.r.s.) σ, $tr_{A,B}(x)_{[\sigma]}$ represents the transcript 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 $tr_{A(y),B}(x)_{[\sigma]}$ respectively. Let $A=(A_1,A_2)$, B and C be machines where A_1 can coordinate with A_2 by transferring state information to it, $(\langle B,A_1\rangle,\langle A_2,C\rangle)$ represents the interactions 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 $(\langle B,A_1\rangle,\langle A_2,C\rangle)$'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 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[16-17]) $ZPoK_R=(D_{crs},P,V,Sim=(Sim_1,Sim_2))$ 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 we no longer explicitly include σ in all the following expressions unless for emphasis. Now $ZPoK_R$ is defined as a *zero-knowledge proof protocol for relation R*(or equivalently for the language L), if the following properties are 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 $P[V(x;A)_{[\sigma]}=1]=0$[3];

(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 s as one of their inputs.

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

(4)There exists a negligible function $\eta(k)$(*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 transcripts 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 transcripts generated by $\boxed{Sim_2(s)}] - \eta(k)$.

---

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

It's easy to see that the above definition implies that $NMZPoK_R$ is a zero-knowledge proof of knowledge. In [16-17] Garay-MacKenzie-Yang developed an efficient method to derive non-malleable zero-knowledge proof protocol based-on simulation-sound tag-based commitment scheme and Ω-protocol(proposed in [17]). We'll apply this technique in section 4 to instantiate our general construction for private set-intersection computation protocol.

3 A General Protocol Construction for Private Set-Intersection Computation

Let Ψ denote the real-world private set-intersection computation protocol. Π=(Setup, UKG, E,D) is a selective ANO_CPA anonymous IBE scheme, $\Delta^{\Pi}{}_{\text{Blind-UKG}}$ is the real-world protocol for Π's user private-keys blind generation. Let *NMZPoK*(*w*:*R*(*x*,*w*)=1) denote a non-malleable zero-knowledge proof protocol for a *P*-relation R, where *w* is the witness. C=(Cmt, φ, FakeCmt, FakeDmt) is a non-interactive perfectly-hiding/computationally-binding equivocable commitment scheme[9], *H* is a collision-free hash function. Let P_1 and P_2 be two real-world parties with a public common plaintext M_0 as the c.r.s. The general construction is in figure 1(recall that the symbol "*v*ρ" means a random selection of ρ).

This Ψ is a $\Delta^{\Pi}{}_{\text{Blind-UKG}}$-hybrid protocol and we require $\Delta^{\Pi}{}_{\text{Blind-UKG}} \rightarrow^{\text{UC}} F^{\Pi}{}_{\text{Blind-UKG}}$. However, this first construction cannot guarantee UC-security but only "half UC-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 be always 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 Computation Protocol with Extractor*) given IBE scheme Π=(Setup,UKG,E,D) and $\Delta^{\Pi}{}_{\text{Blind-UKG}} \rightarrow^{\text{UC}} F^{\Pi}{}_{\text{Blind-UKG}}$, 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) provides usk(a) for P_2(as in figure 1). σ denotes $\varDelta^{\Pi}{}_{\text{Blind-UKG}}$'s c.r.s. This $\varDelta^{\Pi}{}_{\text{Blind-UKG}}$ is defined as *extractable*, if there exists P.P.T. algorithm $\text{Ext}_{\Pi}=(\text{Ext}_{\Pi,1}, \text{Ext}_{\Pi,2})$ and a negligible function $\delta(k)$, called the error function, such that 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):

(1) $\text{Ext}_{\Pi,1}(k)$ outputs (σ_0,τ) such that $\sigma_0 \overset{P.P.T.}{\approx} \sigma$;

(2) for any $(\sigma_0,\tau)\leftarrow\text{Ext}_{\Pi,1}(k)$: $P[\text{Ext}_{\Pi,2}(mpk||\tau\ ;A(a))_{[\sigma 0]}=a]>P[A_a(mpk;\ P_1(mpk,msk))_{[\sigma 0]}=\text{UKG}(msk,a)] - \delta(k)$ where (mpk,msk) is Π's master public/secret-keys owned by P_1 and a is $P^*{}_2$'s private input.

$P_1(X_1)$ M_0 is a public common plaintext $P_2(X_2)$

$v\rho$; $(mpk, msk)\leftarrow\text{Setup}(\rho)$;
for every data item x_i in X_1 compute
vr_i; $\xi_i\leftarrow E(mpk, x_i, M_0; r_i)$;
$h\leftarrow H(\xi_1||\ldots||\xi_{N1})$;
$(cmt, dmt)\leftarrow\text{Cmt}(h)$;

$\xrightarrow{\quad mpk \,||\, cmt \quad}$

input $y_1||\ldots||y_{N2}$ where y_j goes all elements in X_2

input(mpk, msk) $\longleftrightarrow$ $\boxed{\varDelta^{\Pi}{}_{\text{Blind-UKG}}}$ $\longleftrightarrow$

obtain N_2 obtain usk(y_1),…, usk(y_{N2})

$\xrightarrow{\quad \xi_1||\ldots||\xi_{N1}||dmt \quad}$

verify $\varphi(cmt,dmt,H(\xi_1||\ldots||\xi_{N1}))=1$
/*φ is commitment scheme C's verification function.*/

$\xrightarrow{\quad NMZPoK\,((x_i,r_i):\ \xi_i=E(mpk,\, x_i,\, M_0;\, r_i), i=1,\ldots,N_1) \quad}$

$X_0\leftarrow\{y_j\in X_2$: there exists ξ_i s.t. $D(mpk, \text{usk}(y_j), \xi_i)=M_0\}$

output(N_2) output($N_1||X_0$)

Figure 1 Anonymous IBE scheme(Π) based Unidirectional Secure Set-Intersection Computation Protocol Ψ (*NMZPoK*'s arrow points from the zero-knowledge proof's prover to verifier).

We stress that both extractors in definition 2.3(non-malleable zero-knowledge proof protocol) and definition 3.1 are *non-rewinding*, which is necessary for proving UC-security.

Combined with all the instantiations of subprotocols(presented in next section) in this general construction, it's easy to see that we can get constant-round and

$O(N_1+N_2)$ message-complexity solution to this problem. Furthermore P_1, P_2 has computation-complexity $O(N_1+N_2)$ and $O(N_1N_2)$ repectively. At last, all instantiated subprotocols(also UC-secure) are in CRS-model, so is Ψ which c.r.s is concatenation of M_0 and all subprotocols'c.r.s.'s

Theorem 3.1 *Suppose that* Π=(Setup,UKG,E,D) *is a selective* ANO_CPA *anonymous IBE scheme*, $\Delta^{\Pi}_{\text{Blind-UKG}} \rightarrow^{UC} F^{\Pi}_{\text{Blind-UKG}}$ *with extractor* $\text{Ext}_{\Pi}=(\text{Ext}_{\Pi,1}, \text{Ext}_{\Pi,2})$ *and error function* δ *as in def.3.1*, NMZPoK *is a non-malleable zero-knowledge proof protocol*, C=(Cmt,φ,FakeCmt,FakeDmt) *is a non-interactive perfect-hiding/P.P.T.-binding trapdoor commitment scheme*, *H is a collision-free hash function, then* $\Psi \rightarrow^{UC} F_{\text{INT}}$ *assuming static corruptions*.

Proof At first its easy to verify that Ψ produces the correct intersection $X_1 \cap X_2$. Now we prove UC-security in two cases that the real-world adversary *A* corrupts P_1 or P_2 respectively.

(1)*A* corrupts P_1: for simplicity we first make the proof in $F^{\Pi}_{\text{Blind-UKG}}$–hybrid model, then complete the proof by universal composition theorem. Let $X_1=\{x^*_1,\ldots,x^*_{N1}\}$ be *A*'s own set, $X_2=\{y^*_1,\ldots,y^*_{N2}\}$ be P^*_2's own set. We need to construct an ideal adversary S_1. S_1 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_{INT}, S_1 computes $(\sigma,s,\tau)\leftarrow$ *NMZPoK*::$\text{Ext}_1(k)$(to avoid ambiguity, we use Γ::*f* to represent a protocol Γ's function *f*), generates N_2 data-items $y_1,\ldots,y_{N2}$ at random and then starts $A(\sigma)$;

A sends *mpk*||*cmt*, S_1 interacts with *A* as an honest party in model of $F^{\Pi}_{\text{Blind-UKG}}$ and obtains usk(y_1),…,usk(y_{N2});

S_1 intercepts the message $\xi_1||\ldots||\xi_{N1}||dmt$ sent from *A*,verifys whether $\varphi(cmt,dmt,H(\xi_1||\ldots||\xi_{N1}))=1$, participates in the zero-knowledge protocol *NMZPoK*((x^*_i,r_i): $\xi_i=E(mpk, x^*_i, M_0; r_i)$, $i=1,\ldots,N_1$ as an honest verifier and calls the extractor *NMZPoK*::Ext_2(taking the trapdoor τ as one of its input) to extracts the witness (x^*_i,r_i), $i=1,\ldots,N_1$;

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

Let tr(A,S_1) denote the transcripts due to the interation between S_1 and A, tr^{ψ}(A, $P_2(X_2)$) denote the transcripts due to the interation between A and $P_2(X_2)$ in the real-world protocol Ψ(P_2 is the real-world party possessing the same private set X_2 as P^*_2). From A's perspective, the difference between tr(A,S_1) and tr^{ψ}(A, $P_2(X_2)$) is that the former provides $F^{\Pi}_{\text{Blind-UKG}}$ with $\{y_1,\ldots,y_{N2}\}$ as the input, the latter provides $F^{\Pi}_{\text{Blind-UKG}}$ with $\{y^*_1,\ldots,y^*_{N2}\}$, but according to $F^{\Pi}_{\text{Blind-UKG}}$'s specification, A knows nothing about what data-items are provided to $F^{\Pi}_{\text{Blind-UKG}}$ by the other party except the number N_2, as a result, $tr(A,S_1) \approx^{PDF} tr^{\psi}(A,P_2(X_2))$(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 *NMZPoK*'s error function, $Adv_C^{binding}$ be attacker's advantage against C's binding property, $Adv_H^{collision}$ be attacker's advantage against H's collision-free 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-items $\{x^*_1,\ldots,x^*_{N1}\}$is greater than $P[P_2(mpk||\xi_1||\ldots||\xi_{N1};A_1)=1]-N_1(\eta+Adv_C^{binding})-Adv_H^{collision} \geq P[X_0=X_1 \cap X_2] - N_1(\eta+Adv_C^{binding})-Adv_H^{collision}$, therefore, the difference between the probability with which $P^*_2(X_2)$ outputs $X_1 \cap X_2$ under the ideal-world adversary S_1 and the probability with which $P_2(X_2)$ outputs $X_1 \cap X_2$ under the real-world adversay A against Ψ is upper-bounded by $N_1(\eta+Adv_C^{binding})+Adv_H^{collision}$, also a negligible function in k. Combining all the above facts, for any P.P.T. environment Z we have $\text{output}_Z(\psi,A) \approx^{PPT} \text{output}_Z(F_{\text{INT}},S_1)$.

