Additive Conditional Disclosure of Secrets And Applications*

***** Manuscript. August 8, 2006 *****

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 for set $\mathcal{S}$, the receiver obtains sender's secret iff receiver's input to the protocol belong to $\mathcal{S}$. By constructing a new seeded randomness extractor, we extend the CDS protocol to work over additively homomorphic cryptosystems and construct a CDS protocol for every set from $\mathbf{NP/poly}$. Some of the proposed applications are a oblivious transfer protocol with log-squared communication and a millionaire's protocol with logarithmic communication. 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. The only hardness assumption in all these protocols is that the underlying public-key cryptosystem is IND-CPA secure.

Keywords. Conditional disclosure of secrets, oblivious transfer, two-party computation.

1 Introduction

In a *homomorphic two-message protocol*, the receiver forwards some homomorphically encrypted ciphertexts to the sender who, after computing on ciphertexts, sends some ciphertexts to Calista, who can be either the receiver or a coalition of third parties for decryption. The underlying cryptosystem may be either multiplicatively homomorphic (MH) like Elgamal or additively homomorphic (AH) like Paillier. MH two-message protocols—that run over a MH cryptosystem—often have limited applicability, because there the receiver usually has to compute a discrete logarithm to recover the protocol outcome. Thus, MH two-message protocols are usually only feasible if the protocol outcome is "small" (e.g., in two-candidate e-voting protocols). If instead an AH cryptosystem PKC [Pai99,DJ01] is used then one can omit the costly discrete logarithm computing step. Thus, AH two-message protocols—that run over an AH cryptosystem–can used when one needs exponentially larger output space and thus also, input space. Efficient AH two-message protocols exist for computationally-private information retrieval (CPIR, [AIR01,Ste98,Lip05]), millionaire's problem [BK04], and various privacy-preserving data mining tasks (e.g., private scalar product [WY04,GLLM04], private set intersection cardinality).

Since in two-message homomorphic protocols, only one party obtains an output, one is only interested in *relaxed-security*—i.e., receiver-privacy and sender-security in the malicious model—, leaving full receiver-security to an upper level protocol. See, e.g., [AIR01,FIPR05] for a fuller explanation. Computational receiver-privacy of an AH two-message protocol in the malicious model follows from the IND-CPA security of PKC. However, if the receiver encrypts invalid inputs then he either can obtain extra information or attack the correctness of the protocol. To avoid this, the receiver is usually required to prove in zero-knowledge that his inputs are valid. Unfortunately, this either increases the number of messages or requires a security model with non-complexity-theoretic assumptions, e.g., common reference string (CRS) or random oracles. While CRS is a plausible assumption in protocol design, the current non-interactive zero-knowledge protocols for NP in the CRS model are always not really practical.

Conditional disclosure of secrets $\mathsf{CDS}_\ell^{\mathcal{S}}$ [GIKM00,AIR01], also known as input verification gadget [BGN05], offers an alternative. In the $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol, Calista obtains sender's input iff the receiver encrypted an element from the set $\mathcal{S} \in \{0,1\}^\ell$. In parallel and in another protocol, the sender uses this secret to mask the output

* Third public version. Compared to the second version (21.11.2005), this version has better readability. The most important additions: the use of Elliptic Curve Method of factoring to achieve additional security, and the unified explanation of several protocols by using a forked compostion together with a communication-efficient CPIR, see Thm 2.

values, sent to Calista. The resulting two-message protocols [AIR01,BGN05] are relaxed-secure in the complexity-theoretic model. However, the Aiello-Ishai-Reingold $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol [AIR01] works only in conjunction with an IND-CPA secure homomorphic public-key cryptosystem PKC that has plaintext space of prime order n such as Elgamal [Elg85], while almost all known AH cryptosystems have a composite n with large prime factors. The Boneh-Goh-Nissim $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol [BGN05] uses a 2-DNF homomorphic PKC that works on groups of composite order, and allows the sender to compute-on-ciphertexts any quadratic functions of the plaintexts. Paradoxically, their CDS protocol is secure exactly because their PKC has inefficient decryption: also here, one has to compute a discrete logarithm to decrypt.

Our contributions. All previous CDS protocols require protocol's output space to be small. To overcome this, we construct an AH two-message CDS protocol. Due to the state of the art it has to work over plaintext groups of composite order. More precisely, the new CDS protocol can be used in conjunction with an IND-CPA secure PKC that satisfies substantially weaker *algebraic* properties than required by [AIR01,BGN05]; in particularly, PKC has to be (1) AH: this is weaker than the property required by [BGN05], and (2) the smallest prime factor $\mathsf{spf}(n)$ of n has to be sufficiently large: this is weaker than the property required by [AIR01]. Let us call a PKC that satisfies (2) *rough*. Additionally, we need that the correctness of the used public key is verified. This aspect will be thoroughly studied in Sect. 9. Briefly, we can either assume (a) the PKI model where the sender has a certified copy of receiver's public key, (b) that the receiver and the sender execute the key correctness proof once, and the same key is thereafter used in many protocols, or (c) the ECM factoring method is used online to verify that the public key is sufficiently rough. Thus, one can use standard model but that incurs either some extra communication or computation. We stress that most of the previous papers on AH homomorphic protocols either explicitly or implicitly assume (a) and/or (b), we are just formalizing the—given the current state of the art—inevitable.

Our construction consists of several steps. In a disclose-if-equal (DIE) protocol, the receiver obtains sender's private input β if his own private input α is equal to some fixed public constant b. We construct an AH two-message AH protocol for DIE that is crucially based on a randomness extractor for the distribution family $\mathcal{D} = \{m\mathbb{Z}_n : m \in \mathbb{Z}_n \wedge m \neq 0\}$, i.e., for the family of uniform distributions of all nonzero subgroups of $\mathbb{Z}_n$. Functionally, it is sufficient for the extractor to deterministically extract the lower order bits of its input. The conceptual difficulty is that the extractor needs to be computable-on-ciphertexts, i.e., to be an affine map modulo n, while bit extraction is not affine. Similarly, most of the known deterministic extractors for any non-trivial distribution families are not affine. Instead, we design an affine *seeded* extractor Ext for $\mathcal{D}$, that we also show—in the full version—to be close to optimal. This is the crucial step in our construction that takes care of the composite group order; the next steps work with any AH cryptosystem given the existence of such extractor.

To simplify the presentation of the next steps, in Sect. 3 we define a *forked composition* of two-message protocols. In a forked composition of several protocols with the same first message, the sender computes a database of the second messages of all composed protocols, and then the receiver uses a communication-efficient two-message CPIR protocol to recover the database elements he needs to obtain the protocol outcome. The rest of our protocols are all forked compositions of suitable DIE protocols and in particular can be constructed on top of the extractor Ext and of a communication-efficient two-message CPIR. Such a unifying framework is important because it lets us to derive some of the subsequent proofs automatically from the general security proof of forked compositions.

