Additive Conditional Disclosure of Secrets And Applications

Sven Laur[1] and Helger Lipmaa[2]

[1] Helsinki University of Technology, Finland
[2] Cybernetica AS and University of Tartu, Estonia

Abstract. During a conditional disclosure of secrets (CDS) protocol, Alice obtains a secret, held by Bob, if and only if her inputs to the protocol were "valid". As an output masking technique, CDS protocol can be used as a subroutine in other protocols to guarantee either Bob-privacy or correctness against a malicious Alice. Using a simple seeded randomness extractor, we extend the Aiello-Ishai-Reingold CDS protocol to work over additively homomorphic public-key cryptosystems. Based on this, we construct several new two-message protocols like an oblivious transfer protocol with log-squared communication and a millionaire's protocol with logarithmic communication. Additionally, we show how to implement private, universally verifiable and robust multi-candidate electronic voting so that all voters only transmit an encryption of their vote. Importantly, the only cryptographic hardness assumption in these protocols is that the underlying public-key cryptosystem is IND-CPA secure.
Keywords. Conditional disclosure of secrets, electronic voting, homomorphic encryption, malicious model, millionaire's problem, oblivious transfer, two-party computation.

1 Introduction

It is well known that one can implement secure computation by using garbled circuit techniques. However, since using garbled circuits often results in too inefficient protocols (that in particular take communication, linear in the circuit size), cryptographers tend to use alternative mechanisms. As an important example, the El Gamal cryptosystem [El 84] was used successfully to construct e-voting [CGS97] and other protocols. However, due to the limitations of El Gamal-like multiplicatively homomorphic public-cryptosystems, in such protocols the intended receiver has to compute discrete logarithm to recover the protocol outcome. Therefore, basing protocols on multiplicatively homomorphic public-key cryptosystems is often only feasible in tasks where the expected outcome is "small" (e.g., in two-candidate e-voting protocols [CGS97]).

Additively homomorphic (AH) public-key cryptosystems PKC like the Paillier [Pai99] make it possible to efficiently implement many interesting cryptographic protocols without the need of solving discrete logarithms. In this paper we consider protocols where Alice forwards some PKC-ciphertexts to Bob who, after computing on ciphertexts, sends some other ciphertexts to Chalice (who, say, can be the same person as Alice or a coalition of third parties who will threshold-decrypt the ciphertexts). We call such protocols either *return-to-sender* or *threshold additively homomorphic (AH) two-message protocols*. Some noteworthy tasks that can be efficiently solved by using AH two-message protocols include computationally-private information retrieval (CPIR, [AIR01,Ste98,Lip05]), millionaire's problem [BK04], linear algebraic tasks (e.g., private matrix multiplication), various privacy-preserving data mining tasks (e.g., private scalar product [WY04,GLLM04] and private set intersection cardinality), multi-candidate electronic voting [DJ01] and electronic auctions [LAN02].

Now, computational Alice-privacy of an AH two-message protocol in the malicious model follows directly from the IND-CPA security of the underlying AH public-key cryptosystem. On the other hand, many common return-to-sender AH two-message protocols (e.g., the basic variants of [AIR01,Ste98,Lip05,BK04,WY04,GLLM04]) are statistically Bob-private only in the semi-honest model, that is, under the assumption that Alice encrypts correct values. Often, there is no guarantee of Bob-privacy whatsoever in the case of malicious Alice. Analogously, many threshold AH two-message protocols [DJ01,LAN02] are correct only in the semi-honest model. Therefore, one must guarantee that if Alice encrypts invalid inputs then she obtains no new information (in the case of return-to-sender protocols) or Chalice will be able to detect the sending of invalid inputs (in the case of threshold protocols). These desiderata are usually achieved by using zero-knowledge proofs that either increase the number of rounds or require a security model with non-complexity-theoretic assumptions (e.g., random oracles).

Alternatively, one can use conditional disclosure of secrets ($\mathsf{CDS}_\ell^{\mathcal{S}}$) from [GIKM00,AIR01], also known as input verification gadget [BGN05], where Chalice obtains Bob's secret input if and only if Alice encrypted an ℓ-bit string from the set $\mathcal{S}$. In parallel and in another protocol, this secret can be used by Bob to mask the output values, sent

to Chalice. However, the Aiello-Ishai-Reingold CDS protocol [AIR01] works only in conjunction with an IND-CPA secure homomorphic public-key cryptosystem PKC that has plaintext space of prime order n. The only widely known such cryptosystem, ElGamal [El 84], is multiplicatively homomorphic while in the case of almost all AH cryptosystems, n is a large composite integer with sufficiently large prime factors. Alternatively, the Boneh-Goh-Nissim CDS protocol [BGN05] uses a PKC that satisfies stronger requirements than additive homomorphicity: namely, there Bob can compute-on-ciphertexts any quadratic functions of the plaintexts. The PKC of [BGN05] works also on plaintext groups of composite order but paradoxically, their CDS protocol is secure exactly because the Boneh-Goh-Nissim PKC has inefficient decryption. (Also here, one has to compute discrete logarithms to decrypt.) We would like to achieve security even in the case when PKC has efficient decryption.

In this paper, we modify the Aiello-Ishai-Reingold CDS protocol so that it can be used in conjunction with a PKC that satisfies substantially weaker (algebraic) properties than required by [AIR01,BGN05]; in particularly, PKC has to be AH, IND-CPA secure, and the smallest prime factor $\Phi(n)$ of n has to be large. (See Thm. 2 for the precise requirements.) Let us call such PKC *CDS-friendly*; the cryptosystems from [Pai99,DJ01] are CDS-friendly.

Our construction consists of several steps. First, we design a lightweight seeded randomness extractor [GRS04,GR05a] for the family of distributions $\mathcal{D} = \{m\mathbb{Z}_n : m \in \mathbb{Z}_n \wedge m \neq 0\}$, and use it to construct an AH two-message 1-out-of-ν-oblivious transfer protocol for ℓ-bit strings (where ℓ is reasonably large) that is computationally Alice-private and statistically ε-Bob-private assuming that the underlying public-key cryptosystem is CDS-friendly. This is the first AH two-message oblivious transfer protocol, where honest Alice's first message is just an homomorphic encryption of the database index, and thus interesting by itself.

Second, we show how to transform any two-message 1-out-of-ν-computationally-private information retrieval (CPIR) protocol into a two-message statistically Bob-private oblivious transfer protocol and a $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol with exactly the same communication. If we use the Gentry-Ramzan CPIR protocol [GR05b] then the resulting protocols have communication $\Theta(\log \sharp\mathcal{S})$ but Bob's computation will be $\Theta(\sharp\mathcal{S})$. Assume that the input size is equal to $\log_2 \sharp\mathcal{S}$. Using arithmetic circuit evaluation, we show that $\mathcal{S}$ has a $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol with polynomial resources iff $\mathcal{S} \in \mathbf{P}/\mathbf{poly}$. Because we use an AH public-key cryptosystem, the resources will often be low-degree logarithmic.

Third, we propose the *conditional disclosure of secrets (CDS) transformation* that transforms an arbitrary AH two-message protocol Π that securely implements a *public* function f in the semi-honest model to a protocol that is secure in the malicious model. The basic idea is as in [AIR01]: in parallel with Π, Alice and Bob execute a CDS protocol for Alice's input so that the first message of the two protocols coincide. Bob masks the output with secrets, corresponding to all different Alice's inputs. Therefore, Alice recovers any of the outputs only if all of his inputs belong to the valid input sets. The resulting AH two-message protocol is efficient whenever the valid input set $\mathcal{S}$ has an efficient conditional disclosure of secrets protocol and the number of outputs λ is reasonably small. If we require only computational Bob-privacy then the resulting protocol is communication-efficient also for the large values of λ.

Until now, the CDS protocol has been largely overlooked in literature, with only a couple of published papers [GIKM00,AIR01,BGN05] that do more than mention it and with many papers using zero-knowledge proofs where the CDS protocol could provide a simpler solution. This situation might be partially due to the relatively small number of proposed applications. To remedy this situation and to popularise the CDS protocol, we propose several interesting applications to demonstrate the power of the new tools. First, we construct a new oblivious transfer protocol and a new CDS protocol with log-squared communication and computation. (See Cor. 2.) Second, in Sect. 6, we construct a private millionaire's protocol with logarithmic communication. Third, we construct efficient private protocols for a few other tasks like conditional oblivious transfer and multiplicative relationship (see Sect. 6), and scalar product(see Sect. 7). We also show how to construct efficient *threshold* AH protocols for e-voting and e-auctions. (See Sect. 6.) All constructed protocols are round-optimal, computationally Alice-private and statistically Bob-private solely under the assumption that the underlying public-key cryptosystem PKC is CDS-friendly. Importantly, the only complexity-theoretic hardness assumption is that PKC is IND-CPA secure.

Road-map. In Section 2, we state preliminaries. In Section 3, we propose a new seeded randomness extractor. In Section 4, we propose a new AH two-message protocol for oblivious transfer and prove its security. In Section 5, we propose an AH two-message protocol for conditional disclosure of secrets and show how to implement it efficiently for all sets in $\mathbf{P}/\mathbf{poly}$. In Section 6, we propose several interesting applications. In Section 7, we present our generic CDS transform and prove its security. In Appendices, we give proofs of some results.

2 Preliminaries

For an integer n, let $[n] := \{1, 2, \ldots, n\}$ and let $\Phi(n)$ be the smallest prime divisor of n. We say that n is p-*rough* if $\Phi(n) \geq p$. The statistical difference of two distributions X and Y over a discrete support Z is defined as $\mathsf{Dist}\,(X\|Y) := \max_{S\subseteq Z} |\Pr[X \in S] - \Pr[Y \in S]|$. We say that X and Y are ε-close ($X \overset{\varepsilon}{\sim} Y$) if $\mathsf{Dist}\,(X\|Y) \leq \varepsilon$. For an arbitrary set Z, let $U(Z)$ denote the uniform distribution over it; we sometimes identify Z with $U(Z)$. Let $\mathcal{D}$ be a family of distributions on some set M_1. A function $\mathsf{Ext} : M_1 \times S \to M_2$ is a *seeded* ε*-extractor* [GRS04,GR05a] for $\mathcal{D}$ if for every distribution X in $\mathcal{D}$, $\mathsf{Ext}(X, U(S)) \overset{\varepsilon}{\sim} U(M_2)$. A set Z with a binary operation $\circ : Z^2 \to Z$ is a quasigroup if and only if for every $a \in Z$, $a \circ U(Z) = U(Z) = U(Z) \circ a$.