Now replace the ideal functionality $F^{\Pi}_{\text{Blind-UKG}}$ with $\Delta^{\Pi}_{\text{Blind-UKG}}$ in Ψ. By what is just proved, $\Delta^{\Pi}_{\text{Blind-UKG}} \rightarrow^{UC} F^{\Pi}_{\text{Blind-UKG}}$ and the universal composable theorem, we still have the above UC-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 extractor's computation time.

(2) A corrupts P_2: Denote A's own set as $X_2=\{y^*_1,\ldots,y^*_{N2}\}$, P^*_1's own set as $X_1=\{x^*_1,\ldots,x^*_{N1}\}$, we need to construct an ideal adversary S_2. S_2 corrupts P^*_2, generates (σ, s)←*NMZPoK*::$Sim_1(k)$, runs $A(\sigma)$ 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_{INT}, S_2 generates data-items

$x_1,\ldots,x_{N1}$ and a seed ρ at random, computes (mpk, msk)←Setup(ρ) and ξ_i←E(mpk, x_i, M_0; r_i) for every x_i where r_i is generated at random during the computation, computes (pk_C, cmt^0, π)←FakeCmt(k), starts A and sends the message $mpk||cmt^0$ to A;

S_2 interacts with A as an honest participant in $\Delta^{\Pi}{}_{\text{Blind-UKG}}$'s session and calls the extractor $\Delta^{\Pi}{}_{\text{Blind-UKG}}$::$\text{Ext}_{\Delta}$ to extract $y^*_1,\ldots,y^*_{N2}$, send message (sid,"input", $\{y^*_1,\ldots,y^*_{N2}\}$) to F_{INT};

S_2 sends (sid,"intersection") to F_{INT} and gets the response $\{y^*_{j1},\ldots,y^*_{jt}\}$(i.e., the set-intersection. To simplify the symbol, denote this response set as $\{y^*_1,\ldots,y^*_t\}$).

S_2 computes $vr^*_i.\xi^*_i$←E(mpk, y^*_i, M_0; r^*_i)(r^*_i's are selected at random)for i=1,…,t, replaces arbitrary t ξ_i's with ξ_i*'s and keeps other N_1-t ξ_i's unchanged, to get a new sequence denoted as $\xi'_1||\ldots||\xi'_{N1}$, computes dmt^0←FakeDmt(pk_C,π, H($\xi'_1||\ldots||\xi'_{N1}$)). S_2 sends the message $\xi'_1||\ldots||\xi'_{N1}||dmt^0$ to A, interacts with A in $NMZPoK$((x^0_i,r'_i): ξ'_i=E(mpk, x^0_i, M_0; r'_i),i=1,…,N_1)'s session as an honest prover, where $x^0_i = y^*_i$ for t i's and x^0_i=x_i for other i's.

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^{Ψ}($P_1(X_1)$,A) denote the transcripts due to the interaction between A and the real-world party $P_1(X_1)$(which owns the same data set $X_1=\{x_1^*,\ldots,x^*_{N1}\}$ 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 cmt^0 output by FakeCmt and cmt output by Cmt(H(E(mpk, x^*_1, M_0; r_1)||…||E(mpk,x^*_{N1}, M_0; r_{N1}))) respectively; b) dmt in these two transcripts are dmt^0 output by FakeDmt and dmt output by Cmt(H(E(mpk, x^*_1, M_0; r_1)||…||E(mpk,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 id public-key(i.e., x^*_i) but the remaining N_1-t ciphertexts having different id public-keys; d)there are t NMZPoK-witness' with the same x^0_i.

Because of C's perfect hiding property, (cmt,dmt) has the same distribution in both cases; because of IBE scheme Π's selective ANO_CPA property, $\xi_1||\ldots||\xi_{N1}||dmt$ in both cases are P.P.T.-indistinguishable(otherwise suppose they are P.P.T.-distinguishable with δ≥1/poly(k), it's easy to construct a selective ANO_CPA attacker against Π with an advantage at least δ/N_1, contradicting with Π's selective ANO_CPA anonymity). 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 *NMZPoK* as $NMZPoK^{(1)}(=tr_{S2(x1,\ldots,xN1),A}(mpk||M_0||\xi_1^{(1)}||\ldots||\xi_{N1}^{(1)}))$ and $NMZPoK^{(2)}(=tr_{P1(x*1,\ldots,x*N1),A}(mpk||M_0||\xi_1^{(2)}||\ldots||\xi_{N1}^{(2)})))$ respectively, by the above analysis we have $\xi_1^{(1)}||\ldots||\xi_{N1}^{(1)} \approx^{PPT} \xi_1^{(2)}||\ldots||\xi_{N1}^{(2)}$; by *NMZPoK*'s zero-knowledge property, we have

$$NMZPoK^{(1)} \approx^{PPT} NMZPoK::Sim_2(mpk||M_0||\xi_1^{(1)}||\ldots||\xi_{N1}^{(1)}, s)$$

and $$NMZPoK^{(2)} \approx^{PPT} NMZPoK::Sim_2(mpk||M_0||\xi_1^{(2)}||\ldots||\xi_{N1}^{(2)}, s)$$

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

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 $P[A(mpk;P_1(mpk,msk))=UKG(msk,y^*_i)]-\delta$, so the probability with which S_2 correctly extracts *A*'s all data-items $\{y^*_1,\ldots,y^*_{N2}\}$ is at least $P[A(mpk;P_1(mpk, msk))=UKG(msk,y^*_i): i=1,\cdots,N_2]-N_2\delta \geq P[X_0=X_1\cap X_2]-N_2\delta$. As a result, S_2's output is P.P.T.-indistinguishable from *A*'s output in Ψ with an error upper-bounded by $N_1(k)\,Adv_{\Pi}^{ANO\_CPA}(k)+N_2\delta$, also negligibale in *k*. Note that in both cases the other party $P^*_1(X_1)$ and $P_1(X_1)$ always output the same N_2, we have the consequence that $output_Z(\psi,A) \approx^{PPT} output_Z(F_{INT},S_2)$ and it's easy to estimate that 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 extractor's computation-time.

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

4 An Instantiation via Boyen-Waters IBE Scheme

Theorem 3.1 presents exact security conditions for general construction Ψ, among which some are available from existing works, e.g., the commitment scheme can be directly borrowed from the efficient scheme in [9]. In fact the subprotocols which require new efficient constructions are only IBE scheme Π's user private-keys generation protocol and the related non-malleable zero-knowledge proof protocol $NMZPoK((a,r){:}\ \xi=E(mpk,\ a,\ M_0;\ r))$. In this section we develop all these sub-constructions based-on Boyen-Waters IBE scheme to obtain an efficient instantiation of the general Ψ.

4.1 Boyen-Waters IBE[12]

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, $P\in G_1$, $e:G_1\times G_1\to G_2$ is a non-degenerate pairing, *Boyen-Waters* IBE consists of:

Setup(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), $a\in Z_p$:

$r_1, r_2\leftarrow^{\$}Z_p$;

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

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(ξ);

D(mpk, usk(a), $(\xi_{00}, \xi_0, \xi_1, \xi_2, \xi_3, \xi_4)$), 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($\xi_{00}T$);

[12] has proven that assuming the decisional bilinear Diffie-Hellman problem(D-BDHP)'s hardness on J, this scheme is selective IND_CPA secure(data-privacy); assuming the decisional linear problem(D-LP)'s hardness, this scheme is selective ANO_CPA anonymous. Note 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 $\Delta^{Boyen-Waters}_{Blind-UKG}$ and Its UC-Security

Figure 2 is the real-world user private-keys blind generation protocol for *Boyen-Waters* IBE scheme. For simplicity we only present how to blindly generate usk(a) for a single user-id a, but generalization for multiple user-id's $a_1||\ldots||a_N$ to blindly generate usk(a_1)||…||usk(a_N) is trival and still constant-round, though the total message-complexity is linearly increased.