Then, using circuit evaluation and a suitable forked composition of DIE protocols, we show that every $\mathcal{S} \in \mathbf{NP/poly}$ has a $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol with polynomial resources. In particular, this protocol is often much more efficient than proving in non-interactive zero-knowledge that receiver's input belongs to $\mathcal{S}$. Finally, we propose the *CDS transformation* that transforms any private AH two-message protocol Π to a relaxed-secure AH two-message protocol. The CDS transformation is basically a forked composition of Π and a CDS protocol. The sender masks the output of Π with secrets, corresponding to all different receiver's inputs of Π. Thus, the receiver recovers any of the outputs only if all of her 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 CDS protocol and the number of outputs λ is "small". We construct a computationally sender-secure protocol that is communication-efficient for larger values of λ. All constructed protocols are relaxed-secure in the PKI model assuming only that the underlying IND-CPA AH cryptosystem is rough.

Applications. The CDS protocol has been largely overlooked in literature, with only a couple of published papers [GIKM00,AIR01,BGN05] more than mentioning it, and with many papers using zero-knowledge proofs where the CDS protocol can provide a simpler solution. We propose several interesting applications that demonstrate the power of the new tools. The CDS protocols from [AIR01,BGN05] can be applied in most of these settings but often with an exponentially smaller output space. First, we use the forked composition to construct a CPIR to OT transformation. Based on that, we propose an OT protocol with log-squared communication. (See Thm. 6.) Second, in Sect. 7, we construct a private millionaire's protocol with logarithmic communication. Third, we construct efficient private protocols for a few other tasks like conditional OT and multiplicative relationship (see Sect. 7) and scalar product (see Sect. 8). Finally, we show how to construct efficient *threshold* AH protocols for e-voting and e-auctions. (See Sect. 7.) All new protocols are round-optimal (in the PKI model), computationally receiver-private and statistically sender-secure solely under the assumption that the underlying IND-CPA secure AH cryptosystem PKC is rough.

Due to the space limitations, all proofs have been moved to the Appendix.

2 Preliminaries

For an integer n, let $[n] := \{1, 2, \ldots, n\}$ and let $\mathsf{spf}(n)$ be the smallest prime divisor of n. We say that n is p-*rough* if $\mathsf{spf}(n) \geq p$. The statistical difference of two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ over a discrete support Z is defined as $d(\mathcal{D}_1, \mathcal{D}_2) := \max_{S \subseteq Z} |\Pr[\mathcal{D}_1 \in S] - \Pr[\mathcal{D}_2 \in S]|$. $\mathcal{D}_1$ and $\mathcal{D}_2$ are ε-close, $\mathcal{D}_1 \overset{\varepsilon}{\sim} \mathcal{D}_2$, if $d(\mathcal{D}_1, \mathcal{D}_2) \leq \varepsilon$. For an arbitrary set Z, $U(Z)$ denotes the uniform distribution over it; we sometimes identify Z with $U(Z)$. A quasigroup $(Z, \circ)$ is a set with a binary operation $\circ : Z^2 \to Z$, such that 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 it is a constant, and that the adversary works in time that is less than some fixed public constant τ. App. A discusses adversaries that work in time that is polynomial in input size.

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 randomized 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. Denote $\mathsf{Enc}_{\mathsf{pk}}(m) := \mathsf{Enc}_{\mathsf{pk}}(m; U(\mathcal{R}))$. For an algorithm A, define $\mathsf{Adv}^{\text{IND-CPA}}_{\mathsf{PKC}}(A) := \left|\mathsf{Succ}^{\text{IND-CPA}}_{\mathsf{PKC},1}(A) - \mathsf{Succ}^{\text{IND-CPA}}_{\mathsf{PKC},0}(A)\right|$, where $\mathsf{Succ}^{\text{IND-CPA}}_{\mathsf{PKC},b}(A) := \Pr[(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}, (m_0, m_1) \leftarrow A(\mathsf{pk}) : A(\mathsf{pk}, m_0, m_1, \mathsf{Enc}_{\mathsf{pk}}(m_b)) = 1]$; the probability is taken over the coin tosses of Gen, Enc and A. 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 cryptosystem PKC is *homomorphic*, if for any key pair $(\mathsf{sk}, \mathsf{pk})$, any $x_1, x_2 \in \mathcal{M}$ and $r_1, r_2 \in \mathcal{R}$, $\mathsf{Enc}_{\mathsf{pk}}(x_1; r_1) \cdot \mathsf{Enc}_{\mathsf{pk}}(x_2; r_2) = \mathsf{Enc}_{\mathsf{pk}}(x_1 + x_2; r_1 \circ r_2)$, where $+$ is a group operation. We additionally assume that $\circ$ is a quasigroup operation—this is necessary for rerandomization—, that $\mathcal{M}$ and $\mathcal{R}$ are efficiently samplable, and that membership in $\mathcal{C}$ can be efficiently verified. PKC is *additively homomorphic* (AH) if $\mathcal{M} = \mathbb{Z}_n$ for some n, and *multiplicatively homomorphic*, if $\mathcal{M}$ is a multiplicative group where computing discrete logarithm is difficult (e.g., a prime-order subgroup of $\mathbb{Z}^*_n$, or an elliptic curve group). Many well-known homomorphic cryptosystems [Elg85,OU98,NS98,Pai99,DJ01,DJ03] are IND-CPA secure under some complexity assumptions. The Elgamal cryptosystem [Elg85] is MH (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 cryptosystems, with $\mathcal{M} = \mathbb{Z}_n$, $\mathcal{R} = \mathbb{Z}^*_n$ and $\mathcal{C} = \mathbb{Z}^*_{n^2}$ for an RSA modulus n. Thus, n is $\sqrt{n}/2$-rough.

3 AH Two-Message Protocols

Let α denote the private input of the receiver and β denote the private input of the sender. A (single output) two-message protocol between the receiver and the sender implements the following functionality for a public function f: an unbounded receiver learns $f(\alpha, \beta)$ and nothing more, and a computationally bounded sender learns no new information. The dual case of bounded receiver and unbounded sender, considered say in [CCKM00], is out of the scope for the current paper.