Throughout this paper, we omit the security parameter k by assuming that the adversary works in time that is less than some fixed public constant τ, then also k is a constant. Sometimes, one needs security against adversaries that work in time, polynomial in the input size κ of the protocol Π. Then, k will depend on κ. More precisely, assume that the underlying computationally hard problem, with input n of size $\kappa := \log_2 n$, can be broken in time $L_n[a, b] := \exp(a(\ln n)^b \cdot (\ln\ln n)^{1-b})$ for some $0 < b \leq 1$. To guarantee security against such polynomial adversaries, it is necessary that $L_n[a, b] = \omega(\kappa^c)$ for every constant c, or that $k^b \cdot \ln^{1-b} k = \omega(\ln \kappa)$. Omitting the logarithmic factor, we get that $k = \Omega(\ln^{1/b} \kappa)$. E.g., when basing a protocol on the Decisional Composite Residuosity Assumption [Pai99] with $b = 1/3$, we must assume that $k = \Omega(\log^{3-o(1)} \kappa)$.

Public-key cryptosystem is a triple $\mathsf{PKC} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, where Gen is a key generation algorithm that returns a secret and public key pair $(\mathsf{sk}, \mathsf{pk})$, Enc is a randomised encryption algorithm and Dec is a decryption algorithm such that $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m; r)) = m$. For a fixed PKC and for a fixed public key, let $\mathcal{R}$ be the randomness space, let $\mathcal{M}$ be the plaintext space and let $\mathcal{C}$ be the ciphertext space. For an algorithm A, define $\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{PKC}}(A) := \frac{1}{2} \cdot \left|\mathsf{Succ}^{\text{IND-CPA}}_{\mathsf{PKC},1}(A) - \mathsf{Succ}^{\text{IND-CPA}}_{\mathsf{PKC},0}(A)\right|$, where

$$\mathsf{Succ}^{\text{IND-CPA}}_{\mathsf{PKC},b}(A) := \Pr[(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}, (m_0, m_1) \leftarrow A(\mathsf{pk}) : A(\mathsf{pk}, m_0, m_1, \mathsf{Enc}_{\mathsf{pk}}(m_b; U(\mathcal{R}))) = 1] \ ;$$

the probability is taken over the coin tosses of Gen and A, and over the choice of random variables. We say that PKC is (ε, τ)-*IND-CPA-secure* if $\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{PKC}}(A) \leq \varepsilon$ for any τ-time probabilistic algorithm A.

A public-key cryptosystem PKC is *homomorphic*, if for any key pair $(\mathsf{sk}, \mathsf{pk})$, any $x_1, x_2 \in \mathcal{M}$ and any $r_1, r_2 \in \mathcal{R}$, $\mathsf{Enc}_{\mathsf{pk}}(x_1; r_1) \cdot \mathsf{Enc}_{\mathsf{pk}}(x_2; r_2) = \mathsf{Enc}_{\mathsf{pk}}(x_1 + x_2; r_1 \circ r_2)$, where $+$ is a group operation in $\mathcal{M}$ and $\circ$ is a groupoid operation in $\mathcal{R}$. We say that PKC is *additively homomorphic* (AH) if $\mathcal{M} = \mathbb{Z}_n$ for some n, and *multiplicatively homomorphic*, if $\mathcal{M} = \mathbb{Z}_n^*$ for some n. Several well-known homomorphic cryptosystems [El 84,OU98,NS98,Pai99,DJ01,DJ03] are IND-CPA secure under reasonable complexity assumptions. The ElGamal cryptosystem [El 84] is multiplicatively homomorphic (and the only one where $\mathcal{M}$ has an odd prime order), while other cryptosystems are AH with a usually rough composite n.

The Paillier cryptosystem [Pai99] is one of the most efficient known IND-CPA secure AH public-key cryptosystems. Here, $\mathcal{M} = \mathbb{Z}_n$, $\mathcal{R} = \mathbb{Z}_n^*$ and $\mathcal{C} = \mathbb{Z}_{n^2}^*$ for an RSA modulus n. Thus, n is $\sqrt{n}/2$-rough. Its IND-CPA security follows from the Decisional Composite Residuosity Assumption [Pai99].

Return-to-sender AH two-message protocols. Let α denote the private input of Alice and β denote the private input of Bob. We mainly consider return-to-sender two-message protocols between Alice and Bob that implement the following functionality for some *public* function f: an unbounded Alice learns $f(\alpha, \beta)$ and nothing more, and a computationally bounded Bob learns no new information. An *return-to-sender AH two-message protocol for* f consists of the next phases. In the initialisation phase that is usually shared by different protocols, Alice generates his key pair $(\mathsf{sk}, \mathsf{pk})$ for PKC by executing Gen and transfers pk to Bob. In the following, we explicitly assume that the validity of public keys is assured. This can be achieved either in the presence of PKI (this assumption is normal in the case of applications like e-voting), or by letting Alice to prove once, separately and in an isolated manner (no other messages of different protocols are sent by Alice at the same time), to every Bob that pk is valid.

During an AH two-message protocol Π, on input α, a Alice computes a randomised message mq and sends it to Bob. We assume that one can efficiently verify, given only pk, that mq is a valid message. In our setting, $\mathsf{mq} = (\mathsf{Enc}_{\mathsf{pk}}(\alpha_1; r_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(\alpha_\mu; r_\mu))$, where $\alpha = (\alpha_1, \ldots, \alpha_\mu)$, $\alpha_i \in \mathcal{M}$ and $r_i \in \mathcal{R}$, for some $\mu \geq 1$. Therefore, validity verification can be done efficiently if and only if membership in $\mathcal{C}$ can be tested efficiently. In the second round, Bob replies with the second message mt, computed on inputs (β, mq). We assume that $\mathsf{mt} = \perp$ if Bob does not have the public key, Bob halts or mq is malformed. Finally, Alice recovers the answer by using a special recovery algorithm. In our setting, this means that Bob applies a number of randomised operations to the received ciphertexts, and returns the resulting ciphertexts; Alice then decrypts the received ciphertexts, and applies some local algorithm to the resulting plaintexts. The *communication* of an AH two-message protocol is equal to $|\mathsf{mq}| + |\mathsf{mt}|$.

We use the "standard" relaxed security definitions (see, e.g., [AIR01,FIPR05]) where one only cares about the correctness and the privacy of participants. Briefly, (1) Π is *correct* if in the case of honest Alice and Bob, Alice always recovers $f(\alpha,\beta)$; (2) Π is (ε,τ)-Alice-private, if no τ-time adversary who tries to impersonate Bob can distinguish between the different possible inputs Alice might hold with probability larger than $\varepsilon/2$ (see the earlier definition of IND-CPA security), and (3) Bob-privacy is defined by considering an ideal trusted party that gets the inputs f, β and α, and gives $f(\alpha,\beta)$ to Alice. We require in the real implementation that Alice does not get any information beyond the value of $f(\alpha,\beta)$. Due to the structure of AH two-message protocols, to prove Bob's privacy, it suffices to define a simulator that on inputs (f,α^*,β) generates an output distribution that is ε-close to Alice's view of the real protocol, where α^* is the input, actually submitted by Alice; i.e., α^* can be uniquely recovered from the first message of a protocol by just decrypting it. If such a simulator exists, then we say that the protocol is ε-*Bob-private*. We say that Π is $(\varepsilon,\tau;\varepsilon')$-*private* if it is (ε,τ)-Alice-private and ε'-Bob-private.

In a threshold AH two-message protocols, the secret key is owned by Chalice, most usually a coalition of servers. Alice encrypts her inputs by using Chalice's public key, forwards some ciphertexts to Bob, who applies some operations to them, and forwards the resulting ciphertexts to Chalice who threshold-decrypts them. This setting is usual in e-voting and e-auction protocols [CGS97,DJ01,LAN02]. In this case we also care about the correctness. See [Gol04] for the standard security definitions.

Computationally-private information retrieval and oblivious transfer. During a 1-*out-of-*ν *computationally-private information retrieval* (CPIR_ℓ^ν) protocol for ℓ-bit strings, Alice fetches β_α from the database $\beta=(\beta_1,\ldots,\beta_\nu)$ maintained by Bob, $\beta_i\in\{0,1\}^\ell$, so that computationally bounded Bob does not know which entry Alice is learning. In the following, we will also need the case where $\beta=(\beta_i)_{\beta_i\in\mathcal{S}}$ for an arbitrary public set $\mathcal{S}\subseteq\mathbb{Z}_n$. In this case, we call the resulting protocol a $\mathsf{CPIR}_\ell^{\mathcal{S}}$ *protocol*. Many two-message computationally-private information retrieval protocols (e.g., [Ste98,AIR01,Lip05]) are (return-to-sender) AH two-message protocols. The most efficient known AH two-message CPIR_ℓ^ν protocol by Lipmaa [Lip05] was until recently the most efficient CPIR_ℓ^ν at all. When based on the Damgård-Jurik length-flexible AH public-key cryptosystem [DJ01], Lipmaa's protocol has communication $(\log_2^2\nu+(s+\frac{3}{2})\cdot\log_2\nu+1)k$, where $k:=\log_2 n$ is the security parameter and $s:=\lceil\ell/k\rceil$. (For a non-length-flexible PKC, a close-to-polylogarithmic AH two-message protocol was proposed by Stern [Ste98].) Only recently, a more communication-efficient CPIR_ℓ^ν protocol, with communication $\Theta(\log\nu+\ell+k)$, was proposed by Gentry and Ramzan [GR05b]. Their protocol is not an AH protocol. Moreover, the Gentry-Ramzan protocol does not have polylogarithmic Alice-side computation that might be relevant in some applications.