P$_1$(mpk, msk) P$_2$($mpk||a$)

$\xrightarrow{NMZPoK_{II}(msk:\ mpk=\mathrm{Setup}(msk))}$ $v r_1 r_2 y_1 y_2 y_3 y_4$;

$U_i \leftarrow g^{r_i}$, $V_i \leftarrow (g_0 g_1^a)^{-r_i}$, i=1,2

$h_j \leftarrow g^{y_j} g_1^a$, j=1,2,3,4

$\xleftarrow{U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4}$

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

$\longleftarrow$

$v\sigma\, r_1'\, r_2'$; $d_0 \leftarrow (g^{r_1'} U_1^{\sigma})^{t_1 t_2}\ (g^{r_2'} U_2^{\sigma})^{t_3 t_4}$;

$d_1' \leftarrow g^{-\varpi t_2}\ (h_1 g_0)^{-r'_1 t_2} V_1^{\sigma t_2}$; $d_1'' \leftarrow g^{r_1' t_2}$;

$d_2' \leftarrow g^{-\varpi t_1}\ (h_2 g_0)^{-r'_1 t_1} V_1^{\sigma t_1}$; $d_2'' \leftarrow g^{r_1' t_1}$;

$d_3' \leftarrow (h_3 g_0)^{-r'_2 t_4} V_2^{\sigma t_4}$; $d_3'' \leftarrow g^{r_2' t_4}$;

$d_4' \leftarrow (h_4 g_0)^{-r'_2 t_3} V_2^{\sigma t_3}$; $d_4'' \leftarrow g^{r_2' t_3}$;

$\xrightarrow{d_0||d_1'||d_1''||d_2'||d_2''||d_3'||d_3''||d_4'||d_4''}$

$d_j \leftarrow d_j' d_j''^{y_j}$, j=1,2,3,4

output(d_0, d_1, d_2, d_3, d_4)

Figure 2 *Boyen-Waters* IBE's user private-key blind generation protocol $\Delta^{Boyen-Waters}_{Blind-UKG}$ (*NMZPoK*'s arrow points from zero-knowledge's prover to verifier)

$NMZPoK_{II}$ and $NMZPoK_{III}$ are two non-malleable zero-knowledge proof protocols for the specific relations. Since in *Boyen-Waters* scheme *msk* itself is used as the random seed in Setup so here we use a simpler expression Setup(*msk*).

It's easy to see by direct calculation that this protocol outputs the correct usk(a)=(d_0,d_1,d_2,d_3,d_4) where $d_0 = g^{(r_1'+r_1\sigma)t_1 t_2 + (r_2'+r_2\sigma)t_3 t_4}$, $d_1 = g^{-\varpi t_2}(g_0 g_1^a)^{-(r_1'+r_1\sigma)t_2}$, $d_2 = g^{-\varpi t_1}(g_0 g_1^a)^{-(r_1'+r_1\sigma)t_1}$, $d_3 = (g_0 g_1^a)^{-(r_2'+r_2\sigma)t_4}$, $d_4 = (g_0 g_1^a)^{-(r_2'+r_2\sigma)t_3}$).

Theorem 4.1 *If both* $NMZPoK_{II}$ *and* $NMZPoK_{III}$ *are non-malleable zero-knowledge proof protocols and the bilinear group pairing* J has D-BDHP *hardness*, *then* $\Delta^{Boyen-Waters}_{Blind-UKG} \rightarrow^{UC} F^{Boyen-Waters}_{Blind-UKG}$ *assuming static corruptions and* $\Delta^{Boyen-Waters}_{Blind-UKG}$ *satisfies definition* 3.1.

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

Now we prove $\Delta_{Blind-UKG}^{Boyen-Waters}$'s UC-security in two cases that the real-world adversary A corrupts P_1 or P_2 respectively.

(1)A corrupts P_1: Suppose A'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, generates $(\sigma,s,\tau) \leftarrow NMZPoK_{II}::Ext_1(k)$, runs $A(\sigma)$ as a black-box. S_1 simulates the real-world honest party P_2 to interact with A:

In session of $NMZPoK_{II}(msk:mpk=\mathrm{Setup}(msk))$ launched by A, S_1 interacts with A as an honest verifier, extracts msk via $NMZPoK_{II}::Ext_2$(taking τ as one of its inputs), and 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 fig.2 to compute $U_1,U_2,V_1,V_2,h_1,h_2,h_3,h_4$, sends the message $U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4$ to A, participates in $NMZPoK_{III}$ as an honest prover;

S_1 outputs whatever A outputs to the environment.

Let $W \equiv U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4$. 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 $P_2(a^*)$(possessing the same private input as the ideal-world party P^*_2) differs in a)W depends on a in the former case while depends on a^* in the latter; b)$NMZPoK_{III}$'s witness depends on a in the former case while depends on a^* in the latter.

Let $W(a)$, $NMZPoK_{III}(a)$ and $W(a^*)$, $NMZPoK_{III}(a^*)$ denote protocol-messages in these two cases respectively. Let $g_0 \equiv g^{\alpha}$, $g_1 \equiv g^{\beta}$, expand $W(a)$ 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, a^*, r_1, r_2, y_1, y_2, y_3, y_4, α and β are probabilistically independent and all are unknown to A, so $W(a) \approx^{PDF} W(a^*)$; by $NMZPoK_{III}$'s zero-knowledge property, there exists $NMZPoK_{III}$'s simulator such that

$$NMZPoK_{III}::Sim_2(W(a), s) \approx^{PPT} NMZPoK_{III}(a)$$

and $$NMZPoK_{III}::Sim_2(W(a^*), s) \approx^{PPT} NMZPoK_{III}(a^*)$$

so $NMZPoK_{III}(a) \approx^{PPT} NMZPoK_{III}::Sim_2(W(a), s) \approx^{PDF} NMZPoK_{III}::Sim_2(W(a), s) \approx^{PPT} NMZPoK_{III}(a^*)$. As a result, from A's perspective the transcripts due to its interactions with S_1 has the same distribution as that due to its interactions with $P_2(a^*)$, in particular, the output of A due to its interactions with S_1 has the same distribution as its output due to its interactions with $P_2(a^*)$ in $\Delta_{Blind-UKG}^{Boyen-Waters}$.

Let η_{II} denote $NMZPoK_{II}$'s knowledge extractor's error function(a negligible function in k)，then the probability with which $P^*_2(a^*)$ outputs 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$_Z$($\Delta_{Blind-UKG}^{Boyen-Waters}$,A_1) $\approx^{PPT}$ output$_Z$($F_{Blind-UKG}^{Boyen-Waters}$,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 (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. S_2 corrupts P^*_2, runs A as a black-box, simulates the honest real-world party P_1 to interact with A:

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

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

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

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

$(\sigma,s,\tau) \leftarrow NMZPoK_{III}::Ext_1(k)$;

S_2 starts $A(\sigma)$ and launches $NMZPoK_{II}$(msk:mpk=Setup(msk)) in role of an honest prover.

When A sends $U_1||U_2||V_1||V_2||h_1||h_2||h_3||h_4$ and then launches $NMZPoK_{III}$ ((a, r_1, r_2, y_1, y_2, y_3, y_4):…), S_2 participates the session as an honest verifier and calls $NMZPoK_{III}$::Ext_2(taking τ as one of its input) to extract (a, r_1, r_2, y_1, y_2, y_3, y_4);

S_2 sends the message (sid||1,a) to $F_{Blind-UKG}^{Boyen-Waters}$ 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 $NMZPoK_{II}$'s session. Let $NMZPoK_{II}(*)$ and $NMZPoK_{II}()$ denote the messages generated by $P_1(mpk^*, msk^*)$ and S_2 in this session respectively. By $NMZPoK_{II}$'s zero-knowledge property, there exists the P.P.T.-simulator such that

$$NMZPoK_{II}::\mathrm{Sim}_2(mpk^*, s) \approx^{PPT} NMZPoK_{II}(*)$$

and

$$NMZPoK_{II}::\mathrm{Sim}_2(mpk, s) \approx^{PPT} NMZPoK_{II}()$$

Let Ω_R denote a random element on group G_2. Since $\omega^*, \omega, t_i^*, t_i$ (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 hard)

$\approx^{PDF}$ $(G_1, G_2, p, e, \Omega_R, g, g_0, g_1, v_1, v_2, v_3, v_4)$ (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 hard)

$\equiv mpk$

So $NMZPoK_{II}(*) \approx^{PPT} NMZPoK_{II}::\mathrm{Sim}_2(mpk^*, s) \approx^{PPT} NMZPoK_{II}::\mathrm{Sim}_2(mpk, s) \approx^{PPT} NMZPoK_{II}()$.

Now consider the last 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 have

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

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 have

$$D^* \equiv g^{-\varpi^* t_2^*} (h_1 g_0)^{-r'_1 t_2^*} V_1^{\sigma t_2^*} \| g^{r_1' t_2^*} \| g^{-\varpi t_1^*} (h_2 g_0)^{-r'_1 t_1^*} V_1^{\sigma t_1^*} \| g^{r_1' t_1^*} \| (h_3 g_0)^{-r'_2 t_4^*} V_2^{\sigma t_4^*} \| \\ \| g^{r_2' t_4^*} \| (h_4 g_0)^{-r'_2 t_3^*} V_2^{\sigma t_3^*} \| g^{r_2' t_3^*}$$

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

Since r_1' and r_2' are probabilistically independent, 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, we finally partition D^* into 4 independent components D_i^* as:

$D_1^* \equiv g^{-\varpi^* t_2^*}(h_1g_0)^{-r'_1t_2^*}V_1^{\sigma t_2^*} \| g^{r'_1t_2^*}$ $\quad$ $D_2^* \equiv g^{-\varpi t_1^*}(h_2g_0)^{-r'_1t_1^*}V_1^{\sigma t_1^*} \| g^{r'_1t_1^*}$

$D_3^* \equiv (h_3g_0)^{-r'_2t_4^*}V_2^{\sigma t_4^*} \| g^{r'_2t_4^*}$ $\quad$ $D_4^* \equiv (h_4g_0)^{-r'_2t_3^*}V_2^{\sigma t_3^*} \| g^{r'_2t_3^*}$

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 reduced to analysis about relationship between D_i and D^*_i. Consider $D_3^* \equiv (h_3g_0)^{-r'_2t_4^*}V_2^{\sigma t_4^*} \| g^{r'_2t_4^*}$ and $D_3 \equiv d_3^*/d_3^{''y_3} \| d_3^{''}$: obviously $D_3 \approx^{\text{PDF}} (h_3g_0)^{-\tilde{r}_2t_4^*}/g^{y_3r'_2t_4^*} \| g^{r'_2t_4^*}$ so it's adequate to analyze the relationship between $(h_3g_0)^{-r'_2t_4^*}V_2^{\sigma t_4^*}$ and $(g_0g_1{}^a)^{-\tilde{r}_2t_4^*}/g^{y_3r'_2t_4^*}$. Further note that $(h_3g_0)^{-r'_2t_4^*} \approx^{\text{PDF}} (h_3g_0)^{-\tilde{r}_2t_4^*}$, $V_2^{\sigma t_4^*} \approx^{\text{PDF}} g^{-y_3r'_2t_4^*}$, $(h_3g_0)^{-\tilde{r}_2t_4^*}$ and $g^{r'_2t_4^*}$ are independent each other, so $D_3^* \approx^{\text{PDF}} D_3$. For the same reason $D_4^* \approx^{\text{PDF}} D_4$.