We state our results in the PKI model, where one assumes that a trusted key generator initially (TKG) runs Gen for an AH cryptosystem PKC, and then privately sends $(\mathsf{sk}, \mathsf{pk})$ to the receiver and pk to the sender. In particular, the sender knows that pk corresponds to this fixed receiver. This key pair is then possibly used in many different protocol runs. The PKI model is normal in applications like e-voting. Still, we stress that we use the PKI model only for the sake of simplicity of security proofs (and for some gain in efficiency). In Sect. 9, we investigate the conditions under which AH two-message protocols stay secure in the "standard model", where the receiver generates $(\mathsf{sk}, \mathsf{pk})$ and sends pk to the sender without the presence of any trusted third parties. Next, fix an AH cryptosystem PKC and a key pair $(\mathsf{sk}, \mathsf{pk})$.

During an AH two-message protocol Π, and on input α, the receiver computes $\mathfrak{q} \leftarrow (\mathsf{Enc}_{\mathsf{pk}}(\alpha_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(\alpha_\mu))$, where $\alpha = (\alpha_1, \ldots, \alpha_\mu)$, $\alpha_i \in \mathcal{M}$, for some $\mu \geq 1$. He sends $\mathfrak{q}$ to the sender. Note that one can efficiently verify, given only pk, that $\mathfrak{q}$ is a valid message because we assumed that membership test of $\mathcal{C}$ is efficient. After that, the sender replies with the second message $\mathfrak{a} = \mathfrak{a}(\beta, \mathfrak{q})$ by applying some randomized algorithm to the received ciphertexts and returning the resulting ciphertexts. We assume that $\mathfrak{a} = \perp$ if the sender does not have the public key, the sender halts or $\mathfrak{q}$ is malformed. Finally, the receiver obtains the answer by decrypting the received ciphertexts and then applying some local algorithm to the resulting plaintexts. The *communication* of this protocol is equal to $|\mathfrak{q}| + |\mathfrak{a}|$.

CPIR/OT protocols. During a 1-*out-of-*ν *CPIR* (CPIR_ℓ^ν) protocol for ℓ-bit strings, the receiver fetches β_α from the database $\beta = (\beta_1, \ldots, \beta_\nu)$ maintained by the sender, $\beta_i \in \{0,1\}^\ell$, so that a computationally bounded sender does not know which entry the receiver is learning. Clearly, the protocol where the sender just transfers the whole database to the receiver is a CPIR_ℓ^ν protocol. In the case of 1-*out-of-*ν *oblivious transfer*, OT_ℓ^ν, also sender's privacy is guaranteed. For a fixed CPIR/OT protocol Γ, let $C_{\Gamma,i}(\nu, \ell')$ denote the length of its ith message. A close-to-polylogarithmic CPIR_ℓ^ν protocol working over a non-length-flexible PKC, was proposed by Stern [Ste98]. Lipmaa's CPIR_ℓ^ν protocol [Lip05], based on a length-flexible AH cryptosystem [DJ01], has polylogarithmic receiver-computation, linear sender-computation, and communication $(\log_2^2 \nu + (s + \frac{3}{2}) \cdot \log_2 \nu + 1)k = \Theta(k \cdot \log^2 \nu + \ell \cdot \log \nu)$, where $k = \Omega(\log n)$ is the security parameter and $s := \lceil \ell / k \rceil$. A CPIR_ℓ^ν protocol with communication $\Theta(\log \nu + \ell + k)$ but with superpolylogarithmic receiver-computation was recently proposed by Gentry and Ramzan [GR05b].

Threshold AH two-message protocols. In threshold AH two-message protocols, the secret key is owned by Calista, most usually a coalition of servers (here, this coalition could be a coalition of senders). The receiver encrypts her inputs by using Calista's public key, forwards ciphertexts to the sender, who applies some operations on them, and forwards the resulting ciphertexts to Calista who then threshold-decrypts them. This setting is common in the e-voting and e-auction protocols [CGS97,DJ01,LAN02]. In this case, our methods provide full security [Gol04] of Calista against a malicious receiver and a semihonest sender. The security definitions carry over.

Relaxed-security of AH two-message protocols. We use "standard" relaxed security definitions (see, e.g., [AIR01,FIPR05]) where one cares about the correctness, receiver-privacy and sender-security. Since the sender obtains no output, this is equal to the full security against semihonest sender and malicious receiver. Briefly, (1) Π is *correct* if in the case of the honest receiver and honest sender, the receiver always recovers $f(\alpha, \beta)$; (2) Π is (ε, τ)-receiver-private, if after seeing the protocol transcript no τ-time adversary can distinguish between any pair of possible receiver's inputs with advantage larger than ε (cf. the earlier definition of IND-CPA security), and (3) sender-security is defined in comparison with the ideal model where there exists a trusted party that gets the inputs f, α and β, and returns $f(\alpha, \beta)$ to the receiver. We require in the real implementation that the receiver does not get any information beyond the value of $f(\alpha, \beta)$. Due to the structure of AH two-message protocols, to prove sender's privacy, it suffices to define a universal non-rewinding probabilistic polynomial-time simulator that first generates a key pair $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}$, sends $(\mathsf{sk}, \mathsf{pk})$ to the receiver and pk to the sender, decrypts receiver's first message, sends the resulting plaintext tuple α^* to the TTP, obtains TTP's output $f(\alpha^*, \beta)$, and then outputs a string $view^*$, such that $(\mathsf{sk}, \mathsf{pk}, view^*)$ is ε-close to the joint distribution of the key pair and of receiver's view of the real protocol. If such a *canonical* simulator exists, then we say that the protocol is ε*-sender-secure*. We say that Π is $(\varepsilon, \tau; \sigma)$*-relaxed-secure* if it is correct, (ε, τ)-receiver-private and σ-sender-secure.

General security theorem. For a fixed key pair $(\mathsf{sk}, \mathsf{pk})$ and a fixed protocol Π with honest participants, denote the distribution of queries $\mathfrak{q}$ by $\mathcal{Q}(\alpha)$ and the distribution of answers $\mathfrak{a}$ by $\mathcal{A}(\alpha, \beta)$. We say that Π is σ*-simulatable* if there exists an efficient algorithm Σ such that $\Sigma(\mathsf{pk}, f(\alpha, \beta)) \overset{\sigma}{\sim} \mathcal{A}(\alpha, \beta)$ for any α, β. Intuitively, correctness

means that the receiver can recover the value f from any correctly formed second message while simulatability means that correctly distributed second message can be generated from the value of f. It is straightforward to prove the next general theorem.

Theorem 1. *Let Π be an AH two-message protocol in the PKI model that uses an (ε,τ)-IND-CPA-secure AH cryptosystem* PKC. *If Π is correct and σ-simulatable then Π is $(\mu\varepsilon, \tau - O(\mu); \sigma)$-relaxed-secure.*

Thus, computational receiver-privacy of an AH two-message protocol in the malicious model follows from the IND-CPA security of PKC. Analogously, many threshold AH two-message protocols are secure only in the semihonest model. To achieve sender-security in the malicious model, one usually designs a protocol that is sender-secure in the semihonest model and then applies zero-knowledge proofs to assure that the receiver behaves honestly, i.e., encrypts "valid" inputs. However, this either increases the number of messages or requires a security model with a common reference string (CRS) or random oracles.