A CPIR_ℓ^ν protocol is a *computationally Alice-private and statistically Bob-private* 1-*out-of-*ν *oblivious transfer protocol for* ℓ*-bit strings* (an OT_ℓ^ν protocol) if also Bob's privacy is guaranteed; $\mathsf{OT}_\ell^{\mathcal{S}}$ protocols are defined analogously. The next private AH two-message OT_ℓ^ν protocol was defined by Aiello, Ishai and Reingold [AIR01]. Let PKC be an IND-CPA secure homomorphic cryptosystem such that for all possible secret keys sk, $\mathcal{M}$ is a cyclic group with public prime order $|\mathcal{M}|\geq 2^\ell$. To obtain the element $\beta_\alpha\in\mathcal{M}$, Alice sends to Bob a random encryption $c=\mathsf{Enc}_{\mathsf{pk}}(\alpha;U(\mathcal{R}))$. For all $i\in[\nu]$, Bob replies with a random encryption of $(\alpha-i)U(\mathcal{M})+\beta_i$. Alice obtains $\beta_\alpha\leftarrow\mathsf{Dec}_{\mathsf{sk}}(c_\alpha)$. Unfortunately, the only well-known homomorphic public-key cryptosystem that works over large message spaces of prime order, ElGamal, is multiplicative.

Millionaire's problem. Millionaire's problem is as follows: given Alice's private input α and Bob's private input β from $\mathbb{Z}_{2^\ell}$, decide whether $\alpha>\beta$ without leaking anything else. Though there have been proposed numerous protocols for this problem (see, for example, [Fis01,BK04,ST04]), none of the proposals is completely satisfactory. For example, one of the most elegant previous millionaire's protocols, an additively homomorphic two-message protocol proposed by Blake and Kolesnikov [BK04], is Bob-private only in the semi-honest model.

3 A Seeded Randomness Extractor

We will need a seeded randomness extractor $\mathsf{Ext}:\mathbb{Z}_n\times S\to\mathbb{Z}_n$ where n is a large composite integer and S is a suitably chosen subset of $\mathbb{Z}_n$. The set of distributions $\mathcal{D}$ is defined as $\{U(p\mathbb{Z}_n):p\in\mathbb{Z}_n\setminus\{0\}\}$. Since we compute Ext on ciphertexts, it has to have the structure $\mathsf{Ext}(m,(s_0,s_1))=s_0m+s_1 \mod n$ for some set $S=\{(s_0,s_1)\}$ that is chosen so that Ext is an ε-extractor for $\mathcal{D}$ for as small ε as possible.

Theorem 1. *Let n be a positive integer, $0<\varepsilon\leq 1$, and let $\ell:=\lfloor\log_2\Phi(n)-\log_2(1/\varepsilon)+1\rfloor$. Denote $T:=\lfloor 2^{-\ell}n\rfloor$ and $S:=2^\ell\cdot\mathbb{Z}_T$. Let $\mathsf{Ext}(m,s):=m+s \mod n$ for $m\in\mathbb{Z}_n$ and $t\in S$. Then* Ext *is an ε-extractor for $\mathcal{D}$.*

For this theorem we first need the next technical lemma. (Here, $U(S) \mod p$ denotes the distribution that we get by first picking an element of $U(S)$ and then taking its remainder modulo p.)

Lemma 1. *Fix integers n and s, such that $s < n/2$ and $\gcd(s,n) = 1$. Set $T := \lfloor s^{-1}n \rfloor$. For any non-trivial factor p of n and for an arbitrary $m \in \mathbb{Z}_s$, $(m + U(s\mathbb{Z}_T)) \mod p \overset{\varepsilon}{\sim} U(\mathbb{Z}_p)$ for $\varepsilon \leq s/(2\Phi(n))$.*

Proof. Fix $m \in \mathbb{Z}_s$. Let p be a non-trivial factor of n. Then $T = ap + b$ for a non-negative a and for a $b \in [0, p-1]$. Since $\gcd(p,s) = 1$, then s is a generator of $\mathbb{Z}_p$. Therefore, we can partition $\mathbb{Z}_p$ into two sets $\mathcal{T}_0 = \{c \in \mathbb{Z}_p : \Pr[(m + U(s\mathbb{Z}_T)) \mod p = c] = a/T\}$ and $\mathcal{T}_1 = \{c \in \mathbb{Z}_p : \Pr[(m + U(s\mathbb{Z}_T)) \mod p = c] = (a+1)/T\}$. According to the definition of statistical difference, $(m + U(s\mathbb{Z}_T)) \mod p \overset{\varepsilon}{\sim} U(\mathbb{Z}_p)$, where $\varepsilon = \max\{\sharp\mathcal{T}_0 \cdot (1/p - a/T), \sharp\mathcal{T}_1 \cdot ((a+1)/T - 1/p)\}$. Since $b = T - ap$, $\sharp\mathcal{T}_0 = p - b$ and $\sharp\mathcal{T}_1 = b$, then $\varepsilon = \max\{((p-b)b)/(Tp), (b(p-b))/(Tp)\} = (b(p-b))/(Tp) \leq p/(4T)$. From $T = \lfloor n/s \rfloor \geq (n-s)/s \geq n/(2s)$ it follows that $\varepsilon \leq p/(4T) \leq n/(4\Phi(n)T) \leq s/(2\Phi(n))$. □

Proof (Thm. 1). Assume that $X \in \mathcal{D}$, i.e., that $X = p\mathbb{Z}_n$ for some $p \in \mathbb{Z} \setminus \{0\}$. If $\gcd(p,n) = 1$ then $p\mathbb{Z}_n = \mathbb{Z}_n$ and the claim follows. Otherwise, set $s := 2^\ell$. Therefore, as $\sharp p\mathbb{Z}_n = n/p$, $\Pr[\mathsf{Ext}(m, U(S)) \equiv y \pmod n] = p/n \cdot \Pr[\mathsf{Ext}(m, U(S)) \equiv y \pmod p] = p/n \cdot \Pr[m + U(S) \equiv y \pmod p]$ for any $y \in \mathbb{Z}_n$. Lem. 1 assures that $\mathsf{Dist}\,(\mathsf{Ext}(X, U(S)) \| U(\mathbb{Z}_n)) = \mathsf{Dist}\,(\mathsf{Ext}(X, U(S)) \mod p \| U(\mathbb{Z}_p)) = \mathsf{Dist}\,((m + U(S)) \mod p \| U(\mathbb{Z}_p)) \leq 2^{\ell-1}/(\Phi(n)) \leq \varepsilon$. □

4 AH Two-Message Protocol for Oblivious Transfer

We need an AH two-message oblivious transfer protocol. Given the state of the art in AH public-key cryptosystems, it means that this oblivious transfer protocol must work on groups of composite order. In [Lip03,Cha04], the authors have tried to generalise the Aiello-Ishai-Reingold protocol correspondingly. Lipmaa [Lip03] claimed that the Aiello-Ishai-Reingold protocol is a "weakly" Bob-private 1-out-of-ν-oblivious transfer protocol under the assumption that n is ν-rough; weak security meaning that a malicious Alice will, even in the case of incorrect inputs, obtain information about exactly one database element. Lipmaa's proof in [Lip03] is however faulty, because in the case of a malicious Alice, the Aiello-Ishai-Reingold protocol leaks partial information about s database elements, where s is the number of different prime factors of n. Namely, assume that $n = \prod p_i^{a_i}$ for different primes $p_1 < p_2 < \cdots < p_s$. If Alice's input α^* is such that $\alpha^* \equiv \alpha_i \mod p_i$ for some mutually different values $\alpha_i \in [\nu]$, then Alice can straightforwardly compute the values $\beta_{\alpha_i} \mod p_i$, $i \in [s]$, even if $\alpha \notin [\nu]$. Alice who knows how to factor n can therefore easily, by using the Chinese Remaindering Theorem, compute the required α^*.

The same observation underlies, in a constructive way, Chang's 2-out-of-ν-oblivious transfer protocol; [Cha04] actually proved that if n is a product of two safe primes then no more information than $\beta_{\alpha_i} \mod p_i$, $i \in [2]$, is revealed. Chang [Cha04] also proposed a 1-out-of-ν-oblivious transfer protocol; however, since there a honest Alice has to encrypt values that depend on the secret key and thus is unusable in the CDS protocol, proposed in Sect. 5.

Next, we propose a new 1-out-of-ν-oblivious transfer protocol (see Prot. 1) that achieves Bob-privacy. It is an extension of the Aiello-Ishai-Reingold OT_ℓ^ν protocol to the case where the order of the underlying group is composite but still sufficiently rough. In this protocol, we need a randomised function $\mathsf{encoding} : \{0,1\}^\ell \to \mathbb{Z}_n$, computable on ciphertexts, such that for any $p \in \mathbb{Z}_n \setminus \{0\}$ given in an encrypted form, any $\beta_1, \beta_2 \in \mathbb{Z}_n$, and for as small ε as possible, $\mathsf{encoding}(p, \beta_1) \overset{\varepsilon}{\sim} \mathsf{encoding}(p, \beta_2)$; while on the other hand, if the encrypted value is $p = 0$, then one can efficiently recover β given $\mathsf{encoding}(p, \beta)$.