Consider $D_1^* \equiv g^{-\varpi^* t_2^*}(h_1g_0)^{-r'_1t_2^*}V_1^{\sigma t_2^*} \| g^{r'_1t_2^*}$ and $D_1 \equiv d_1^*/d_1^{''y_1} \| d_1^{''}$: obviously $D_1 \approx^{\text{PDF}} g^{-\varpi^* t_2^*}(g_0g_1{}^a)^{-\tilde{r}_1t_2^*}/g^{r'_1t_2^*y_1} \| g^{r'_1t_2^*}$, by similar analysis as before we have $D_1^* \approx^{\text{PDF}} D_1$. For the same reason $D_2^* \approx^{\text{PDF}} D_2$. Therefore:

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

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 trivial calculation)upper-bounded by $\eta_{III} + 2\,Adv_J^{D-BDHP}(k)$ where η_{III} is *NMZPoK*$_{III}$'sextractor's error function. As a result, for any P.P.T. environment Z we have $\text{output}_Z(\Delta_{Blind-UKG}^{Boyen-Waters}, A_2) \approx^{\text{PPT}} \text{output}_Z(F_{Blind-UKG}^{Boyen-Waters}, S_2)$ 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 *NMZPoK*$_{III}$'s extractor's computation-time.

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

4.3 Non-Malleable Zero-Knowledge Proof Protocols' Construction

The critical components in figure-1 and figure-2 are three non-malleable zero-knowledge proof protocols *NMZPoK*, *NMZPoK*$_{II}$ and *NMZPoK*$_{III}$. We apply *GMY/MY* techniques[16-17] to make our solutions. All contructions are constant-round and highly efficient. Note that by a general theorem proven in [16], the non-malleable zero-knowledge proof protocol UC-emulates the ideal zero-knowledge proof functionality, and that's why our method can be successful to make Ψ UC-secure and this feature can be even preserved when we develop the constructions for GUC-secure Ψ.

4.3.1 Tools: Paillier Scheme, Ω-Protocol and *GMY*/*MY* Techniques

Some powerful tools are required. At first, we use a *Paillier* scheme revised by Damgard and Catalano et al in [18-19]. Let $N=p_1p_2$ be a RSA modular, $s<p_1, p_2$, [18] proved that the order of $1+N$ modulo N^{s+1} is N^S(*Paillier*'s original scheme has s=1). [19]-revised scheme is: public-key pk=RSA public-key (e,N), private-key $sk=d$ where $ed=1 \bmod \varphi(N)$, for plaintext $m\in Z_N$ the encryption $E(pk, m)=(1+mN)r^e \bmod N^2$ where $r\leftarrow^{\$} Z^*_N$, the decryption on y is $r\leftarrow y^d \bmod N$ and then $m\leftarrow((r^{-e}y - 1)\bmod N^2) / N$. [19] proved that this scheme is IND_CPA secure under the decisional e-residues hardness. All details can be found in [18-19] and note that this scheme is homomorphic such that $E(pk, m_1)E(pk, m_2)=E(pk, (m_1+m_2)\bmod N) \bmod N^2$.

The second tool is the honest verifier zero-knowledge Ω-protocol proposed in [17] and how to transform a Ω-protocol for relation R via the simulation-sound trapdoor commitment scheme into a non-malleable zero-knowledge proof protocol for R. The reader can refer [16-17](particularly [16]'s theorem 4.2)for all details.

Definition 4.1(Ω-protocol for relation R[17]) R's Ω-protocol $\Omega^R=(D_{crs},A,Z,\Phi,Sim, Ext=(Ext_1,Ext_2))$ is a group of P.P.T. algorithms, where $D_{crs}(k)$ generates c.r.s. σ and all other algorithms take σ as one of their inputs(so σ is no longer explicitly expressed in these algorithms' inputs unless for emphasis); A, Z, Φ are prover's and verifier's algorithms. The protocol's structure is as follows:

P(x,*w*): R(x,*w*)=1	c.r.s. σ	V(x)
vr; $a\leftarrow A(x,w,r)$;	a $\longrightarrow$	
	c $\longleftarrow$	vc;
$z\leftarrow Z(x,w,r,c)$;	z $\longrightarrow$	
		verify $\Phi(x,a,c,z)=1$

The simulator Sim(x,c) generates (a*, z*) for given c and x$\in L_R$ such that for the transcript (a,c,z) between honest prover P(x,w) and verifier V it is true that (a*,c,z*)$\approx^{\text{P.P.T.}}$(a,c,z).

For P.P.T. extractor Ext=(Ext_1,Ext_2)), $Ext_1(k)$ generates (σ_1,τ) such that $\sigma_1 \approx^{\text{P.P.T.}} \sigma$, τ is called *extractor trapdoor*. Ext_2 can always extracts some thing, however, if there exist two transcripts (a,c,z) and (a,c',z') accepted by V but c≠c'(but the first messages are the same a), i.e., $\Phi(\sigma_1,x,a,c,z)=\Phi(\sigma_1,x,a,c',z')=1$, then x$\in L_R$ and $Ext_2(\sigma_1,x,\tau,(a,c,z))$ generates a witness w: R(x,w)=1. We stress that Ext_2 doesn't rewind P which is a significant feature in Ω-protocol.

Given relation R and its Ω-protocol Ω_R, *GMY/MY* techniques transform Ω_R into R's non-malleable zero-knowledge proof protocol via the following construction in figure 3, where SIG_1 is a one-time signature scheme, TC is a non-interactive simulation-sound tag-based trapdoor commitment scheme, Cmt and Vf are TC's commiting and verifying algorithm, pk is TC's public-key, A、Z、Φ are algorithms of Ω_R in definition 4.1. The protocol's c.r.s. is σ||pjk.

P(x,w): R(x,w)=1 c.r.s. σ||pk V(x)
(vk,sk)←SIG_1::KGen(k);
vr; a←A(σ,x,w,r);
(cmt,dmt)←TC::Cmt(pk, vk, a); vk||cmt →
← c vc;
z←Z(σ,x,w,r,c); s←SIG_1::Sign(sk,vk||cmt||c||a||dmt||z);
a||dmt||z||s → verify TC::Vf(pk, vk, a, cmt,dmt)=1
∧Φ(σ,x,a,c,z)=1∧SIG_1::Vf(vk,vk||cmt||c||a||dmt||z,s)=1

Figure 3 Transformation from Ω_R into $NMZPoK_R$[16]

In figure 3 Ω_R is the only R-specific constituent. Other constituents can all be borrowed from existing works, e.g., some efficient constructions for TC are given in [16-17](but not Ω-protocols). As a result, our efficient instantiation is finally reduced to the efficient constructions for those mathematical relations in fig.1 and fig.2 with respect to Boyen-Waters IBE scheme[4].

4 Theoretically it's also feasible to apply non-interactive non-malleable zero-knowledge proof schemes to

4.3.2 Constructing *NMZPoK*, *NMZPoK*$_{\mathrm{II}}$ and *NMZPoK*$_{\mathrm{III}}$

In case of Boyen-Waters scheme NMZPoK$_{\mathrm{II}}$(*msk*: *mpk*=Setup(*msk*)) is

$$\mathrm{NMZPoK}((\omega, t_1, t_2, t_3, t_4): \Omega = e(g,g)^{t_1 t_2 \omega} \wedge v_1 = g^{t1} \wedge v_2 = g^{t2} \wedge v_3 = g^{t3} \wedge v_4 = g^{t4})$$

Note that $\Omega = e(g,g)^{t_1 t_2 \omega} = e(v_1, v_2)^{\omega}$ so the desired protocol is equivalent to

$$\mathrm{NMZPoK}((\omega, t_1, t_2, t_3, t_4): \Omega = e(v_1, v_2)^{\omega} \wedge v_1 = g^{t1} \wedge v_2 = g^{t2} \wedge v_3 = g^{t3} \wedge v_4 = g^{t4}) \quad (4\text{-}1)$$

Now we analyze how to construct

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

Observe that (the pairing e is non-degenerate and G_1, G_2 are both prime-order) $V_i = (g_0 g_1^a)^{-r_i}$ *iff* $e(g, V_i) = e(g^{r_i}, g_0 g_1^a)^{-1} = e(U_i, g_0 g_1^a)^{-1} = e(U_i, g_0)^{-1} e(U_i, g_1)^{-a}$, i.e.,

$$e(g, V_i)\, e(U_i, g_0) = e(U_i, g_1)^{-a}, i=1,2$$

$h_j = g^{y_j} g_1^a$ *iff* $e(U_1, g_0 h_j) = e(U_1, g_0 g_1^a)\, e(U_1, g)^{y_j} = e(g, V_1)^{-1} e(U_1, g)^{y_j}$, i.e.,

$$e(U_1, g_0 h_j)\, e(g, V_1) = e(U_1, g)^{y_j}, j=1,2,3,4$$

The above expression is also true if U_2 replaces U_1. Denote publicly-computable items $F_i \equiv e(g, V_i)\, e(U_i, g_0)$, $f_i \equiv e(U_i, g_1)^{-1}$, $H_j \equiv e(U_1, g_0 h_j)\, e(g, V_1)$, $h \equiv e(U_1, g)$, then NMZPoK$_{\mathrm{III}}$ becomes

$$\mathrm{NMZPoK}((a, r_1, r_2, y_1, y_2, y_3, y_4): \wedge_{i=1,2} U_i = g^{r_i} \wedge_{i=1,2} F_i = f_i^{\,a} \wedge_{j=1,2,3,4} H_j = h^{y_j})$$