Forked composition. Several of the later defined protocols are what we will now define to be forked compositions of simpler protocols. We also state a general security theorem for forked composition from which several of the later theorems follow straightforwardly.

Fix PKC and a key pair. Clearly, if the honest receiver has the same input α in two protocols Π_1 and Π_2, then the distribution $\mathcal{Q}(\alpha)$ of the first message in the both protocols is the same (it's just a tuple of encryptions). We define a forked composition of m protocols Π_i, $i \in [m]$, with respective receiver-sender input pairs (α,β_1), ... , (α,β_m), as follows. In $(\bigotimes_{i=1}^m \Pi_i)(\alpha,\beta_1,\ldots,\beta_m)$, the receiver sends $\mathfrak{q} \leftarrow \mathcal{Q}(\alpha)$ as the first message. The sender answers with an m-tuple of the second messages $\mathfrak{a}_i$ from all m protocols Π_i. The next lemma is straightforward.

Lemma 1. *Let m AH two-message protocols Π_i for functionalities f_i be respectively $(\varepsilon,\tau;\sigma_i)$-relaxed-secure in the PKI model. Then $(\bigotimes_{i=1}^m \Pi_i)(\alpha,\beta_1,\ldots,\beta_m)$ is $(\varepsilon,\tau-O(\mu);\sum_{i=1}^m \sigma_i)$-relaxed-secure in the PKI model.*

Since m can be relatively large and in many cases—as we will show abundantly in the rest of the paper—the receiver does not need the whole database $\mathfrak{a} = (\mathfrak{a}_1,\ldots,\mathfrak{a}_m)$ to recover his private output, we can use an AH two-message CPIR protocol Γ to transfer only $m^* \ll m$ necessary values $\mathfrak{a}_i$. In practice, it sometimes suffices to have $m^* = 1$. Applying Lem. 1 again on forking-composition of $(\bigotimes_{i=1}^m \Pi_i)$ and Γ, denoted as $(\bigotimes_{i=1}^m \Pi_i)_\Gamma(\alpha,\beta_1,\ldots,\beta_m)$, we get the following theorem as a direct conclusion.

Theorem 2. *Let m AH two-message protocols Π_i for functionalities f_i be respectively $(\varepsilon,\tau;\sigma_i)$-relaxed-secure in the PKI model. Let Γ be a (ε_c,τ)-receiver-private AH two-message* CPIR *protocol that allows to retrieve necessary replies $\mathfrak{a}_i$. Then $(\bigotimes_{i=1}^m \Pi_i)_\Gamma(\alpha,\beta_1,\ldots,\beta_m)$ is $(\varepsilon+\varepsilon_c,\tau-O(t_\mathfrak{q});\sum_{i=1}^m \sigma_i)$-relaxed-secure in the PKI model, where $t_\mathfrak{q}$ is time needed to compute receiver's first message.*

More precisely, assume that in the worst case, $m^* \ll m$ second messages $\mathfrak{a}_i$ suffice for the receiver to recover his private output. In parallel with $\bigotimes \Pi_i$, the receiver and the sender execute Γ to receive m^* database elements where sender's database consists of the m messages $\mathfrak{a}_i$, $i \in [m]$, that the senders of Π_i would have sent on corresponding inputs $(\mathfrak{q},\beta_i)$. That is, the receiver sends out $\mathfrak{q}$ and the first message of CPIR, and the send replies with the second message of CPIR applies to $(\mathfrak{a}_1,\ldots,\mathfrak{a}_m)$.

Note that if α does not correspond to a valid input of the receiver in Π_i then $\mathfrak{a}_i$ gives no useful input to the receiver. As always, we can use the trivial CPIR where the sender just sends all database elements to the receiver. A similar result holds also for non-AH CPIR protocols like [GR05b]. However, for AH CPIR protocols that are based on the same PKC as Π_i, the only assumption is that PKC is IND-CPA-secure while for non-AH CPIR protocols, we have to make a separate assumption that $\mathcal{C}$ is receiver-private.

4 New Seeded Randomness Extractor

Let $\mathcal{D}$ be a family of distributions on some set X_1. A map $\mathsf{Ext} : X_1 \times S \to X_2$ is a *seeded ε-extractor* for $\mathcal{D}$ if for every distribution $\mathcal{D}$ in $\mathcal{D}$, $\mathsf{Ext}(\mathcal{D}, U(S)) \overset{\varepsilon}{\sim} U(X_2)$. (See [GRS04,GR05a,KM05b,CFGP05] for related references.) We need a seeded randomness extractor $\mathsf{Ext} : \mathbb{Z}_n \times S \to \mathbb{Z}_n$ where n is a large composite integer and $\mathcal{D} := \{U(x\mathbb{Z}_n) : x \in \mathbb{Z}_n \setminus \{0\}\}$. Since we compute Ext on ciphertexts, it has to be affine, i.e., $\mathsf{Ext}(m,(s_0,s_1)) = s_0 m + s_1 \mod n$ for some set $S = \{(s_0,s_1)\} \subseteq \mathbb{Z}_n^2$, chosen so that Ext is an ε-extractor for $\mathcal{D}$ for as small ε as possible. This requirement makes common extractors unusable.

Theorem 3. *Let $n > 0$ be an odd integer, $0 < \varepsilon \leq 1$, and let $\ell := \lfloor \log_2 \mathsf{spf}(n) - \log_2(1/\varepsilon) + 1 \rfloor$. Denote $T := \lfloor 2^{-\ell} n \rfloor$ and $S := 2^\ell \cdot \mathbb{Z}_T = \{2^\ell t : 0 \leq t \leq T-1\}$. Let $\mathsf{Ext}(m,s) := m + s \mod n$ for $m \in \mathbb{Z}_n$ and $s \in S$. Then Ext is an ε-extractor for $\mathcal{D}$, i.e., $xU(\mathbb{Z}_n) + U(2^\ell \cdot \mathbb{Z}_T) \stackrel{\varepsilon}{\sim} U(\mathbb{Z}_n)$ for any $x \not\equiv 0 \pmod n$.*

5 AH Two-Message Protocol for Disclose-If-Equal

A *disclose-if-equal* (DIE) protocol DIE_ℓ^b for ℓ-bit strings fulfills the next functionality: the receiver has a private input $\alpha \in \{0,1\}^\ell$, and the sender has a private input $\beta \in \{0,1\}^\ell$. The common input $b \in \{0,1\}^\ell$ is public. The receiver obtains sender's private input β exactly if $\alpha = b$. Like always in the case of AH two-message protocols, receiver obtains some element from some almost β-independent distribution otherwise.