Let $\mathcal{D}$ and $S = 2^\ell \cdot \mathbb{Z}_T$ be defined as in Sect. 3. Now, notice that for $p \in \mathbb{Z}_n$, $pU(\mathbb{Z}_n) \in \mathcal{D}$ if $p \neq 0$ and $pU(\mathbb{Z}_n) = 0$ otherwise. Therefore, it is quite straightforward to construct $\mathsf{encoding}$ by using a seeded randomness extractor on top of exponentiation with a random group element. More precisely, (a) we first multiply the encrypted value with a random element of $\mathbb{Z}_n$; (b) after that, we use a seeded ε-extractor Ext for $\mathcal{D}$, such that $\mathsf{Ext}(X, U(S)) \overset{\varepsilon}{\sim} U(\mathbb{Z}_n)$ for any $X \in \mathcal{D}$, while $\mathsf{Ext}(0, U(S)) \equiv 0 \mod 2^\ell$ for a relatively large ℓ. The extractor defined in Thm. 1 is sufficient. Thus, we will use an (again, relatively lightweight) randomised encoding $\mathsf{encoding}(p, \beta_0) := \mathsf{Ext}(pU(\mathbb{Z}_n), U(2^\ell \cdot \mathbb{Z}_T)) + \beta_0 \mod n = pU(\mathbb{Z}_n) + U(2^\ell \cdot \mathbb{Z}_T) + \beta_0 \mod n$. Now, we are ready to prove the next theorem:

Theorem 2. *Let $\mathsf{PKC} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an AH public-key cryptosystem that satisfies the next requirements: (a) it is (ε, τ)-IND-CPA secure, (b) $n := \sharp\mathcal{M}$ is $(2^{\ell+1}\nu/\varepsilon')$-rough for some $0 < \varepsilon' \leq 1$ with $\ell \leftarrow \lfloor \log_2 \Phi(n) - \log_2 \nu - \log_2(1/\varepsilon') + 1 \rfloor$, (c) $\mathcal{R}$ is a quasigroup, (d) $\mathcal{M}$ and $\mathcal{R}$ are efficiently samplable, and (e) membership in $\mathcal{C}$ can be efficiently verified. Assume that pk is valid. Then Protocol 1 is an $(\varepsilon, \tau - O(1); 2^\ell\nu/\Phi(n) \leq \varepsilon')$-private AH two-message protocol for OT_ℓ^ν.*

Common parameters: ℓ, $T := \lfloor 2^{-\ell} n \rfloor$, $\mathsf{PKC} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and pk.
Private input: Alice has sk and $\alpha \in \{1, \ldots, \nu\}$, Bob has a tuple $\beta_1, \ldots, \beta_\nu \in \{0,1\}^\ell$.
Private output: Alice obtains β_α.

First message, by Alice: Send $\mathsf{mq} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\alpha; U(\mathcal{R}))$ to Bob.
Second message, by Bob:
If $\mathsf{mq} \notin \mathcal{C}$ then return $\perp$.
// Now, mq is an encryption of some element α^*.
For $j \in [\nu]$: Let c_j be a random encryption of $(\alpha^* - j)U(\mathcal{M}) + U(2^\ell \cdot \mathbb{Z}_T) + \beta_j$.
// I.e., set $c_j \leftarrow (\mathsf{mq}/\mathsf{Enc}_{\mathsf{pk}}(j; *))^{U(\mathcal{M})} \cdot \mathsf{Enc}_{\mathsf{pk}}(U(2^\ell \cdot \mathbb{Z}_T) + \beta_j; U(\mathcal{R}))$, where $*$ is an arbitrary element of $\mathcal{R}$.
Return $(c_1, \ldots, c_\nu)$.
Postprocessing, by Alice: Return $\mathsf{Dec}_{\mathsf{sk}}(c_\alpha) \mod 2^\ell$.

Protocol 1: An AH two-message protocol for OT^ν_ℓ

Proof. First, if pk is valid then $\mathsf{Dec}_{\mathsf{sk}}(c_j)$ is distributed as $\mathsf{Ext}((\alpha^* - j)U(\mathbb{Z}_n), U(2^\ell \cdot \mathbb{Z}_T)) + \beta_j \mod n$, where $\alpha^* := \mathsf{Dec}_{\mathsf{sk}}(\mathsf{mq})$. *Correctness* is straightforward, since $\mathsf{Dec}_{\mathsf{sk}}(c_\alpha)$ is distributed as $\mathsf{Ext}((\alpha - \alpha)U(\mathbb{Z}_n), U(2^\ell \cdot \mathbb{Z}_T)) + \beta_\alpha = U(2^\ell \cdot \mathbb{Z}_T) + \beta_\alpha$. Since always $\mathsf{Dec}_{\mathsf{sk}}(c_\alpha) < n$ then $\mathsf{Dec}_{\mathsf{sk}}(c_\alpha) \equiv \beta_\alpha \mod 2^\ell$. *Computational Alice-privacy* follows directly from the IND-CPA security of PKC, since Bob sees only a random encryption of α. *Statistical Bob-privacy*: Assume that Bob is honest and that pk is valid. Then, for any j, $D_{\mathsf{sk}}(c_j)$ is distributed according to $V_j := \mathsf{Ext}((\alpha - j(U(\mathbb{Z}_n), U(2^\ell \cdot \mathbb{Z}_T)) + \beta_j$. Moreover, since $\mathcal{R}$ is a quasigroup then c_j is a *random* encryption of a random element from V_j and thus reveals no more information than a random element of V_j. Due to Thm. 1, $\mathsf{Dist}\,(V_j \| U(\mathbb{Z}_n)) \le 2^\ell/(2\Phi(n))$. As all ν distributions V_j are independent then $\mathsf{Dist}\,((V_1, \ldots, V_\nu) \| U(\mathbb{Z}_n^\nu)) \le \nu 2^\ell/(2\Phi(n)) \le \nu/(2\Phi(n)) \cdot \Phi(n)/\nu \cdot 2 \cdot \varepsilon' = \varepsilon'$. Here, the second inequality follows from (b). Thus, we can simulate Alice's view as follows: Compute sk that corresponds to pk. Set $\alpha^* \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathsf{mq})$, set $\mathsf{mt} = \perp$ if mq is not a valid ciphertext. For all $j \in [\nu]$ do: If $j \neq \alpha^*$, set $\hat{c}_j \leftarrow \mathsf{Enc}_{\mathsf{pk}}(U(\mathbb{Z}_n); U(\mathcal{R}))$. If $j = \alpha^*$, set $\hat{c}_j \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\mathsf{Ext}(0, U(2^\ell \cdot \mathbb{Z}_T)) + \beta_{\alpha^*}; U(\mathcal{R}))$. Output $\hat{c} = (\hat{c}_1, \ldots, \hat{c}_\nu)$. If adversary is semi-honest then the simulation is perfect, otherwise the statistical difference between $\hat{c}$ and the real view is less or equal than $\nu 2^\ell/(2\Phi(n))$. □

All well-known homomorphic public-key cryptosystems [El 84,OU98,NS98,Pai99,DJ01,DJ03] have the required properties. In all practical situations, we can assume that $\varepsilon' = 2^{-80}$ and $\nu \le 2^{40}$, then a 2^{120}-rough n is sufficient for Boolean inputs. If PKC is Paillier's cryptosystem then n is $\sqrt{n}/2$-rough, and consequently, one can take $\ell \leftarrow \lfloor \frac{1}{2} \cdot \log_2 n - \log_2 \nu - \log_2(1/\varepsilon') \rfloor$. For $\log_2 n = 1024$ and $\varepsilon' = 2^{-80}$, we get $\ell = \lfloor 433 - \log_2 \nu \rfloor \ge 393$. Finally, Protocol 1 can be straightforwardly modified to transfer $\ell' > \ell$ bits by repeating the second message of the proposed oblivious transfer protocol $\lceil \ell'/\ell \rceil$ times.

Corollary 1. *Let ε, ε', τ and ℓ be as in Thm. 2, and let $\mathcal{S} \subseteq \mathcal{M}$ be an arbitrary public index set. There exists an $(\varepsilon, \tau - O(1); \varepsilon')$-private AH two-message $\mathsf{OT}^{\mathcal{S}}_\ell$ protocol with communication $\Theta(\nu)$.*

Proof. As in Prot. 1, but let c_j be a random encryption of $\mathsf{Ext}((\alpha^* - h_j)U(\mathbb{Z}_n), U(2^\ell \cdot \mathbb{Z}_T)) + \beta_j$ for $h_j \in \mathcal{S}$, and set $\mathsf{mt} \leftarrow (c_j)_{j \in \mathcal{S}}$. □

Given an arbitrary $\mathsf{CPIR}^\nu_{\ell'}$ (resp., $\mathsf{CPIR}^{\mathcal{S}}_{\ell'}$) protocol with $\ell' > \log \sharp\mathcal{C}$, one can construct an efficient OT^ν_ℓ (resp., $\mathsf{OT}^{\mathcal{S}}_\ell$) protocol as follows: as in Protocol 1, Alice sends $\mathsf{Enc}_{\mathsf{pk}}(\alpha; r)$ to Bob, who computes the values c_i as in Prot. 1 (resp., Cor. 1) but without sending them to Alice. In parallel, Alice uses the $\mathsf{CPIR}^\nu_{\ell'}$ protocol to retrieve c_α. In particular, [Lip05] gives us the next result.

Corollary 2. *Let ε' and ℓ be as in Thm. 2. Let PKC be a length-flexible AH cryptosystem [DJ01] that satisfies the same properties as required in Thm. 2. Let $\mathcal{S} \subset \mathcal{M}$ be an arbitrary public index set with $\sharp\mathcal{S} = \nu$. There exists an AH two-message $(\varepsilon \cdot \log_2 \nu, \tau - \mathrm{polylog}(\nu); \varepsilon')$-private $\mathsf{OT}^{\mathcal{S}}_\ell$ protocol with communication $\Theta(k \cdot \log^2 \nu + \ell \cdot \log \nu)$, where k is a possibly non-constant security parameter.*

Proof. Correctness is obvious. *Alice-privacy* is the same as in Lipmaa's computationally-private information retrieval protocol from [Lip05]. *Bob-privacy* follows from Thm. 2. □

Lipmaa [Lip05] proved that applying the Aiello-Ishai-Reingold oblivious transfer protocol results in an oblivious transfer protocol with log-squared communication if one assumes both that PKC is IND-CPA secure and the Decisional Diffie-Hellman problem is hard. Thus, Cor. 2 achieves the same result but under a weaker security assumption.

For a non-length-flexible PKC, application of Cor. 1 to the CPIR_ℓ^ν protocol from [Ste98] gives an OT_ℓ^ν protocol with communication $\Theta(\ell \cdot 2^{\sqrt{\log \nu}} + k \cdot \sqrt{\log \nu} \cdot 2^{\sqrt{\log \nu}})$. Moreover, due to [GR05b], there exists a non-AH two-message $\mathsf{OT}_\ell^{\mathcal{S}}$ protocol with communication $\Theta(\log \nu + \ell + k)$, assuming both that PKC is IND-CPA secure and that Φ Hiding is hard.