A further observation tells that $F_1 = f_1^{\,a}$ and $F_2 = f_2^{\,a}$ are not independent: in fact, let $F_1 = f_1^{a_1}$ and $F_2 = f_2^{a_2}$ then via bilinear pairing we have $e(f_1, F_2) = e(f_1, f_2)^{a_2}$ and $e(F_1, f_2) = e(f_1, f_2)^{a_1}$, i.e., $e(f_1, F_2) = e(F_1, f_2)$ *iff* $a_1 = a_2$ so one statement of $F_1 = f_1^{\,a}$ or $F_2 = f_2^{\,a}$ can imply another one by publicly checking $e(f_1, F_2) = e(F_1, f_2)$. Therefore the desired NMZPoK$_{\mathrm{III}}$ is equivalent to

$$\mathrm{NMZPoK}((a, r_1, r_2, y_1, y_2, y_3, y_4): \wedge_{i=1,2} U_i = g^{r_i} \wedge F_1 = f_1^{\,a} \wedge_{j=1,2,3,4} H_j = h^{y_j}) \quad (4\text{-}2)$$

Now analyze NM*ZPoK*((a,r): ξ=E(*mpk*, a, M_0; r)). In case of *Boyen-Waters* scheme, denote the public common plaintext as M_0 and the scheme's ciphertext as $\xi \equiv (\xi_{00}, \xi_0, \xi_1, \xi_2, \xi_3, \xi_4)$, then ZPoK(($a$,r): ξ=E(*mpk*, a, M_0; r)) becomes ZPoK((a,s,s_1,s_2): $\xi_{00} = \Omega^s M_0 \wedge \xi_0 = (g_0 g_1^{\,a})^s \wedge \xi_1 = v_1^{s-s1} \wedge \xi_2 = v_2^{s1} \wedge \xi_3 = v_3^{s-s2} \wedge \xi_4 = v_4^{s2}$). Because in theorem 3.1's proof what is needed is just the witness a, with respect to protocol Ψ it's adequate to construct *NMZPoK*((a,s): $\xi_{00} = \Omega^s M_0 \wedge \xi_0 = (g_0 g_1^{\,a})^s$).

---

instantiate our UC/GUC-secure constructions, however, so far we don't know how to construct such non-interactive schemes for the desired relations in case of Boyen-Waters IBE(Groth et al's work published at Eurocrypt'08 cannot be directly applied here, their schemes are witness indistinguishable in general, only zero-knowledge in some special conditions).

Note that ξ_{00}, Ω, $M_0 \in G_1$ and $\xi_0, g_0, g_1 \in G_2$. If $G_1=G_2$ then by $e(\Omega,\xi_0)=e(\Omega^s,g_0)e(\Omega^s, g_1)^a$ it's easy to see that the desired protocol is equivalent to $NMZPoK((a,s)$: $\xi_{00}M_0^{-1}=\Omega^s \wedge e(\Omega,\xi_0)e(\xi_{00}M_0^{-1},g_0)^{-1}=e(\xi_{00}M_0^{-1}, g_1)^a$), in the same form as (4-1) and (4-2). Unfortnately, in general G_1 and G_2 are not the same group, e.g., G_1 is usually a prime-order subgroup on elliptic curve while G_2 is a multiplicative subgroup in some finite field, so new approach is needed. In fact, denote $\chi_{00} \equiv \xi_{00}M_0^{-1}$, $t \equiv as$, then $\chi_{00}=\Omega^s$, $\xi_0=(g_0 g_1{}^a)^s = g_0^s g_1^t$ and it's easy to see that $NMZPoK((a,s)$: $\xi_{00}=\Omega^s M_0 \wedge \xi_0=(g_0 g_1{}^a)^s)$($a = ts^{-1} \bmod q$) is equivalent to:

$$\text{NMZPoK}((s,t)\text{: } \chi_{00}=\Omega^s \wedge \xi_0= g_0^s g_1^t) \qquad (4\text{-}3)$$

So far all desired non-malleable zero-knowledge proof protocols are explicitly presented and all relations in them can be unified to a group of linear exponent equations on prime-order group G in (4-4)(more generally each equation in (A-4) can be on a different group, but this case can be processed by a trivial generalization of the uniform case in which all equations are on the same group, so we only deal with the latter):

$$\prod_{j=1}^{n} B_{ij}^{x_j} = h_i \text{ i}=1,\ldots,m \qquad (4\text{-}4)$$

where B_{ij} and h_i are in G and x_i's are the integer witness. By *GMY/MY* techniques it's adequate to construct (4-4)'s efficient Ω-protocols. For simplicity but w.l.o.g., we present a Ω-protocol only for (4-3), i.e., the relation $\chi_{00}=\Omega^s \wedge \xi_0= g_0^s g_1^t$, in figure 4.

$|G|=q$,q is prime, the Ω-protocol has a RSA modular N as its c.r.s. where $N=p_1p_2$ and q can divide neither p_1-1 nor p_2-1(e.g., p_1, $p_2>4q$). Note that every e_{ij} is a *Paillier* ciphertexts. The simulator Sim(N,c) is specified as follows and it's easy to verify that Sim(N,c) has zero-knowledge simulation property specified in definition 4.1:

$z_{11}, z_{12} \xleftarrow{\$} Z_q$; $z_{21}, z_{22} \xleftarrow{\$} Z^*_N$; $e_{11}, e_{21} \xleftarrow{\$} Z^*_{N^2}$;

$\Theta \leftarrow \xi_0^{-c}\, g_0^{z_{11}} g_1^{z_{12}}$; $U \leftarrow \chi_{00}^{-c}\, \Omega^{z_{11}}$;

$e_{12} \leftarrow e_{11}^{-c}(1+z_{11}N)z_{21}^q \bmod N^2$;

$e_{22} \leftarrow e_{21}^{-c}(1+z_{12}N)z_{22}^q \bmod N^2$;

return($\Theta||U||e_{11}||e_{12}||e_{21}||e_{22}, z_{11}||z_{12}||z_{21}||z_{22}$);

. For extractor Ext=(Ext_1,Ext_2), $\text{Ext}_1(k)$ is:

generate at random RSA primes p_1, $p_2>q$ *s.t.* q dividing neither p_1-1 nor p_2-1;

$N \leftarrow p_1 p_2$; $\sigma \leftarrow N$; $\tau \leftarrow \varphi(N)$ /*φ is Euler function.*/

return(σ,τ); /*τ is extractor's trapdoor.*/

Obviously the σ generated by $\mathrm{Ext}_1(k)$ has the same distribution as c.r.s.

$\mathrm{Ext}_2(N, \tau, (\Theta||U||e_{11}||e_{12}||e_{21}||e_{22}, c, z_{11}||z_{12}||z_{21}||z_{22}))$ is:

Compute d: $qd = 1 \bmod \varphi(N)$;

$\alpha_1 \leftarrow e_{11}^{\ d} \bmod N$; $\hat{s} \leftarrow ((\alpha_1^{-q} e_{11} - 1) \bmod N^2) / N$;

$\alpha_2 \leftarrow e_{21}^{\ d} \bmod N$; $\hat{t} \leftarrow ((\alpha_2^{-q} e_{21} - 1) \bmod N^2) / N$;

return($\hat{s}, \hat{t}$);

P($\chi_{00}||\Omega||\xi_0||g_0||g_1$, $s||t$) V($\chi_{00}||\Omega||\xi_0||g_0||g_1$)

$r_1, r_2 \leftarrow^{\$} Z_q$; $\alpha_1, \alpha_2, \beta_1, \beta_2 \leftarrow^{\$} Z^*_N$;

$\Theta \leftarrow g_0^{r_1} g_1^{r_2}$; $U \leftarrow \Omega^{r_1}$;

$e_{11} \leftarrow (1+sN)\, \alpha_1^{\,q} \bmod N^2$; $e_{12} \leftarrow (1+r_1 N)\, \beta_1^{\,q} \bmod N^2$;

$e_{21} \leftarrow (1+tN)\, \alpha_2^{\,q} \bmod N^2$; $e_{22} \leftarrow (1+r_2 N)\, \beta_2^{\,q} \bmod N^2$;

$$\xrightarrow{\Theta||U||e_{11}||e_{12}||e_{21}||e_{22}}$$

$$\xleftarrow{\quad c \quad} \qquad c \leftarrow^{\$} Z_q$$

$z_{11} \leftarrow r_1 + sc$; $z_{12} \leftarrow r_2 + tc$;

$z_{21} \leftarrow \alpha_1^{\,c} \beta_1 \bmod N$; $z_{22} \leftarrow \alpha_2^{\,c} \beta_2 \bmod N$;

$$\xrightarrow{z_{11}||z_{12}||z_{21}||z_{22}}$$

$$\text{verify } g_0^{z_{11}} g_1^{z_{12}} = \Theta \xi_0^c \wedge \Omega^{z_{11}} = \chi_{00}^c U$$
$$\wedge\, e_{11}^c e_{12} = (1+z_{11}N) z_{21}^q \bmod N^2$$
$$\wedge\, e_{21}^c e_{22} = (1+z_{12}N) z_{22}^q \bmod N^2$$

Figure 4 Ω-protocol for $NMZPoK((s,t)$: $\chi_{00}=\Omega^s \wedge \xi_0 = g_0^s g_1^t$)'s construction