Following [AIR01], one can define the next AH two-message DIE protocol. Let PKC be an IND-CPA secure AH cryptosystem with $\mathcal{M} = \mathbb{Z}_n$. The receiver sends to the sender a random encryption of α. The sender replies with a random encryption $\mathfrak{a}$ of $(\alpha - b)U(\mathcal{M}) + \beta$. The receiver obtains $\beta^* \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathfrak{a})$. Clearly if $\alpha = b$ then $\beta^* = \beta$. This protocol is relaxed-secure if n is prime. However, for a composite n, consider the input $\alpha \leftarrow n/\mathsf{spf}(n) + b$. Then $(\alpha - b)U(\mathcal{M})$ is a random member of a subgroup of $\mathbb{Z}_n$ of order $\mathsf{spf}(n)$ and thus does not completely hide β.

New DIE protocol. As emphasized before, there are no known IND-CPA secure AH cryptosystems with prime n. Our goal is to modify the above DIE protocol so that it would work together with an AH public-key cryptosystem with composite n. For this, define $T := \lfloor 2^{-\ell} n \rfloor$ and

$$\mathsf{DIE2}_\ell^b(\mathfrak{q}, \beta) := (\mathfrak{q}/\mathsf{Enc}_{\mathsf{pk}}(b))^{U(\mathcal{M})} \cdot \mathsf{Enc}_{\mathsf{pk}}(U(2^\ell \cdot \mathbb{Z}_T) + \beta) \ ,$$

i.e., $\mathsf{DIE2}_\ell^b$ is a random encryption of $(\mathsf{Dec}_{\mathsf{sk}}(\mathfrak{q}) - b)U(\mathcal{M}) + U(2^\ell \cdot \mathbb{Z}_T) + \beta$. The next DIE protocol, depicted by Prot. 1, is an extension of the Aiello-Ishai-Reingold OT protocol to the case where the order of the underlying group is composite but still sufficiently rough.

Query phase: The receiver sends $\mathfrak{q} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\alpha)$, where $\alpha \in \{0,1\}^\ell$, to the sender.
Transfer phase: If $\mathfrak{q} \notin \mathcal{C}$ then the sender returns $\perp$. Otherwise, the sender returns $\mathfrak{a} \leftarrow \mathsf{DIE2}_\ell^b(\mathfrak{q}, \beta)$, where $\beta \in \{0,1\}^\ell$.
Postprocessing: The receiver returns $\mathsf{Dec}_{\mathsf{sk}}(\mathfrak{a}) \mod 2^\ell$.

Protocol 1: Protocol DIE_ℓ^b

Theorem 4. *Let PKC be an (ε,τ)-IND-CPA secure AH cryptosystem, such that $n := \sharp\mathcal{M}$ is $(2^{\ell+1}/\sigma)$-rough for some $0 < \sigma \leq 1$ with $\ell \leftarrow \lfloor \log_2 \mathsf{spf}(n) - \log_2(1/\sigma) + 1 \rfloor$. Let $T := \lfloor 2^{-\ell} n \rfloor$. Then Prot. 1 is an $(\varepsilon, \tau - O(1); \sigma)$-relaxed-secure AH two-message DIE_ℓ^b protocol in the PKI model.*

Choice of ℓ. In practice, we can assume that $\sigma = 2^{-80}$, then a 2^{80}-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 $\ell \leftarrow \lfloor \frac{1}{2} \cdot \log_2 n - \log_2(1/\sigma) \rfloor$. For $\log_2 n = 1024$ and $\sigma = 2^{-80}$, we get $\ell = 433$.

6 AH Two-Message Protocol for Conditional Disclosure of Secrets

A conditional disclosure of secrets protocol $\mathsf{CDS}_\ell^{\mathcal{S}}$ [AIR01] for a public set $\mathcal{S}$ and ℓ-bit inputs is a (two-party) two-message protocol for the next functionality: $f(\alpha,\beta) = \beta$ if $\alpha \in \mathcal{S}$ and $f(\alpha,\beta)$ is a random element of $\{0,1\}^\ell$ otherwise. Next, we construct, given any public $\mathcal{S} \subseteq \mathbb{Z}_{2^\ell}$, an AH two-message protocol for $\mathsf{CDS}_\ell^{\mathcal{S}}$.

We first define the *conjunctive affine equality test protocol* $\mathsf{CAET}_\ell^{a,b}$. For a $\mu \times \upsilon$ matrix $A = (a_{ij})$ and a υ-element vector b with a_{ij}, $b_j \in \{0,1\}^\ell$, on receiver's input $(\alpha_1, \ldots, \alpha_\mu) \in \{0,1\}^{\mu\ell}$ and sender's input

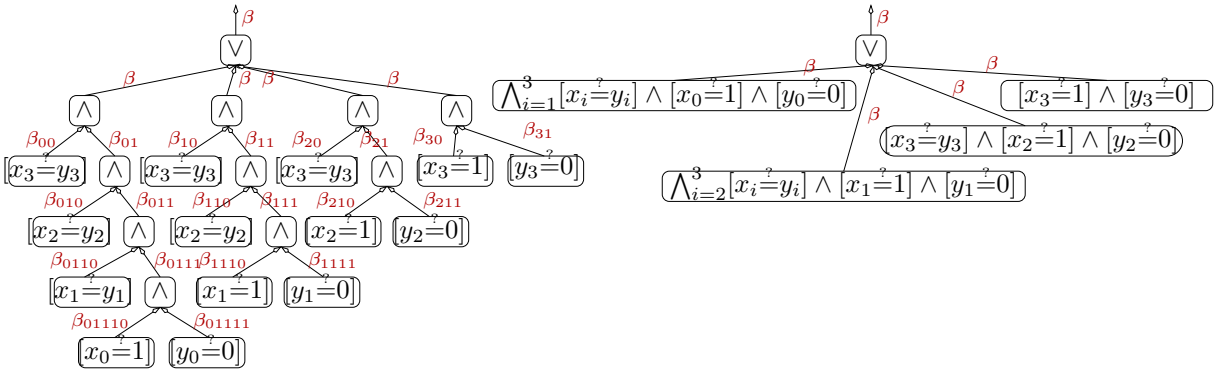

Fig. 1. An unoptimized and an optimized circuit for $\mathrm{GT}_4(y)$. The circuit on the left does not use *conjunctive* affine equality tests