On the optimality of the extractor. Recall the notation from Sect. 3. Assume that $\varepsilon \gg 1/n$ and $\varepsilon < 1 - 1/n$. All distributions $X \in \mathcal{D}$ have at least $\log_2 \Phi(n)$ bits of entropy, $H(X) \geq \log_2 \Phi(n)$, and for some $X \in \mathcal{D}$, $H(X) = \log_2 \Phi(n)$. Thus for every $X \in \mathcal{D}$, for $\mathsf{Ext}(X, U(S))$ to be ε-close to $U(\mathbb{Z}_n)$ and thus to have $H(\mathsf{Ext}(X, U(S))) \geq \log_2 n + \frac{2}{n} \cdot \log_2 n$, we need that $H(U(S)) \geq \log_2 n + \frac{2}{n} \cdot \log_2 n - \log_2 \Phi(n)$. Thus, for any fixed x and in particular for $x = 0$, $H(\mathsf{Ext}(x, U(S))) \geq \log_2 n + \frac{2}{n} \log_2 n - \log_2 \Phi(n)$. This means that if we transmit any element z from $\mathbb{Z}_n$ that is masked by an output of $\mathsf{Ext}(X, U(S))$ for some $X \in \mathcal{D}$ then one can only recover $\Phi(n) - \frac{2}{n} \log_2 n$ bits of z. We defer a longer discussion of the optimality to the full version of the paper. Therefore, Ext is quite close to the optimal. This is remarkable especially since Ext is so lightweight. (Compare this to the elaborated constructions in [GRS04,GR05a].)

5 AH Two-Message Protocol for Conditional Disclosure of Secrets

For the purposes of the current paper, a conditional disclosure of secrets protocol $\mathsf{CDS}_\ell^{\mathcal{S}}$ [AIR01] for a public set $\mathcal{S}$ and ℓ-bit inputs is a (return-to-sender) two-message protocol for the next functionality: $f(\alpha, \beta) = \beta$ if $\alpha \in \mathcal{S}$ and $f(\alpha, \beta)$ is a random element from some β-independent distribution otherwise. If the $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol is AH then Bob also obtains an encryption of α. The next result is straightforward:

Corollary 3. *Let Π be a $(\varepsilon, \tau; \varepsilon')$-private two-message protocol for $\mathsf{OT}_\ell^{\mathcal{S}}$. Then there exists a $(\varepsilon, \tau; \varepsilon')$-private two-message protocol Π' for $\mathsf{CDS}_\ell^{\mathcal{S}}$. If Π is AH then Π' is AH.*

Proof. Follows from Cor. 1 and Cor. 2 by executing $\mathsf{OT}_\ell^{\mathcal{S}}$ with database β', where $\beta'[i] = \beta$ for all $i \in \mathcal{S}$. □

As previously, one can base $\mathsf{CDS}_\ell^{\mathcal{S}}$ on the Gentry-Ramzan oblivious transfer protocol, given that the AH property is not required. Therefore, every public (efficiently computable) set $\mathcal{S}$ has a CDS protocol with $\Theta(\log \sharp\mathcal{S})$ communication. However, because Bob's computation in an $\mathsf{OT}_\ell^{\mathcal{S}}$ protocol is linear, then Bob's computation in the resulting $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol is also linear in $\sharp\mathcal{S}$. In some of the protocols, $\sharp\mathcal{S} = 2^\ell$ for ℓ defined as in Thm. 2, and thus Bob's computation becomes prohibitive.

Next, we will show how to use explicit circuit evaluation to construct, given any public $\mathcal{S} \subseteq \mathbb{Z}_{2^\ell}$, an AH two-message protocol for $\mathsf{CDS}_\ell^{\mathcal{S}}$ that is often computationally much more efficient. Namely, we specify the predicate $[\alpha \in \mathcal{S}]$ by a set of constraints on Alice's input α, where for the sake of efficiency, α might be broken down to several smaller inputs—e.g., bits—so that from them, Bob can recover the original encrypted inputs by using homomorphic operations. The constraints can be written down as a suitable *monotone Boolean formula $\Psi_{\mathcal{S}}$ with affine zero tests*, where we allow Boolean operations $\wedge$ and $\vee$ and affine zero tests $[\sum a_i \alpha_i \stackrel{?}{=} b]$. Here, the $\vee$ gates may have an arbitrary fan-in. We require that $\Psi_{\mathcal{S}}(x) = 0$ if and only if $x \notin \mathcal{S}$. We motivate the next discussion with the problem where $\mathcal{S}$ is equal to $\mathrm{GT}_\mu(y) := \{x \in \{0,1\}^\mu : x > y\}$, for μ-bit strings that are split into μ one-bit inputs. Writing $x = (x_{\mu-1}, \ldots, x_0)$,

$$\begin{aligned}\Psi_{\mathrm{GT}_\mu(y)}(x) :=& ([x_{\mu-1} \stackrel{?}{=} 1] \wedge [y_{\mu-1} \stackrel{?}{=} 0]) \vee \\ & ([x_{\mu-1} \stackrel{?}{=} y_{\mu-1}] \wedge [x_{\mu-2} \stackrel{?}{=} 1] \wedge [y_{\mu-2} \stackrel{?}{=} 0]) \vee \\ & ([x_{\mu-1} \stackrel{?}{=} y_{\mu-1}] \wedge [x_{\mu-2} \stackrel{?}{=} y_{\mu-2}] \wedge [x_{\mu-3} \stackrel{?}{=} 1] \wedge [y_{\mu-3} \stackrel{?}{=} 0]) \vee \cdots \vee \\ & ([x_{\mu-1} \stackrel{?}{=} y_{\mu-1}] \wedge [x_{\mu-2} \stackrel{?}{=} y_{\mu-2}] \wedge \cdots \wedge [x_1 \stackrel{?}{=} y_1] \wedge [x_0 \stackrel{?}{=} 1] \wedge [y_0 \stackrel{?}{=} 0]) \ .\end{aligned}$$

Circuit evaluation is done as follows (see Fig. 1, left): Construct a circuit where every internal node implements a Boolean operation and every leaf implements an affine zero test. Process the circuit recursively from top to bottom. Assign $\beta \in \{0,1\}^\ell$ to the output wire of the circuit. For every $\wedge$ gate ψ with secret β_ψ assigned to its output wire, pick $\beta_{\psi,1} \leftarrow U(\{0,1\}^\ell)$ and $\beta_{\psi,2} \leftarrow \beta_\psi - \beta_{\psi,1} \mod 2^\ell$, and assign $\beta_{\psi,1}$, $\beta_{\psi,2}$ to the two input wires of ψ. For every $\vee$ gate, just push the output secret downwards. The resulting AH two-message $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol consists of the three phases:

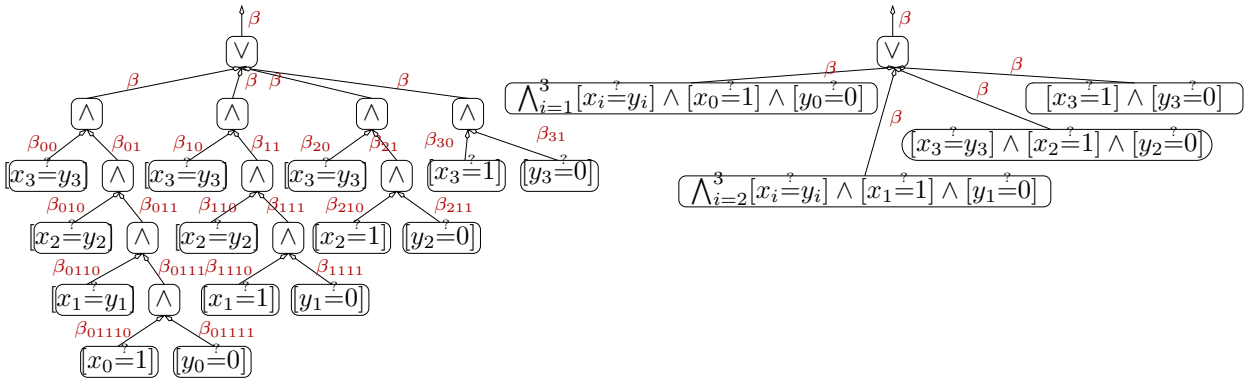

Fig. 1. Circuit for $\mathrm{GT}_4(y)$: unoptimised and optimised versions

Query phase: For every $i \in [\mu]$, Alice transfers one ciphertext $P_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\alpha_i; U(\mathcal{R}))$.

Transfer phase: For every leaf ψ with the corresponding affine zero test $[\sum_{i=1}^{\mu} a_i\alpha_i \stackrel{?}{=} b]$ and output secret β_ψ, Bob replies with a random encryption of $\mathsf{Ext}((\sum_{i=1}^{\mu} a_i\alpha_i - b)U(\mathcal{M}), U(2^\ell \cdot \mathbb{Z}_T)) + \beta_\psi$ as in Protocol 1.

Recovery phase: Alice decrypts ciphertexts that correspond to the correct branch in the circuit, and recovers β (modulo 2^ℓ).

Thus, Alice transfers μ ciphertexts and Bob transfers $\mathcal{L}(\Psi_{\mathcal{S}})$ ciphertexts, where $\mathcal{L}(\Psi_{\mathcal{S}})$ is the number of affine zero tests. Clearly, this protocol is correct. Following our motivating example, $\Psi_{\mathrm{GT}_\mu(y)}$ (see Fig. 1, left), we get a circuit with $\mathcal{L}(\Psi_{\mathrm{GT}_\mu(y)}) = \mu(\mu+3)/2$, and a $\mathsf{CDS}_\ell^{\mathrm{GT}_\mu(y)}$ protocol with communication $\mu(\mu+5)/2$ ciphertexts.