To make sure this protocol is indeed a Ω–protocol, we need to prove the fact that when there exist two transcripts ($\Theta||U||e_{11}||e_{12}||e_{21}||e_{22}$, c, $z_{11}||z_{12}||z_{21}||z_{22}$) and ($\Theta||U||e_{11}||e_{12}||e_{21}||e_{22}$, c', $z'_{11}||z'_{12}||z'_{21}||z'_{22}$) with c≠c' mod q but all accepted by the verifier V, Ext_2 really outputs a witness (s,t). At first we observe that c≠c' mod q implies g.c.d.(c-c', N)=1, because N's prime factors $p_1, p_2 > q > |c-c'|$, furthermore there exists $(c-c')^{-1} \bmod N$. Now by

$$\Omega^{z_{11}} = \chi_{00}^c U \text{ and } \Omega^{z'_{11}} = \chi_{00}^{c'} U$$

we have $\Omega^{z'_{11} - z_{11}} = \chi_{00}^{c'-c}$, i.e., (c'- c)s = z'$_{11}$ – z$_{11}$ mod q; by

$$g_0^{z_{11}} g_1^{z_{12}} = \Theta \xi_0^c \text{ and } g_0^{z'_{11}} g_1^{z'_{12}} = \Theta \xi_0^{c'}$$

we have $g_0^{z'_{11}-z_{11}} g_1^{z'_{12}-z_{12}} = \xi_0^{c'-c}$ and by (c'- c)s = z'$_{11}$ – z$_{11}$ mod q we derive (c'- c)t = z'$_{12}$ – z$_{12}$ mod q; by $e_{11}^c e_{12} = (1+z_{11}N) z_{21}^q \bmod N^2$ and $e_{11}^{c'} e_{12} = (1+z'_{11}N) z'^q_{21} \bmod N^2$ (note that $1+mN=(1+N)^m$ mod N^2) we have $e_{11}^{c'-c} = (1+N)^{z'_{11}-z_{11}} (z'_{21} z_{21}^{-1})^q$ = $(1+N)^{(c'-c)s+uq} (z'_{21} z_{21}^{-1})^q = (1+N)^{(c'-c)s} \rho^q \bmod N^2$ (where u and ρ are unnecessary to be explicitly computed), and recall(in subsection 4.3.1)that the order of 1+N modular N^2 is N, now raise both sides to the power of (c-c')$^{-1}$ mod N (recall that (c-c')$^{-1}$ mod N exists)then $e_{11} = (1+N)^s \gamma^q = (1+sN)\gamma^q \bmod N^2$, i.e., e_{11} is s's *Paillier* ciphertext, so $\hat{s}$=s.

Finally by $e_{21}^{c} e_{22} = (1+z_{12}N) z_{22}^{q} \bmod N^2$ and $e_{21}^{c'} e_{22} = (1+z'_{12}N) z'^{q}_{22} \bmod N^2$ we have $e_{21}^{c'-c} = (1+N)^{z'_{12}-z_{12}} (z'_{22} z_{22}^{-1})^q \bmod N^2$ and by (c'-c)t = z'$_{12}$ − z$_{12}$ mod q a similar calculation derives $e_{21} = (1+tN)\lambda^q \bmod N^2$, i.e., e_{21} is *t's Paillier* ciphertext, so $\hat{t}$=*t*.

5 Generalization to GUC-Security

To generalize our UC-secure set-intersection computation protocol ψ to the GUC-secure one, its structure(fig.1) is unchanged while only all the underlying non-malleable zero-knowledge proof protocols are replaced with new, enhanced zero-knowledge proof protocols, i.e., ID-augmented non-malleable zero-knowledge proof protocols.

5.1 defines the new type of zero-knowledge proof protocol, 5.2 presents a general framework to construct it, 5.3 applies this tool to obtain the GUC-secure protocol ψ*.

5.1 Basic Concepts

Recently [15] improves and generalizes the early UC-theory[14] to make a more general 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). This pre-setup can be naturally 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-keys generator. $\overline{G}_{acrs}^{Setup,UKG}$'s program is[15]:

Initialization Phase: *generate ρ at random*; *compute* (*mpk*, *msk*)←Setup(*ρ*); *store* (*mpk*, *msk*);

Running Phase: *on receiving message* ("*CRS request*", P_i*) *from any party*

P_i*, *send* ("CRS", *mpk*) *to* P_i* *and the ideal-world adversay* S;

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

Our GUC-secure protocol is in the ACRS-model. For this goal we introduce some new concepts about commitment and zero-knowledge proofs of knowledge.

Definition 5.1(*Identity-based Trapdoor Commitment Sheme*[15]) 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*,*d*), Vf(*mpk*, *id*, *M*, *cmt*, *d*) outputs 0 or 1, verifying whether *cmt* is *M*'s commitment with respect to *id*. These algorithms have the consistency property, i.e., for any *M* and *id*

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

FakeCmt(*mpk*,*id*,usk(*id*)) generates ($\overline{cmt}$,ξ), FakeDmt(*mpk*, *M*, ξ, $\overline{cmt}$) generates $\overline{d}$ (*M*)(w.l.o.g. ξ can contain *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 three additional properties:

(1)*Hiding*: for any *id* and M_0, M_1, (cmt_i,d_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, $Adv_{IBTC,A}^{binding}(k)$≡P[(*mpk*, *msk*)←Setup($k$); (*id**, *cmt**, M_0*, d_0*, M_1*, d_1*)←$A^{UKG(\mathrm{msk},.)}$(*mpk*): A doesn't query *oracle*-U(*msk*,.) with *id**∧M_0*≠M_1*∧Vf(*mpk*, *id**, M_0*, *cmt**, d_0*)=Vf(*mpk*, *id**, M_1*, *cmt**, d_1*)=1] is always a negligible function 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:

$(mpk, msk) \leftarrow \mathrm{Setup}(k)$;
$(St, id^*, M^*) \leftarrow A_1(mpk, msk)$;
$\mathrm{usk}(id^*) \leftarrow \mathrm{UKG}(msk, id^*)$; $(\overline{cmt}, \xi) \leftarrow \mathrm{FakeCmt}(mpk, id^*, \mathrm{usk}(id^*))$;
$d_1 \leftarrow \mathrm{FakeDmt}(mpk, M^*, \xi, \overline{cmt})$;
$d_0 \leftarrow^{\$} \{0,1\}^{|d1|}$;
$b \leftarrow^{\$} \{0,1\}$;
$b^* \leftarrow A_2(St, d_b)$;

Note that equivocability implies $P[\mathrm{Vf}(mpk, id^*, M^*, \overline{cmt}, d_1^*)=1] > 1-\gamma(k)$ where $\gamma(k)$ is a negligible function in k. [15] presented an efficient IBTC construction and proved its security.

Definition 5.2 and definition 5.3 introduce two powerful tools we need to make GUC-security. They are identity-augmented Ω-protocol and identity-augmented non-malleable zero-knowledge proof protocol. The former is denoted as IA-Ω protocol, the latter IA-NMZPoK protocol.

Definition 5.2 (IA-Ω *Protocol for Relation* R) The IA-Ω protocol for relation R $id\Omega_R$=(D, Setup, UKG, A,Z,Φ, Sim, Ext=(Ext_1,Ext_2)) is a group of P.P.T. algorithms, where D(k) generates identity σ, Setup(k) generates master public/secret-key pair (mpk, msk), UKG(msk,σ) generates σ's private-key usk(σ). More precisely, the valid σ can only have a prefix "sim" or "ext". UKG(msk,"sim"||σ_0) is called simulation-trapdoor, UKG(msk,"ext"||σ_0) is called extraction-trapdoor and UKG outputs nothing for any other σ. All other algorithms take (mpk,σ) as one of its inputs so (mpk,σ) no longer explicitly appears unless for emphasis. The protocol has the same structure and the same properties as the Ω-protocol in definition 4.1.

Definition 5.3(IA-NMZPoK *Protocol for Relation* R) The IA-NMZPoK Protocol for relation R IA-NMZPoK_R=(D,Setup,UKG,P,V,Sim=(Sim_1,Sim_2),Ext=(Ext_1,Ext_2)) is a group of P.P.T. algorithms, where Setup(k) generates master public/secret-key pair (mpk, msk), UKG(msk,σ) generates σ's private-key usk(σ), all other algorithms take (mpk,σ) as one of its inputs(so it no longer explicitly appears unless for emphasis). The protocol has the same properties as R's NMZPoK protocol(definition 2.3).

5.2 IB-NMZPoK Protocol's Construction

5.2.1 A Genral Construction

Theorem 5.1 presents a very general and systematic construction for IA-NMZPoK protocol. It uses a secure(unforgeable) one-time signature scheme, an secure IBTC sheme(definition 5.1) and IA-Ω protocol (definition 5.2) as components. Note that among these components secure one-time signature scheme and IBTC scheme can all be efficiently constructed and only the IA-Ω protocol relates to the specific relation R, therefore theorem 5.1 can be regarded as a transformation from (comparatively weak) IA-Ω protocol to the IA-NMZPoK protocol.

Theorem 5.1 Given a binary relation R and its IA-Ω protocol $id\Omega_R$=(D_ω, Setup, UKG, A,Z,Φ,Sim, Ext=(Ext_1,Ext_2)) with its master public/secret-key pair (mpk_ω, msk_ω); SIG_1=(KGen, Sign, Vf) is a secure(UF_CMA(1)) 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 (mpk_{TC},msk_{TC}); H is a one-way function mapping SIG_1's public-key space to D_ω. The protocol IA-$NMZPoK_R$ is constructed in Figure 5 where its master public-key $mpk=mpk_\omega||mpk_{TC}$, master secret-key $msk=msk_\omega||msk_{TC}$, UKG(msk,"sim"||σ_0) outputs IBTC::UKG(msk_{TC},σ_0), UKG(msk,"ext"||σ_0) outputs $id\Omega^R$::UKG(msk_ω,σ_0) and outputs nothing for other input.

Under these conditions, IA-$NMZPoK_R$ is a IA-NMZPoK protocol for relation R.