$\beta \in \{0,1\}^\ell$, the receivers obtain β in $\mathsf{CAET}_\ell^{a,b}$ if $\bigwedge_{j=1}^{v}[\sum_{i=1}^{\mu} a_{ij}\alpha_i \stackrel{?}{=} b_j]$. Otherwise, the receiver obtains no information. In our $\mathsf{CAET}_\ell^{a,b}$ protocol, an honest receiver sends $\mathfrak{q} := (\mathfrak{q}_1, \ldots, \mathfrak{q}_\mu)$ to the sender, where $\mathfrak{q}_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\alpha_i)$. An honest sender replies with an $\mathfrak{a} = \mathsf{CAET2}_\ell^{a,b}(\mathfrak{q}, \beta)$ that is a random encryption of $\sum_{j=1}^{v}(\sum_{i=1}^{\mu} a_{ij} \cdot \mathsf{Dec}_{\mathsf{sk}}(\mathfrak{q}_i) - b_j)U(\mathcal{M}) + U(2^\ell \cdot \mathbb{Z}_T) + \beta$, where again $T := \lfloor 2^{-\ell} n \rfloor$. Receiver outputs $\mathsf{Dec}_{\mathsf{sk}}(\mathfrak{a}) \mod 2^\ell$ as in DIE_ℓ^b. Clearly, Thm. 4 also holds for this modified protocol.

Now, we are ready to define the CDS protocol. We specify the predicate $[\alpha \in \mathcal{S}]$ by a set of constraints on receiver's input α, where for the sake of efficiency, α might be broken down to several smaller inputs—usually, its bits—so that from them, the sender can recover the original encrypted inputs by using homomorphic operations. We write the constraints down as a suitable formula $\Psi_{\mathcal{S}}$ with conjunctive affine equality tests, where we allow threshold operations $\mathrm{THRESHOLD}_i$ (with $\mathrm{THRESHOLD}_i(x_1, \ldots, x_s) = 0$ iff at least i values x_j are equal to 0, and $\mathrm{THRESHOLD}_i(x_1, \ldots, x_s) = 1$ otherwise), Boolean operations $\wedge$ and $\vee$ and conjunctive affine equality tests. Here, the $\vee$ gates may have an arbitrary fan-in. It is required that $\Psi_{\mathcal{S}}(x) = 0$ iff $x \notin \mathcal{S}$. E.g., assume that $\mathcal{S}$ 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 $x = (x_{\mu-1}, \ldots, x_0)$,

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

Here, every row corresponds to one conjunctive affine equality test. Circuit evaluation is done as follows (Fig. 1): Construct a circuit where every leaf ψ implements acorresponding conjunctive affine equality test and every internal node implements a gate from the set $\{\mathrm{THRESHOLD}, \vee, \wedge\}$. Let $\mathsf{size}(\Psi_{\mathcal{S}})$ be the size of this circuit. Next, enumerate all nodes starting with leafs $\psi \in [\nu]$, where $\nu = \mathcal{L}(\Psi_{\mathcal{S}})$ is the number of leafs (conjunctive affine equality tests) in this circuit, and ending with internal nodes $\psi \in \{\nu+1, \ldots, \mathsf{size}(\Psi_{\mathcal{S}})\}$. Process the circuit recursively from top to bottom, similarly to the approach of [BL88]. Assign a secret $\beta_\psi \leftarrow U(\{0,1\}^\ell)$ to the unique topmost gate ψ of the circuit. For every $\wedge$ gate ψ with children ψ_1, ψ_2 and a secret β_ψ assigned to it, pick $\beta_{\psi_1} \leftarrow U(\{0,1\}^\ell)$ and $\beta_{\psi_2} \leftarrow \beta_\psi - \beta_{\psi_1} \mod 2^\ell$, and assign β_{ψ_i} to ψ_i. For every $\vee$ gate, just push the output secret downwards. For a $\mathrm{THRESHOLD}_k$ gate, generate a random k-degree polynomial f_ψ with $f_\psi(0) = \beta_\psi$ and assign the secret $f_\psi(i)$ to its ith child.

The CDS protocol $\mathsf{CDS}_\ell^{\mathcal{S}}$ is a forked composition of conjunctive affine test protocols CAET, $(\bigotimes_{\psi=1}^{\nu} \mathsf{CAET}_\ell^{(a_\psi, b_\psi)})_{\Gamma, m^*}((\alpha_1, \ldots, \alpha_\mu), \beta'_1, \ldots, \beta'_\nu)$ where the value of $m^* \leq \nu$ depends on the concrete protocol and (a_ψ, b_ψ) specifies the conjunctive affine equality test at leaf ψ. That is, the receiver sets $\mathfrak{q}_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\alpha_i)$

for $i \in [\mu]$. The sender prepares a database $(\mathfrak{a}_1, \ldots, \mathfrak{a}_\nu)$, where $\mathfrak{a}_\psi \leftarrow \mathsf{CAET}^{(a^\psi, b^\psi)}(\alpha, \beta_\psi)$. In parallel, the receiver uses Γ to obtain $m^* \leq \nu$ values $\mathfrak{a}_\psi$ from the correct branch of the circuit that he needs to recover the output. After that, by inversely following circuit generation, the receiver decrypts ciphertexts that correspond to the correct branch in the circuit and recovers β (modulo 2^ℓ). In particular, in the case of a threshold gate, she uses the Lagrange interpolation formula to recover β_ψ (modulo 2^ℓ). Unless all the affine equality tests of ψ are satisfied, the receiver learns nothing from c_ψ; in particular, the receiver does not learn which equality tests fail. We call the resulting protocol $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$.

Theorem 5. *Let* PKC *be an* (ε, τ)*-IND-CPA secure AH cryptosystem, such that* $n := \sharp\mathcal{M}$ *is* $(2^{\ell+1}\nu/\sigma)$*-rough for some* $0 < \sigma \leq 1$ *with* $\ell \leftarrow \lfloor \log_2 \mathsf{spf}(n) - \log_2 \nu - \log_2(1/\sigma) + 1 \rfloor$*. For* $\ell' := \lfloor \log_2 \sharp\mathcal{C} \rfloor$*, let* Γ *be an* (ε_c, τ)*-receiver-private AH two-message* $\mathsf{CPIR}_{\ell'}^{\nu}$ *protocol in the PKI model. Let* m^* *be a protocol-specific value defined as above. The* $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ *protocol is* $(\mu \cdot \varepsilon + \varepsilon_c, \tau - O(t_{\mathsf{q}}); \sigma)$*-relaxed-secure in the PKI model, where* t_{q} *is the time that the honest receiver takes to compute the first message of the protocol.*

Clearly, $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ has receiver-communication of $\mu + m^* \cdot C_{\Gamma,1}(\mathcal{L}(\Psi_{\mathcal{S}}), \log_2 \sharp\mathcal{C})$ ciphertexts and sender-communication of $C_{\Gamma,2}(\mathcal{L}(\Psi_{\mathcal{S}}), \log_2 \sharp\mathcal{C})$ ciphertexts. The use of conjunctive affine equality tests helps one often decrease the communication: as seen from Fig. 1, formulas without conjunctive tests can be much larger. Without counting the time to need to execute CPIR, receiver's worst-case computation is $\Theta(\mu + \mathsf{size}(\Psi_{\mathcal{S}}))$ group operations and sender's worst-case computation is $O(\mu \cdot \mathcal{L}(\Psi_{\mathcal{S}}))$ group operations. Going back to the motivating example, $\mathcal{L}(\Psi_{\mathrm{GT}_\mu(y)}) = \mu$. (See Fig. 1, right. Here, the sender conditionally transfers the same secret μ times.)