We can do better by allowing leaf gates that implement a conjunction of several zero tests. In such a circuit, β is propagated to the bottom as previously. The query phase remains unchanged. Let ψ be an arbitrary leaf gate that corresponds to the conjunction of v_ψ different zero tests, $\bigwedge_{j=1}^{v_\psi}[\sum_{i=1}^{\mu} a_{ij}\alpha_i \stackrel{?}{=} b_j]$; let β_ψ be the output secret of ψ. In the transfer phase, Bob replies with c_ψ, a random encryption of $\mathsf{Ext}(\sum_{j=1}^{v_\psi}(\sum_{i=1}^{\mu} a_{ij}\alpha_i - b_j)U(\mathcal{M}), U(2^\ell \cdot \mathbb{Z}_T)) + \beta_\psi$. Unless all the affine zero tests of ψ are satisfied, Alice learns nothing from c_ψ; in particular, Alice does not learn which zero tests fail. We call this protocol $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$. More precisely, let $\mathcal{L}_2(\Psi_{\mathcal{S}})$ be the number of leaves (that is, of conjunctive affine zero tests) in the latter circuit, and let $\mathsf{size}_2(\Psi(\mathcal{S}))$ be its size.

Theorem 3. *Let $\Psi_{\mathcal{S}} : \{0,1\}^\mu \to \{0,1\}$ be a public monotone Boolean formula with conjunctive affine zero tests. Let* PKC *be an AH public-key cryptosystem that satisfies the same requirements as required in Thm. 2, and* $\mathsf{Ext}(m, s) = m + s \mod n$ *be again the seeded randomness extractor with $\ell \leftarrow \lfloor \log_2 \Phi(n) - \log_2 \mathcal{L}_2(\Psi_{\mathcal{S}}) - \log_2(1/\varepsilon') + 1 \rfloor$. Then the $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ protocol is $(\mu \cdot \varepsilon, \tau - O(1); \varepsilon')$-private.*

Proof. Correctness is clear. *Alice-privacy* is also straightforward, since Bob sees only μ encryptions of 0's and 1's. *Bob-privacy*: the simulator Sim works as follows. First, it computes the secret key sk corresponding to pk. Then, it decrypts all inputs P_i and obtains the corresponding input α. Sim propagates $t = f(\alpha, \beta)$ down to the leaf level and computes the corresponding Bob's second messages. If Alice is honest then the simulation is perfect. Otherwise, the statistical difference between the replies is at most ε' as in the proof of Thm. 2. □

The communication of $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ is $\mu + \mathcal{L}_2(\Psi_{\mathcal{S}})$ ciphertexts, Alice's worst-case computation is $\Theta(\mu + \mathsf{size}_2(\Psi_{\mathcal{S}}))$ group operations and Bob's worst-case computation is $O(\mu \cdot \mathcal{L}(\Psi_{\mathcal{S}}))$ group operations. In particular, this means that if $\Psi_{\mathcal{S}}$ has a polynomial in $\log_2 \sharp\mathcal{S}$ number of gates then $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ has computation and communication that is polynomial in $\log_2 \sharp\mathcal{S}$. Therefore, all languages $\mathcal{S}$ in $\mathbf{P/poly}$ have a family of $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocols with polynomial communication and computation. Since all affine transformations can be presented by using polynomial circuits then the $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ protocol has polynomial resources if and only if $\mathcal{S} \in \mathbf{P/poly}$. This can be compared to the fact that it is only known how to compute on ciphertexts functions from $\mathbf{NC^1}$ [SYY99].

The use of conjunctive affine zero tests helps one often decrease the degree of the polynomial in question. Going back to the motivating example, $\mathcal{L}_2(\Psi_{\mathrm{GT}_\mu(y)}) = \mu$. (See Fig. 1, right. Here, Bob conditionally transfers the same secret μ times.) Therefore, under the same assumptions as in Thm. 3, there exists an $(\mu \cdot \varepsilon, \tau - O(1); \varepsilon')$-private AH

two-message protocol for $\mathsf{CDS}_\ell^{\mathrm{GT}_\mu(y)}$, with the communication of 2μ ciphertexts, logarithmic (in $\sharp\mathcal{S} = 2^\mu$) Alice's computation and log-squared (in $\sharp\mathcal{S} = 2^\mu$) Bob's computation.

Finally, note that the $\mathsf{CircuitCDS}\mathcal{S}\ell$ protocol per se can also be used on top of the additive version of El Gamal public-key cryptosystem.

6 Applications of CDS Protocol

Multiplicative relationships and polynomial arithmetic. A recent paper by Kissner and Song on privacy-preserving set operations [KS05] but also several previous papers like [FNP04,KM05] use AH two-message protocols in a setting where one encrypts the coefficients of some polynomials, where the important quantity is the set of roots of this polynomial. For example, if S_1 is the set of roots of $f_1(x)$ and S_2 is the set of roots of $f_2(x)$ then $S_1 \cup S_2$ is the set of roots of $f_1(x) \cdot f_2(x)$. In such situations one has the next problem: given $\mathsf{Enc}_{\mathsf{pk}}(x;\cdot)$, $\mathsf{Enc}_{\mathsf{pk}}(y;\cdot)$ and $\mathsf{Enc}_{\mathsf{pk}}(z;\cdot)$ for $x, y \in \{0,1\}^{\ell/2}$ and $z \in \{0,1\}^\ell$, encrypted by Alice, Alice must obtain the correct answer only if $z = xy$. Now, by the long multiplication rule, $z = xy$ if and only if $z = \sum_{i=0}^{\ell/2-1} xy_i 2^i$. Therefore, $\Psi_{[z=xy]}$ is a conjunction of the next tests, where d_i are auxiliary encrypted values: (1) $z_i \in \{0,1\}$ for $i \in \mathbb{Z}_\ell$, (2) $x_i \in \{0,1\}$ for $i \in \mathbb{Z}_{\ell/2}$, (3) $[(y_i = 0 \wedge d_i = 0) \vee (y_i = 1 \wedge d_i = x)]$ for $i \in \mathbb{Z}_{\ell/2}$, and (4) $[z = \sum_{i=1}^{\ell/2-1} d_i \cdot 2^i]$, where $z \leftarrow \sum_{i=0}^{\ell-1} z_i \cdot 2^i$ and $x \leftarrow \sum_{i=0}^{\ell/2-1} x_i \cdot 2^i$. Thus, Alice's communication is 2.5ℓ ciphertexts and Bob's communication is $\mathcal{L}_2(\Psi_{[z=xy]}) \le 4\ell + 1$ ciphertexts. Therefore, the total communication is $6.5\ell + 1$ ciphertexts. But then we can also construct a CDS transformation for the multiplication of polynomials, since the ith coefficient of fg is a sum of the products of the coefficients of f and g. Then, e.g., we can verify that for some sets X, Y and Z, where X, Y and Z are represented as the set of roots of some polynomials, it holds that $X \cup Y = Z$.

Two-message millionaire's protocol with logarithmic communication. A slight modification of $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ protocol of Sect. 5 can be used in the case of some private sets $\mathcal{S}$ that depend on β. More precisely, assume that Alice has a private input α and that Server has a private input $\beta \in \{0,1\}^\ell$ and that the $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ protocol is written down in a disjunctive normal form over affine zero tests, $\Psi_{\mathcal{S}} = \bigvee_{i=1}^{\lambda} \bigwedge_{j=1}^{n_i} [\sum_{i=1}^{\mu} a_i \alpha_i \stackrel{?}{=} b]$. Now, modify the $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ protocol as follows. Fix a *public* value t (for example, $t = 0$) and push it down the circuit. For every leaf gate ψ, let Bob to compute c_ψ as previously, but return the values c_ψ in a random order. Be careful to do that so that the number of accepting leaf gates is always either 0 (if $\Psi_{\mathcal{S}} = 0$) or some non-zero constant (if $\Psi_{\mathcal{S}} = 1$); this can be done efficiently for many interesting sets $\mathcal{S}$. Therefore, by testing that at least one of the ciphertexts c_ψ encrypts 0, Alice gets to know whether $\Psi_{\mathcal{S}}(\alpha, \beta)$ is true or not. Since $\mathcal{L}_2(\Psi_{\mathrm{GT}_\mu(y)}) = \mu$, we get a new two-message protocol for millionaire's problem of μ-bit strings that is secure against malicious adversaries, with communication of 2μ ciphertexts, Alice's computation $\Theta(\mu)$ and Bob's computation $\Theta(\mu^2)$, assuming only that the underlying AH public-key cryptosystem is IND-CPA secure.

Conditional oblivious transfer. A *conditional oblivious transfer (*$\mathsf{COT}_\ell^{\mathcal{S}}$*)* protocol [DOR99] is a protocol where Bob has a private input (β_1, β_2). At the end of the protocol, Alice obtains β_2 only if $\Psi_{\mathcal{S}}(\alpha, \beta_1) = 1$ for some public set $\mathcal{S}$ of valid Alice's and Bob's input pairs, and no information, otherwise. To implement $\mathsf{COT}_\ell^{\mathcal{S}}$, we use the same idea as in the case of the millionaire's problem with only one modification: the secret to push down the circuit is $t = 0^L || \beta_2$, where say $L = 80$. This approach works for sets $\mathcal{S}$ that have an efficient implementation for formula $\Psi_{\mathcal{S}}$.

Electronic voting and auctions without random oracles. Conditional disclosure of secrets can also be used to guarantee correctness in the case of threshold AH two-message protocols. As in [BGN05], consider an electronic voting protocol where every voter sends an AH encryption $c_i \leftarrow E_{\mathsf{pk}}(v_i; U(\mathcal{R}))$ to talliers. We assume that the protocol is correct if $v_i \in \mathsf{Valid}$ for some publicly known set Valid; this is true in typical AH e-voting protocols [DJ01]. Now, in the original protocols, it is usually assumed that every voter accompanies his or her vote with a non-interactive zero-knowledge proof that $v_i \in \mathsf{Valid}$. Instead, the talliers can jointly apply the CDS protocol, with output secret 0, to c_i (this can be done very efficiently if Valid is the set of powers of a fixed integer) and then threshold-decrypt the result. If the plaintext is equal to 0, talliers accept the vote as correct. Of course, every step of the talliers has to be accompanied by a zero-knowledge proof of correctness (to each other and to every possible outside observer), but since the number of talliers is significantly smaller than the number of voters, this is doable in practice. See [BGN05] for discussion. As a result, we get a voter-private, universally verifiable and robust e-voting scheme only assuming that there exists an IND-CPA secure AH public-key cryptosystem (and in particular, without using random oracles), where the voters have to only perform one encryption. One can use the same trick to eliminate the need for random oracles in the AH electronic auction scheme of [LAN02] and in many other protocols of similar vein. Compared to the protocols from [BGN05], our protocols are more efficient in the case of multi-candidate elections (this is since [BGN05] allows

Common parameters: ℓ, $T := \lfloor 2^{-\ell} n \rfloor$, $\mathsf{PKC} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, pk, Π, Π^{cds}.
Private input: Alice has inputs sk and $\alpha = (\alpha_1, \ldots, \alpha_\mu)$, Bob has inputs $\beta = (\beta_1, \ldots, \beta_\nu)$.
Private output: Alice obtains $(\delta_1, \ldots, \delta_\lambda) = f(\alpha, \beta)$ where $\delta_j \in \{0,1\}^\ell$.