Proof The proof is essentially a generalization of [16]'s theorem 4.2, for simplicity here we only state the points which are different from there. IA-$NMZPoK_R$'s simulation algorithm Sim=(Sim_1,Sim_2) where $Sim_1(mpk)$ is specified as:

(vk,sk)←SIG_1::KGen(k); σ←H(vk);

s←UKG(msk,"sim"||vk);

/*s is the simulation trapdoor, $msk=msk_\omega||msk_{TC}$ so s=$usk_{TC}(vk)$. This computation is equal to sending message ("*Retrieve*", *sid*, P, *vk) to* $\overline{G}_{acrs}^{Setup,UKG}$ and then get the response s. This is consistent to $\overline{G}_{acrs}^{Setup,UKG}$'s specification since in the proof only the corrupted party needs to run the simulator. */

return(σ, s);

Sim_2(mpk, σ, s, x, c) is:

$(\overline{cmt},\xi)$←IBTC::FakeCmt(mpk_{TC},vk, s);

(a, z)←idΩ^R::Sim(mpk_ω, σ, x, c);

$\overline{d}$ ←FakeDmt(mpk_{TC}, a, ξ, $\overline{cmt}$);

s←SIG_1::Sign(sk, vk||$\overline{cmt}$||c||a||$\overline{d}$||z);

return(vk||$\overline{cmt}$, a||$\overline{d}$||z||s);

The extractor Ext=(Ext_1,Ext_2) where Ext_1(mpk) is:

(vk,sk)←SIG_1::KGen(k); σ←H(vk);

s←UKG(msk,"sim"||vk); τ←UKG(msk,"ext"||σ);

/* s=usk_{TC}(vk), τ=usk_ω(σ). Refer to comments in Sim_1(mpk).*/

return(σ,s,τ);

Ext_2(mpk, σ,τ, (vk||cmt, c, a||dmt||z||s)) is:

Run idΩ^R::V(mpk_ω, σ, x);

if idΩ^R::V outputs 1 *then* w←idΩ^R::Ext_2(mpk_ω, σ,τ, (a, c, z)) *else* w←⊥;

return(w);

mpk=mpk_ω||mpk_{TC}

P(x,w): R(x,w)=1 V(x)

(vk,sk)←SIG_1::KGen(k); σ←H(vk);

∨r; a←A(mpk_ω,σ,x,w,r);

(cmt,dmt)←IBTC::Cmt(mpk_{TC}, vk, a);

$\xrightarrow{vk||cmt}$

$\xleftarrow{c}$ c←$^\$$ $\{0,1\}^k$;

z←Z(mpk_ω,σ,x,w,r,c); σ←H(vk);

s←SIG_1::Sign(sk, vk||cmt||c||a||dmt||z);

$\xrightarrow{a||dmt||z||s}$

verify IBTC::Vf(mpk_{TC}, vk, a, cmt, dmt)=1

∧Φ(mpk_ω,σ, x,a,c,z)=1∧SIG_1::Vf(vk, vk||cmt||c||a||dmt||z,s)=1

Figure 5 IA-NMZPoK protocol IA-$NMZPoK_R$ for R

Now verify that Sim and Ext indeed satisfy the properties in definition 5.3 and definition 2.3, but the analysis here is almost the same as in [16]'s theorem 4.2's proof. The only difference is symbolic: their sk should be replaced with s(sk is TC's trapdoor there; the symbol *sig_vk* used there is *vk* used here, "tag" used there is IBTC scheme's *id* here)，so the details can be omitted and we only present the final consequences: 1)Sim satisfies the zero-knowledge simulation property; 2)the extractor's error function $\eta(k)<O(n)(Adv_H^{OW}(k)+ Adv_{SIG1}^{UF\_CMA(1)}(k))+\sqrt{Adv_{IBTC}^{binding}(k)+2^{-k}}$ where n the number of sessions，$Adv_H^{OW}(k)$, $Adv_{SIG1}^{UF\_CMA(1)}(k)$ and $Adv_{IBTC}^{binding}(k)$ are attacker's advantages for H, SIG_1 and IBTC schemes, all are negligible in k.

Theorem 5.2 shows why IA-NMZPoK protocol is so powerful(we won't apply it, just put it here to show our method is reasonable).

Theorem 5.2 F^R_{ZK} is the ideal zero-knowledge proof functionality for relation R, $IA\text{-}NMZPoK_R$ is an IA-NMZPoK protocol for R, then $IA\text{-}NMZPoK_R \rightarrow^{GUC} F^R_{ZK}$ assuming static corruptions.

Proof The proof is essentially the same as [17]'s theorem 5.1.

5.2.2 Constructions of *IA-NMZPoK*, *IA-NMZPoK*$_{II}$ and *IA-NMZPoK*$_{III}$

Theorem 5.1 reduces IA-NMZPoK protocol's construction to IBTC scheme and IA-Ω protocol of the desired relation R. In fact [15] has presented an efficient realization of the former, so we only need to make an efficient solution to the IA-Ω protocol's construction with respect to those relations in our instantiation. The tool is the elliptic curve *Paillier* scheme proposed by Galbraith in [20].

Galbrith-Paillier scheme works on the elliptic curve E/Z_N over the ring Z_N, where N is a RSA modular($N=p_1p_2$, both $(p_1-1)/2$ and $(p_2-1)/2$ are also primes). On the curve a point's coordinate is represented in projective form [x,y,z]. Given A, $B \in Z_N$ such that g.c.d.(N, $6(4A^3+27B^2)$)=1, the curve with coefficients A, B has the equation

$$E_{A,B}/Z_N\text{: } y^2z=x^3 + Axz^2 + Bz^3$$

(when A, B is not important to discussions we simply use the expression E/Z_N instead and the cardinality of the point group on the curve is denoted as $|E/Z_N|$). Galbraith-Paillier scheme's plaintext space is Z_N. E/Z_N can be also regarded as a curve E/Z_{N^2} over the larger ring Z_{N^2} and for $m \in Z_N$ denote the point $[mN,1,0]$ on E/Z_{N^2}

as P_m. On the other hand, taking A, B modulo N's prime factor $p\in\{p_1, p_2\}$ then E/Z_N can be also regarded as the curve over the field F_p.

If N's factors p_1, p_2 are known then an important quantity $M_{A,B}$=l.c.m.($|E/F_{p1}|$,$|E/F_{p2}|$) can be computed in polynomial-time, e.g., via Schoof-Atkin-Elkies algorithm, on the reverse N can be effectively factorized given $M_{A,B}$[10]. From this observation Galbrith-Paillier scheme can be regarded as an IBE scheme (Setup,UKG,E,D) where complexity parameter k is the bits of N's prime factors, Setup(k) generates mpk=N and msk=N's prime factors (p_1, p_2); id is (A,B,NQ) where $(A,B)\in Z_N\times Z_N$, $(N,6(4A^3+27B^2))=1$, $Q\in E_{A,B}/Z_{N^2}$ (so $M_{A,B}NQ=\infty$, i.e., "zero" in the group); For $(A,B,NQ)\in$ID, UKG(msk,(A,B)) computes usk(A,B,NQ)=$M_{A,B}$ as the user private-key of (A,B,NQ); the plaintext space is Z_N, for $m\in Z_N$ the encryption algorithm E(N, (A,B,NQ), m) selects $r\in Z_N$ at random then computes $y= P_m + rQ_0$ on E/Z_{N^2}, where Q_0=NQ and P_m is as the above; the decryption algorithm D(N, usk(A,B,NQ), y) computes $M_{A,B}y(=M_{A,B}P_m=[mM_{A,B} N,1,0])$'s x-coordinate $X_y\in Z_{N^2}$ and outputs $M_{A,B}^{-1}(X_y/N)$ mod N. Galbraith-Paillier scheme is also homomorphic. All details of this scheme including its security conditions refer to [20].

Similar as in subsection 4.3, it's demonstrative enough to construct the IA-Ω protocol for the relation $\chi_{00}=\Omega^s\wedge\xi_0=g_0^s g_1^t$ on the prime-order group G, |G|=q. This IA-Ω protocol is in figure 6 and in $\overline{G}_{acrs}^{Setup,UKG}$ model, where msk=RSA primes (p_1, p_2), q divides neither p_1-1 nor p_2-1(e.g., p_1, p_2>4q), $mpk=N=p_1p_2$; c.r.s. σ is Galbraith-Paillier scheme's "user-id", i.e., (A,B,NQ) where the coefficients A, B and the random point Q on $E_{A,B}/Z_{N^2}$ can be obtained by hashing the protocol parties' names(realistic hash functions make the probability of g.c.d.$(N,6(4A^3+27B^2))=1$ almost 1, otherwise N can be effectively factorized). For simplicity, denote NQ as Q_0 and the curve's coefficients as P, V (so $M_{P,V}Q_0=\infty$).Note that in figure C.2 all e_{ij} are Galbraith-Paillier ciphertexts.

The protocol's simulator Sim(N,c) is specified as:

$z_{11}, z_{12} \leftarrow^{\$} Z_q$; $z_{21}, z_{22} \leftarrow^{\$} Z^*_N$; $e_{11}, e_{21} \leftarrow^{\$} E_{P,V}/Z^*_{N^2}$;

$\Theta \leftarrow \xi_0^{-c}\, g_0^{z_{11}} g_1^{z_{12}}$; $U \leftarrow \chi_{00}^{-c}\,\Omega^{z_{11}}$;

$e_{12} \leftarrow P_{z_{11}} + qz_{21}Q_0 - ce_{11}$;

$e_{12} \leftarrow P_{z_{12}} + qz_{22}Q_0 - ce_{21}$;

return($\Theta || U || e_{11} || e_{12} || e_{21} || e_{22}$, $z_{11} || z_{12} || z_{21} || z_{22}$) ;

It's easy to verify that Sim(N,c) satisfies the zero-knowledge simulation property. The extraction algorithm Ext=(Ext_1,Ext_2) where $\mathrm{Ext}_1(k)$ is:

Generate RSA primes p_1, $p_2 > q$ and q evenly divides neither p_1-1 nor p_2-1;

$N \leftarrow p_1 p_2$;

$Q \xleftarrow{\$} E_{P,V}/Z^*_{N^2}$; $Q_0 \leftarrow NQ$; $\sigma \leftarrow$(curve $E_{P,V}/Z_N$,Q_0);

$\tau \leftarrow$UKG(msk,(P,V));

/*i.e., τ is the extraction trapdoor $M_{P,V}$. Refer to the comments for $\mathrm{Sim}_1(mpk)$ in theorem 5.1's proof. */

return(σ,τ);

c.r.s. σ=(curve $E_{P,V}/Z_N$, Q_0)

P($\chi_{00} || \Omega || \xi_0 || g_0 || g_1$, $s || t$) V($\chi_{00} || \Omega || \xi_0 || g_0 || g_1$)

$r_1, r_2 \xleftarrow{\$} Z_q$; α_1, α_2, β_1, $\beta_2 \xleftarrow{\$} Z^*_N$;