Thus, under the same assumptions as in Thm. 5 and with using a trivial CPIR, there exists an $(\mu \cdot \varepsilon, \tau - O(\mu); \sigma)$-relaxed-secure AH two-message protocol for $\mathsf{CDS}_\ell^{\mathrm{GT}_\mu(y)}$, $\ell = \mu$, with the communication of 2μ ciphertexts, logarithmic in $\sharp\mathcal{S}$ receiver's computation and log-squared in $\sharp\mathcal{S}$ sender's computation. Together with a CPIR, this protocol is $(\mu \cdot \varepsilon, \tau - O(\mu + C_{\Gamma,1}(\mu, \log_2 \sharp\mathcal{C})); \sigma)$-relaxed-secure, have the communication of $\mu + C_{\Gamma,1}(\mu, \log_2 \sharp\mathcal{C}) + C_{\Gamma,2}(\mu, \log_2 \sharp\mathcal{C})$ ciphertexts—e.g., $\mu + \Theta(\log \mu)$ ciphertexts if the CPIR by Gentry and Ramzan is used—, and with computation increased by the CPIR-ing cost. In this case, $m^* = 1$.

In general, if the size of $\Psi_{\mathcal{S}}$ is polynomial in $\log_2 \sharp\mathcal{S}$ then $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ has computation and communication that is polynomial in $\log_2 \sharp\mathcal{S}$. Moreover, we can include non-monotonous circuits by first pushing all negations down the circuit (using De Morgan laws) to the leaves, and then representing the inputs in a way that gets rid of the negations of affine equality tests. (E.g., $[4x_2 + 2x_1 + x_0 \neq 7]$ is equivalent to $[x_2 = 0] \vee [x_1 = 0] \vee [x_0 = 0]$.) Note that THRESHOLD gates usually do not decrease communication but may slightly decrease computation. Hence, all languages $\mathcal{S}$ in $\mathbf{NP/poly}$—$\mathbf{NP}$, since the known witness can be entered as a part of the input—have a family of $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocols with polynomial communication and computation. Since affine maps can be presented by using polynomial circuits then the $\mathsf{CircuitCDS}_\ell^{\mathcal{S}}$ protocol has polynomial resources iff $\mathcal{S} \in \mathbf{NP/poly}$. This can be compared to the fact that it is only known how to compute-on-ciphertexts maps from $\mathbf{NC^1}$ [SYY99]. For most of the interesting protocols, the communication of the receiver is $\Theta(\log \ell)$ ciphertexts, and the communication of the sender can be reduced to $\Theta(\log \ell)$ by using the Gentry-Ramzan CPIR.

7 Applications

AH Two-Message Protocol for Oblivious Transfer. Aiello, Ishai and Reingold [AIR01] defined an elegant relaxed-secure two-message OT_ℓ^ν protocol. However, as mentioned in the full version of [AIR01], the protocols of [AIR01] only work efficiently if all encrypted values are elements of the underlying group. Moreover, their protocol does not work if the plaintext group has a composite order $n = \prod p_i^{a_i}$, for different primes $p_1 < p_2 < \cdots < p_t$. Really, if receiver's input α^* is such that $\alpha^* \equiv \alpha_i \mod p_i$ for some mutually different values $\alpha_i \in [\nu]$, then the receiver can straightforwardly compute the values $\beta_{\alpha_i} \mod p_i$, $i \in [t]$. The receiver 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] 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 also proposed a 1-out-of-ν-OT protocol; however, there a honest receiver has to encrypt values that depend on the secret key which makes it unusable in some of the applications.

Assume that Γ is an arbitrary AH two-message $\mathsf{CPIR}^{\nu}_{\ell'}$ protocol with $\ell' > \log_2 \sharp\mathcal{C}$. Clearly, the ν-times forked composition $(\bigotimes_{i=1}^{\nu} \mathsf{DIE}^i_\ell)(\alpha, (\beta_1, \ldots, \beta_\nu))_{\Gamma,1}$ is functionally equal to an oblivious transfer protocol: if $\alpha = i$ for some $i \in [\nu]$ then the receiver obtains the secret (database element) β_ν and otherwise the receiver obtains no new information. This construction, otherwise equivalent to the construction of [AIR01] but working over an AH cryptosystems, presents yet another CPIR to OT transformation. The next theorem is a straightforward corollary of Thm. 2/4:

Theorem 6. *Let* PKC*,* Γ *and* ℓ' *satisfy the requirements of Thm. 5. Let* $T := \lfloor 2^{-\ell} n \rfloor$*. Then* $(\bigotimes_{i=1}^{\nu} \mathsf{DIE}^i_\ell)_{\Gamma,1}(\alpha, (\beta_1, \ldots, \beta_\nu))$ *is an* $(\varepsilon + \varepsilon_c, \tau - \mathrm{polylog}(\nu); \sigma)$*-relaxed-secure AH two-message* OT^{ν}_{ℓ} *protocol in the PKI model.*

Note that in the $\mathsf{CPIR}^{\nu}_{\ell'}$ protocol from [Lip05] has communication $\Theta(k \cdot \log^2 \nu + \ell \cdot \log \nu)$, where k is a possibly non-constant security parameter. Lipmaa [Lip05] constructed an OT protocol in the PKI model with log-squared communication, assuming both that PKC is IND-CPA secure and the Decisional Diffie-Hellman problem is hard. Thus, Thm. 6 achieves the same result assuming only that PKC is IND-CPA secure. For a non-length-flexible PKC, Thm. 6 and [Ste98] result in an OT^{ν}_{ℓ} protocol with sublinear-but-superpolylogarithmic communication. Due to [GR05b], there exists a non-AH two-message OT^{ν}_{ℓ} protocol with communication $\Theta(\log \nu + \ell + k)$, assuming both that PKC is IND-CPA secure and that Φ Hiding is hard. Note that two-message oblivious transfer protocol in the standard model that works over message spaces of composite order n was proposed by Kalai [Kal05]. It is relaxed-secure even if n is maliciously chosen, without any need for key correctness proofs. However, in Kalai's protocol, unknown parts of receiver's message are encryptions of completely random values.