Alice's first message:
For $i \in [\mu]$: Let P_i be Alice's first Π-message on input α_i.
Send $(P_1, \ldots, P_\mu)$ to Bob.
Bob's second message:
For $j \in [\lambda]$:
Compute $\widehat{t}_j \leftarrow U(\mathbb{Z}_{n-2^\ell})$ and $t_j \leftarrow \widehat{t}_j \mod 2^\ell$.
Compute the set of ciphertexts $\{c_{ij}\}$ from the output secret t_j as in Π^{cds}.
Compute Δ_j as in Π.
Set $\Delta'_j \leftarrow \Delta_j \cdot \mathsf{Enc}_{\mathsf{pk}}(t_j; *)$ for an arbitrary $* \in \mathcal{R}$.
Send $(\Delta'_1, \ldots, \Delta'_\lambda; \{c_{i1}\}, \ldots, \{c_{i\lambda}\})$ to Alice.
Recovery:
For $j \in [\lambda]$: Alice recovers t_j from $(\alpha, \mathsf{mq}, \{c_{ij}\})$ by using the recovery algorithm of Π^{cds}.
She recovers δ'_j from $(\alpha; \mathsf{mq}; \Delta'_j)$ by using the recovery algorithm of Π.
She sets $\delta_j \leftarrow \delta'_j - t_j \mod 2^\ell$.
Return $(\delta_1, \ldots, \delta_\lambda)$.

Protocol 2: Private computation of a function f in malicious model by using additive CDS transformation

to efficiently decrypt only if the plaintext is small) and is based on an incomparable (but may be a somewhat more standard) security assumption.

7 CDS Transformation

In this section, we present a generic transformation from private in the semi-honest model AH two-message protocols to private in the malicious model AH two-message protocols. It can be called as a *compiler* since this transformation can be constructed in a relatively automatic manner. It can also be used as a subprotocol in many-message protocols. More precisely, fix an AH two-message protocol Π. Denote Alice's input by $\alpha = (\alpha_1, \ldots, \alpha_\mu)$, Bob's input by $\beta = (\beta_1, \ldots, \beta_\nu)$ and Alice's output by $\delta = (\delta_1, \ldots, \delta_\lambda)$. Here, w.l.o.g., we assume that α_i, β_i and δ_i belong to $\{0,1\}^\ell$, where ℓ is defined as in Thm. 2. Larger inputs and outputs can be handled straightforwardly. The query phase consists of sending the elements $\mathsf{Enc}_{\mathsf{pk}}(\alpha_i; r_i)$ and the transfer phase consists of sending the elements $\mathsf{Enc}_{\mathsf{pk}}(\delta_j; r'_j)$ for some r_i and r'_j. We assume that α, β and δ have already been modified to facilitate efficient circuit evaluation. For example, in the case of $\mathrm{GT}_\mu(y)$, every α_i is a bit. Assume that the input α of an honest Alice belongs to some publicly known set Valid that in particular does not depend on the value of sk. Most of the known AH two-message protocols have this property, Chang's OT^ν_ℓ protocol [Cha04] being one of the few exceptions. To simplify the implementations, we assume that if $\alpha \notin \mathsf{Valid}$ then for any input value of an honest Bob, $f(\alpha, \beta)$ is defined to be a uniformly random value from some fixed set.

Let Π an AH two-message protocol for function f with λ outputs from $\{0,1\}^\ell$, and let Π^{cds} be an AH two-message protocol for $\mathsf{CDS}^{\mathcal{S}}_\ell$, where ℓ is as defined in Thm. 2. The idea is to compose an instantiation of Π with an instantiation Π^{cds}, on the same inputs α and β, as follows. Assume that the query phase of both Π and Π^{cds} is the same; this is possible since in the previously constructed AH two-message protocol for $\mathsf{CDS}^{\mathsf{Valid}}_\ell$, Alice learns a secret $t \in \{0,1\}^{\ell\lambda}$ if and only if $\alpha \in \mathsf{Valid}$, and Bob learns the corresponding ciphertexts $P_i = \mathsf{Enc}_{\mathsf{pk}}(\alpha_i; U(\mathcal{R}))$. Therefore, in the transfer phase, Bob can use the ciphertexts P_i as input to an AH two-message protocol Π that evaluates f. Finally, Bob masks the outputs $(\Delta_1, \ldots, \Delta_\lambda)$ of Π with sub-secrets $(t_1, \ldots, t_\lambda)$, $t_i \in \{0,1\}^\ell$, and sends the corresponding encryptions $\Delta_i \cdot \mathsf{Enc}_{\mathsf{pk}}(t_i; U(\mathcal{R}))$ to Alice. Thus, Alice can peel off the masks t_i and recover the outputs if and only if her inputs are in the correct range.

Theorem 4 (Additive CDS transformation). *Fix an AH public-key cryptosystem* $\mathsf{PKC} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. *Fix a concrete* valid *secret and public key pair* $(\mathsf{sk}, \mathsf{pk})$. *Let* Π^{cds} *be an* $(\varepsilon, \tau; \varepsilon'_1)$*-private AH two-message protocol for* $\mathsf{CDS}^{\mathsf{Valid}}_\ell$. *Let* Π *be an* $(\varepsilon, \tau; \varepsilon'_2)$*-private AH two-message protocol for computing* f *in the semi-honest model, such that* Π^{cds} *and* Π *have a common algorithm for computing Alice's first message. Then Prot. 2 is an* $(\varepsilon, \tau - O(1); \varepsilon'_1 + \varepsilon'_2 + \varepsilon'_3)$*-private AH two-message protocol for computing* f *in the malicious mode, where* $\varepsilon'_3 = 2^\ell \lambda / n$.

Proof. Correctness: If $\alpha \in \mathsf{Valid}$ and both parties follow the protocol then recovery phase of the $\mathsf{CDS}_\ell^{\mathsf{Valid}}$ protocol is successful, Alice obtains $\widehat{t}_j \mod 2^\ell$ and consequently the correct end-result, as there are no modular wrappings. *Alice-privacy*: Consider an adversary B^* that obtains advantage ε against Prot. 2; B^* can then used to break the Π^{cds} protocol, since the query phase is exactly the same. For the same reason, Prot. 2 cannot be more Alice-private than Π. *Bob-privacy*: Clearly, Π' is a parallel execution of two statistically Bob-private AH two-message protocols. (The following does not necessarily hold in the case of computationally Bob-private protocols.) Therefore, Π' is an $(\varepsilon'_1+\varepsilon'_2)$-Bob-private AH two-message protocol for $\widehat{f}$ defined as $\widehat{f}_j(\alpha,\beta) = (\delta_j + t_j, t_j)$, if $\alpha \in \mathsf{Valid}$, and $\widehat{f}_j(\alpha,\beta) = (\delta_j + t_j, \perp)$, if $\alpha \notin \mathsf{Valid}$. The claim follows as t_j are almost random plaintexts and $\varepsilon'_3/\lambda = \mathsf{Dist}\,(U(\mathbb{Z}_{n-2^\ell}) \| U(\mathbb{Z}_n)) = 2^\ell/n$. □

With a slight modification (setting $\widehat{t}_j \leftarrow U(\mathbb{Z}_n)$ and using the CDS on a $2^{\ell+1}$ bit secret t_j where one bit indicates that $t_j \geq n - 2^\ell$), one can remove the addend ε'_3. Note that this theorem does not require PKC to be an AH public-key cryptosystem; with a small modification, the same proof goes through also with a multiplicatively homomorphic public-key cryptosystem.

Optimisations. The communication overhead of the CDS transformation is linear in the number of outputs. Therefore, it is not advantageous to use the transformation for functions with many outputs (e.g., private matrix operations). However, if computational Bob-privacy is sufficient, one can use an arbitrary pseudo-random function prf to stretch the transformation's secret to privately implement the function $\widehat{f}$ where $\widehat{f}_j(\alpha,\beta) = (\delta_j + \mathsf{prf}(t,j), t)$ if $\alpha \in \mathsf{Valid}$ and $\widehat{f}_j(\alpha,\beta) = (\delta_j + \mathsf{prf}(t,j), \perp)$ if $\alpha \notin \mathsf{Valid}$, for a single random key t. Such a protocol remains computationally Bob-private as long as prf is secure.

An example application: private scalar product protocol. Assume that Alice has a Boolean vector α of dimension μ and Bob has a Boolean vector β of the same dimension. In a *private scalar product protocol* protocol, Alice's private output is δ, such that $\delta = \sum_{i=1}^{\mu} \beta_i \alpha_i$, and Bob has no private output. It is simple to compute this functionality in the semi-honest model. Assume that c_i is a random encryption of α_i. Then, Bob sends $\Delta = \sum_{i=1}^{\mu} c_i^{\beta_i} \cdot \mathsf{Enc}_{\mathsf{pk}}(0; U(\mathcal{R}))$ to Alice. Alice decrypts Δ. It is straightforward to apply the CDS transformation to get a protocol that is Bob-private in the malicious model. This protocol also computes the private set intersection. Similar ideas can be used to construct private protocols for many other related problems (e.g., matrix-to-vector multiplication and other similar problems from linear algebra).