$\Theta \leftarrow g_0^{r_1} g_1^{r_2}$; $U \leftarrow \Omega^{r_1}$;

$e_{11} \leftarrow P_s + q\alpha_1 Q_0$; $e_{12} \leftarrow P_{r1} + q\beta_1 Q_0$;

$e_{21} \leftarrow P_t + q\alpha_2 Q_0$; $e_{22} \leftarrow P_{r2} + q\beta_2 Q_0$;

$\xrightarrow{\Theta || U || e_{11} || e_{12} || e_{21} || e_{22}}$

$\xleftarrow{\quad c \quad}$ $c \xleftarrow{\$} Z_q$

$z_{11} \leftarrow r_1 + sc$; $z_{12} \leftarrow r_2 + tc$;

$z_{21} \leftarrow c\alpha_1 + \beta_1$; $z_{22} \leftarrow c\alpha_2 + \beta_2$; $\xrightarrow{z_{11} || z_{12} || z_{21} || z_{22}}$

verify $g_0^{z_{11}} g_1^{z_{12}} = \Theta \xi_0^c \wedge \Omega^{z_{11}} = \chi_{00}^c U$

$\wedge ce_{11} + e_{12} = P_{z_{11}} + z_{21} q Q_0$

$\wedge ce_{21} + e_{22} = P_{z_{12}} + z_{22} q Q_0$

Figure 6 IA-Ω protocol for relation $\chi_{00} = \Omega^s \wedge \xi_0 = g_0^s g_1^t$

Obviously the σ generated by $\mathrm{Ext}_1(k)$ has the same distribution as the c.r.s. Ext_2(N, τ, ($\Theta || U || e_{11} || e_{12} || e_{21} || e_{22}$, c, $z_{11} || z_{12} || z_{21} || z_{22}$)) is:

Compute ($\hat{s}$, $\hat{t}$) by *Galbraith-Paillier* decryption algorithm,

i.e., $\hat{s} \leftarrow$D(N, τ, e_{11}); $\hat{t} \leftarrow$D(N, τ, e_{21});

return($\hat{s}$, $\hat{t}$);

We need to prove that if there exist two transcripts ($\Theta||U||e_{11}||e_{12}||e_{21}||e_{22}$, c, $z_{11}||z_{12}||z_{21}||z_{22}$) and ($\Theta||U||e_{11}||e_{12}||e_{21}||e_{22}$, c', $z'_{11}||z'_{12}||z'_{21}||z'_{22}$) with the same first-message but c≠c' mod q and both accepted by V, then Ext_2 extracts the real witness (s,t). In fact, $\Omega^{z_{11}} = \chi_{00}^{c}U$ and $\Omega^{z'_{11}} = \chi_{00}^{c'}U$ imply $\Omega^{z'_{11}-z_{11}} = \chi_{00}^{c'-c}$, i.e.,(c'- c)s = z'$_{11}$ – z$_{11}$ mod q; $g_0^{z_{11}}g_1^{z_{12}} = \Theta\xi_0^{c}$ and $g_0^{z'_{11}}g_1^{z'_{12}} = \Theta\xi_0^{c'}$ imply $g_0^{z'_{11}-z_{11}}g_1^{z'_{12}-z_{12}} = \xi_0^{c'-c}$, by (c'- c)s = z'$_{11}$ – z$_{11}$ mod q we have (c'- c)t = z'$_{12}$ – z$_{12}$ mod q. Furthermore,

$$ce_{11}+e_{12}= P_{z_{11}} + z_{21}qQ_0 \quad \text{and} \quad c'e_{11}+e_{12}= P_{z'_{11}} + z'_{21}qQ_0$$

imply (c' – c)e_{11} = $P_{z'_{11}} - P_{z'_{11}} + (z'_{21}-z_{21})qQ_0$ = $P_{z'_{11}-z_{11}} + (z'_{21}-z_{21})qQ_0$ = $P_{z'_{11}-z_{11}+uq} + (z'_{21}-z_{21})qQ_0$ (where $|u|<q$ and unnecessary to be explicitly computed) so the x-coordinate of $M_{P,V}$(c' – c)e_{11} is (z'$_{11}$ – z$_{11}$+uq)N$M_{P,V}$ mod N^2; Acoording to *Galbraith-Paillier*'s decryption algorithm D, the $\hat{s}$ (=D(N, $M_{P,V}$, e_{11}), output by Ext$_2$) has the x-coordinated of $M_{P,V}e_{11}$=$\hat{s}\,M_{P,V}$ N mod N^2, so the x-coordinate of (c'–c)$M_{P,V}e_{11}$=(c'–c)$\hat{s}\,M_{P,V}$N mod N^2, hence (c'– c)$\hat{s}$=z'$_{11}$ – z$_{11}$ mod$q$. By (c'- c)s=z'$_{11}$ – z$_{11}$ mod q, we get $\hat{s}$ = s mod q. Finally, by ce_{21}+e_{22}= $P_{z_{12}} + z_{22}qQ_0$ and c'e_{21}+e_{22}=$P_{z'_{12}} + z'_{22}\,qQ_0$ we can similarly get $\hat{t}$ =t mod q.

Now apply theorem 5.1 to the above construction we can get all IA-NMZPoK protocols for the desired relations ((4-1)～(4-3)) in the instantiation.

5.3 ψ$\rightarrow^{\text{GUC}}$ F_{INT}和 $\Delta_{Blind-UKG}^{Boyen-Waters} \rightarrow^{\text{GUC}} F_{Blind-UKG}^{Boyen-Waters}$

So far all necessary tools are ready and we can get the final consequences.

Theorem 5.3 If all zero-knowledge proof protocols in $\Delta_{Blind-UKG}^{Boyen-Waters}$ (figure 2) are IA-NMZPoK, then $\Delta_{Blind-UKG}^{Boyen-Waters} \rightarrow^{\text{GUC}} F_{Blind-UKG}^{Boyen-Waters}$ and $\Delta_{Blind-UKG}^{Boyen-Waters}$ satisfies definition 3.1 assuming static corruptions.

Proof The proof's logic is essentially the same as theorem 4.1, with only symbolic differences: protocols *NMZPoK*$_{\text{II}}$'s and *NMZPoK*$_{\text{III}}$'s simulation and extraction algorithms are replaced with *IA-NMZPoK*$_{\text{II}}$'s and *IA-NMZPoK*$_{\text{III}}$'s counterparts, in particular, the simulation trapdoor s and extraction trapdoor τ corresponding to σ are s=UKG(msk,"sim"||σ) and τ=UKG(msk,"ext"||σ) respectively; any other algorithms take (mpk,σ) as one of their inputs. Since s and τ still work in the same way as in theorem 4.1's proof, the consequence can be obtained in the same way.

Theorem 5.4 Protocol $\Delta_{Blind-UKG}^{Boyen-Waters}$ is as the above, the zero-knowledge proof protocol in ψ(figure 1) is IA-NMZPoK and the commitment scheme C is secure IBTC, then $\psi \rightarrow^{\text{GUC}} F_{\text{INT}}$ assuming static corruptions.

Proof Essentially the same as theorem 3.1 for the same reason as stated in theorem 5.3's proof.

REFERENCES

[1] R.Cramer, I. Damgard *Multiparty Computation: an Introduction*, In: Advanced Courses in Contemporary Cryptology, Berlin:Springer-Verlag, 41-88, 2005.

[2] O.Goldreich *Foundations of Cryptography*, Vol 1. *Basic Tools*; Vol 2. *Basic Applications*, Cambridge University Press, 2004.

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

[4] Y. Aumann, Y.Lindell, *Security against Covert Adversaries: Efficient Protocols for Realistic Adversaries*, TCC'07, LNCS Vol.4392, 137-156, 2007.

[5] M.Freedman, Y.Ishai, B.Pinkas et al *Keyword Search and Oblivious Psedorandom Functions*, TCC'05, LNCS Vol.3378, 303-324, 2005.

[6] M.Freedman, K.Nissim, B.Pinkas *Efficient Private Matching and Set Intersection*, Eurocryp'04, LNCS Vol.3027, 1-19, 2004.

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

[8] M.Green, S.Hoenberger *Blind Identity-based Encryption and Simulatable Oblivious Transfer*, Asiacrypt'07, LNCS Vol.4833, 265-282, 2007.

[9] G.Crescenzo, J.Katz, R.Ostrovsky et al, *Efficient and Non-interactive Non-Malleable Commitment*. 42nd Foundations of Computer Science Conference, 2001.

[10] A.Engle *Elliptic Curves and Their Applications to Cryptography*, Kluwer Academic Publishers, 1999.

[11] 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.

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

[13] M. Bellare, A. Boldyreva, A. Desai and D. Pointcheval. *Key-Privacy in Public-key Encryption*. In: C. Boyd ed, Advances in Cryptology - Asiacrypt 2001 Proceedings, LNCS Vol. 2248, Goldcoast Australia:Springer-Verlag, 2001, 566-582.

[14] R.Canneti, *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.

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

[16] P..MacKenzie, K.Yang *On Simulation-Sound Trapdoor Commitments*, Proc. Eurocrypt, LNCS Vol.3027, 382-400, 2004.

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

[18] I.Damgard, M.Jurik, *A Generalization, Simplificatiuon and Some Applications of Paillier's Probabilistic Public-Key Systems*, Proc. Eurocrypt, LNCS Vol.1992, 229-243, 2001

[19] D.Catalano, R.Gennaro, N.Howgrave-Graham et al, *Paillier's Cryptosystem Revisted*, 8th ACM Conf. on Computer and Communications Security, 206-214, 2001.

[20] S.Galbraith *Elliptic Curve Paillier Schemes*, Jounal of Cryptology, 15(2):129-138, 2002, also available at eArchive 2001/050.

[21] Y.Lindell, B.Pinkas *Secure Multiparty Computation for Privacy-Preserving Data Mining* eArchive 2008/147.

[22] R.Steinmetz, K.Wehrle(ed) Peer-to-Peer Systems and Applications, Part IV: Searching and Mining, Springler-Verlag, 2005.