8 Comparison with Related Work

A well-known alternative to the additive CDS transformation is to let Alice to prove in zero-knowledge that mq encrypts a value from Valid; this means that either the resulting protocol takes at least three messages or that the protocol is only secure in the common reference string (or random oracle) model. As we have shown, one can use a mixture of arithmetic and Boolean formulas to construct an efficient AH two-message protocol for $\mathsf{CDS}_\ell^{\mathcal{S}}$. Similar efficiency can be achieved by using non-interactive zero-knowledge proofs, but compared to them, the additive CDS transformation uses simpler basic components. The difference in efficiency comes from the use of a oblivious transfer instead of a zero-knowledge disjunctive proof [CDS94]: the first can be done in the complexity-theoretic model very efficiently, while non-interactive zero-knowledge proofs are not possible in the complexity-theoretic model, and are somewhat more complex to implement in the random oracle model.

Compared to the CDS transformation from [AIR01] applied together with multiplicatively homomorphic PKC, additive CDS transformation is applicable in a wider setting since there exist many efficient protocols that are crucially based additively homomorphic public-key cryptosystems. Using the transformation from [AIR01] in these cases is either impossible or requires one to rely on the Decisional Diffie-Hellman assumption (recall Cor. 2) in addition to the assumption that PKC is IND-CPA secure.

Recently, Boneh, Goh and Nissim [BGN05] proposed a PKC where one can efficiently compute 2-DNF formulas on the ciphertexts. Given such a PKC, it is straightforward to define a CDS protocol for two-element sets. Unfortunately, their PKC has two unsuitable properties. First, to decrypt a ciphertext, one has to compute discrete logarithm (in this sense, their cryptosystem is similar to the additive version of the El Gamal cryptosystem). Therefore, while their "input verification gadget" can be used as a CDS protocol, their PKC is only usable if the output values of the protocol are not too large. Second, their PKC has composite plaintext order. Paradoxically, slow decryption actually makes their protocols secure modulo a composite integer. However, in the case when one would have a 2-DNF homomorphic public key cryptosystem with efficient decryption that works over plaintext groups of composite order, one should use an encoding method similar to ours.

*****Acknowledgements. We would like to thank Vladimir Kolesnikov and anonymous reviewers for useful comments. The work was partially supported by the Finnish Academy of Sciences and by the Estonian Science Foundation, grant 6096.

References

[AIR01] William Aiello, Yuval Ishai, and Omer Reingold. Priced Oblivious Transfer: How to Sell Digital Goods. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 119–135, Innsbruck, Austria, May 6–10, 2001. Springer-Verlag.

[BGN05] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF Formulas on Ciphertexts. In Joe Kilian, editor, *The Second Theory of Cryptography Conference, TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 325–341, Cambridge, MA, USA, February 10–12, 2005. Springer Verlag.

[BK04] Ian F. Blake and Vladimir Kolesnikov. Strong Conditional Oblivious Transfer and Computing on Intervals. In Pil Joong Lee, editor, *Advances on Cryptology — ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 515–529, Jeju Island, Korea, December 5-9 2004. Springer-Verlag.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In Yvo G. Desmedt, editor, *Advances in Cryptology—CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 174–187, Santa Barbara, USA, August 21–25 1994. Springer-Verlag.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In Walter Fumy, editor, *Advances in Cryptology — EUROCRYPT '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 103–118, Konstanz, Germany, 11–15 May 1997. Springer-Verlag.

[Cha04] Yan-Cheng Chang. Single Database Private Information Retrieval with Logarithmic Communication. In Josef Pieprzyk and Huaxiong Wang, editors, *The 9th Australasian Conference on Information Security and Privacy (ACISP 2004)*, volume 3108 of *Lecture Notes in Computer Science*, pages 50–61, Sydney, Australia, July 13–15, 2004. Springer-Verlag.

[DJ01] Ivan Damgård and Mads Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In Kwangjo Kim, editor, *Public Key Cryptography 2001*, volume 1992 of *Lecture Notes in Computer Science*, pages 119–136, Cheju Island, Korea, February 13–15, 2001. Springer-Verlag.

[DJ03] Ivan Damgård and Mads Jurik. A Length-Flexible Threshold Cryptosystem with Applications. In Rei Safavi-Naini, editor, *The 8th Australasian Conference on Information Security and Privacy*, volume 2727 of *Lecture Notes in Computer Science*, pages 350–364, Wollongong, Australia, July 9-11, 2003. Springer-Verlag.

[DOR99] Giovanni Di Crescenzo, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. Conditional Oblivious Transfer and Timed-Release Encryption. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 74–89, Prague, Czech Republic, May 2–6, 1999. Springer-Verlag.

[El 84] Taher El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology: Proceedings of CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18, Santa Barbara, California, USA, August 19–22, 1984. Springer-Verlag, 1985.

[FIPR05] Michael J. Freedman, Yuval Ishai, Benny Pinkas, and Omer Reingold. Keyword Search and Oblivious Pseudorandom Functions. In Joe Kilian, editor, *The Second Theory of Cryptography Conference, TCC 2005*, volume 3378 of *Lecture Notes in Computer Science*, pages 303–324, Cambridge, MA, USA, February 10–12, 2005. Springer Verlag.

[Fis01] Marc Fischlin. A Cost-Effective Pay-Per-Multiplication Comparison Method for Millionaires. In David Naccache, editor, *Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 457–472, San Francisco, CA, USA, 8–12 April 2001. Springer-Verlag. ISBN 3-540-41898-9.

[FNP04] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient Private Matching and Set Intersection. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology — EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19, Interlaken, Switzerland, May 2–6, 2004. Springer-Verlag.

[GIKM00] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. Protecting Data Privacy in Private Information Retrieval Schemes. *Journal of Computer and System Sciences*, 60(3):592–629, June 2000.

[GLLM04] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikäinen. On Secure Scalar Product Computation for Privacy-Preserving Data Mining. In Choonsik Park and Seongtaek Chee, editors, *Information Security and Cryptology - ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 104–120, Seoul, Korea, December 2–3, 2004. Springer-Verlag.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*. Cambridge University Press, 2004.

[GR05a] Ariel Gabizon and Ran Raz. Deterministic Extractors for Affine Sources over Large Fields. In *46th Annual Symposium on Foundations of Computer Science*, pages 407–418, Pittsburgh, PA, USA, October, 22–25 2005. IEEE, IEEE Computer Society Press.

[GR05b] Craig Gentry and Zulfikar Ramzan. Single-Database Private Information Retrieval with Constant Communication Rate. In Luis Caires, Guiseppe F. Italiano, Luis Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *The 32nd International Colloquium on Automata, Languages and Programming, ICALP 2005*, volume 3580 of *Lecture Notes in Computer Science*, pages 803–815, Lisboa, Portugal, 2005. Springer-Verlag.

[GRS04] Ariel Gabizon, Ran Raz, and Ronen Shaltiel. Deterministic Extractors for Bit-Fixing Sources by Obtaining an Independent Seed. In *45th Annual Symposium on Foundations of Computer Science*, pages 394–403, Rome, Italy, October, 17–19 2004. IEEE, IEEE Computer Society Press.

[KM05] Aggelos Kiayias and Antonina Mitrofanova. Testing Disjointness of Private Datasets. In Andrew Patrick and Moti Yung, editors, *Financial Cryptography and Data Security — Ninth International Conference*, volume 3570 of *Lecture Notes in Computer Science*, pages 109–124, Roseau, The Commonwealth Of Dominica, February 28–March 3, 2005. Springer-Verlag.

[KS05] Lea Kissner and Dawn Song. Privacy-Preserving Set Operations. In Victor Shoup, editor, *Advances in Cryptology — CRYPTO 2005, 25th Annual International Cryptology Conference*, volume 3621 of *Lecture Notes in Computer Science*, pages 241–257, Santa Barbara, USA, August 14–18, 2005. Springer-Verlag.

[LAN02] Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure Vickrey Auctions without Threshold Trust. In Matt Blaze, editor, *Financial Cryptography — Sixth International Conference*, volume 2357 of *Lecture Notes in Computer Science*, pages 87–101, Southhampton Beach, Bermuda, March 11–14, 2002. Springer-Verlag.

[Lip03] Helger Lipmaa. Verifiable Homomorphic Oblivious Transfer and Private Equality Test. In Chi Sung Laih, editor, *Advances on Cryptology — ASIACRYPT 2003*, volume 2894 of *Lecture Notes in Computer Science*, pages 416–433, Taipei, Taiwan, November 30–December 4, 2003. Springer-Verlag.

[Lip05] Helger Lipmaa. An Oblivious Transfer Protocol with Log-Squared Communication. In Jianying Zhou and Javier Lopez, editors, *The 8th Information Security Conference (ISC'05)*, volume 3650 of *Lecture Notes in Computer Science*, pages 314–328, Singapore, September 20–23, 2005. Springer-Verlag.

[NS98] David Naccache and Jacques Stern. A New Public Key Cryptosystem Based on Higher Residues. In *5th ACM Conference on Computer and Communications Security*, pages 59–66, San Francisco, CA, USA, 3–5 November 1998. ACM Press.

[OU98] Tatsuaki Okamoto and Shigenori Uchiyama. A New Public-Key Cryptosystem as Secure as Factoring. In Kaisa Nyberg, editor, *Advances in Cryptology — EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 308–318, Helsinki, Finland, May 31 – June 4 1998. Springer-Verlag.

[Pai99] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238, Prague, Czech Republic, May 2–6, 1999. Springer-Verlag.

[ST04] Berry Schoenmakers and Pim Tuyls. Practical Two-Party Computation Based on the Conditional Gate. In Pil Joong Lee, editor, *Advances on Cryptology — ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 119–136, Jeju Island, Korea, December 5-9 2004. Springer-Verlag.

[Ste98] Julien P. Stern. A New and Efficient All or Nothing Disclosure of Secrets Protocol. In Kazuo Ohta and Dingyi Pei, editors, *Advances on Cryptology — ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 357–371, Beijing, China, October 18–22, 1998. Springer-Verlag.

[SYY99] Tomas Sander, Adam Young, and Moti Yung. Non-Interactive CryptoComputing For NC^1. In *40th Annual Symposium on Foundations of Computer Science*, pages 554–567, New York, NY, USA, 17–18 October 1999. IEEE Computer Society.

[WY04] Rebecca N. Wright and Zhiqiang Yang. Privacy-Preserving Bayesian Network Structure Computation on Distributed Heterogeneous Data. In *Proceedings of The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 713–718, Seattle, Washington, USA, August 22–25 2004. ACM.