Choice of ℓ. If $\sigma = 2^{-80}$ and $\nu \leq 2^{40}$ 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 $\ell \leftarrow \left\lfloor \frac{1}{2} \cdot \log_2 n - \log_2 \nu - \log_2(1/\sigma) \right\rfloor$. For $\log_2 n = 1024$ and $\sigma = 2^{-80}$, we get $\ell = \lfloor 433 - \log_2 \nu \rfloor \geq 393$. The protocol can be modified to transfer $\ell' > \ell$ bits by repeating its second message $\lceil \ell'/\ell \rceil$ times.

Multiplicative relationships and polynomial arithmetic. A paper by Kissner and Song on privacy-preserving set operations [KS05] but also several previous papers [FNP04,KM05a] 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 subproblem: given $\mathsf{Enc}_{\mathsf{pk}}(x)$, $\mathsf{Enc}_{\mathsf{pk}}(y)$ and $\mathsf{Enc}_{\mathsf{pk}}(z)$ for $x, y \in \{0,1\}^{\ell/2}$ and $z \in \{0,1\}^{\ell}$, encrypted by the receiver, the receiver must obtain the correct answer only if $z = xy$. Now, by the long multiplication rule, $z = xy$ iff $z = \sum_{i=0}^{\ell/2-1} xy_i 2^i$. Therefore, $\Psi_{[z=xy]}$ is a conjunction of the next tests, where w_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) $[(y_i = 0 \wedge w_i = 0) \vee (y_i = 1 \wedge w_i = x)]$ for $i \in \mathbb{Z}_{\ell/2}$, and (4) $[z = \sum_{i=1}^{\ell/2-1} w_i \cdot 2^i]$, 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, receiver's communication is 2.5ℓ ciphertexts and sender's communication is $\mathcal{L}(\Psi_{[z=xy]}) \leq 4\ell + 1$ ciphertexts. Thus, the total communication is $6.5\ell + 1$ ciphertexts (using a CPIR could decrease it even more). Now, we can also construct a CDS protocol for the set $\mathcal{S} = \{(fg, f, g)\}$, 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, it holds that $X \cup Y = Z$.

Millionaire's protocol with logarithmic communication. The millionaire's problem is, given receiver's private input α and sender's private input β from $\{0,1\}^{\ell}$, decide whether $\alpha > \beta$. Though there have been proposed numerous protocols for this problem (see, for example, [Fis01,BK04,ST04]), none of the proposals is completely satisfactory. E.g., the AH two-message protocol of Blake and Kolesnikov [BK04] is sender-secure only in the semihonest model, while a different protocol by [ST04] uses zero-knowledge proofs to achieve sender-security in the malicious model.

The next generic modification of $\mathsf{CircuitCDS}^{\mathcal{S}}_{\ell}$ can be used in the case of some private sets $\mathcal{S}$ that depend on β. Assume that the receiver has a private input α and that Server has a private input $\beta \in \{0,1\}^{\ell}$ and that the $\mathsf{CircuitCDS}^{\mathcal{S}}_{\ell}$ protocol is written down in a disjunctive normal form over affine equality 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 $\mathsf{CircuitCDS}^{\mathcal{S}}_{\ell}$ protocol as follows. Assume Γ is the trivial CPIR, this is necessary since $\mathcal{S}$ is not public. Fix a *public* value t (for example, $t = 0$) and push it down the circuit.

For every leaf gate ψ, let the sender to compute $\mathfrak{a}_\psi$ as previously, but return the values $\mathfrak{a}_\psi$ in a random order. This must be done so that the number of accepting leaf gates is always either 0 if $\Psi_{\mathcal{S}} = 0$, or some non-zero constant if $\Psi_{\mathcal{S}} = 1$. Thus, by testing that at least one of the ciphertexts $\mathfrak{a}_\psi$ encrypts 0, the receiver gets to know whether $\Psi_{\mathcal{S}}(\alpha, \beta)$ is true or not. Since $\mathcal{L}(\Psi_{\mathrm{GT}_\mu(y)}) = \mu$, 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, receiver's computation $\Theta(\mu)$ and sender's computation $\Theta(\mu^2)$, assuming only that the underlying AH public-key cryptosystem is IND-CPA secure. Compared to the Blake-Kolesnikov protocol [BK04] which is only secure in the semihonest model, this means quadratically more sender-computation but otherwise the new protocol is roughly as efficient.

Conditional OT. In a *conditional oblivious transfer (*$\mathsf{COT}_\ell^{\mathcal{S}}$*)* protocol [DOR99], the receiver has a private input α and the sender has a private input (β_1, β_2). The receiver obtains β_2 exactly if $\Psi_{\mathcal{S}}(\alpha, \beta_1) = 1$ for some public set $\mathcal{S}$ of valid receiver's and sender's input pairs. In the case of a $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol, β_1 is an empty string and therefore a $\mathsf{COT}_\ell^{\mathcal{S}\times\emptyset}$ protocol can be used to implement a $\mathsf{CDS}_\ell^{\mathcal{S}}$ protocol. Assume now that $\mathcal{S} \in \mathbf{NP/poly}$ is such that for some circuit representation $\Psi_{\mathcal{S}}$, the number of accepting leaf gates is always either 0 if $\Psi_{\mathcal{S}} = 0$, or some non-zero constant if $\Psi_{\mathcal{S}} = 1$. To implement $\mathsf{COT}_\ell^{\mathcal{S}}$ for $\mathcal{S}$, 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 $L = 80$.

Electronic voting and auctions without random oracles. In the case of threshold AH two-message protocols, conditional disclosure of secrets can be used to guarantee *full security against a malicious receiver*. As in [BGN05], consider an electronic voting protocol where every voter sends an AH encryption $c_i \leftarrow \mathsf{Enc}_{\mathsf{pk}}(v_i)$ to talliers. We assume that the protocol is secure 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. The same trick can be used to eliminate the need for random oracles in the AH electronic auction scheme of [LAN02] and in many other similar protocols. Compared to the protocols of [BGN05], our protocols are more efficient in the case of multi-candidate elections ([BGN05] allows to efficiently decrypt only if the plaintext is small) and is based on an incomparable but a somewhat more standard security assumption.

8 CDS Transformation

Next, we present a generic transformation from private (i.e., secure in the semihonest model) AH two-message protocols to relaxed-secure AH two-message protocols. It can also be used as a subprotocol in many-message protocols. Fix an AH two-message protocol Π. Denote receiver's input by $\alpha = (\alpha_1, \ldots, \alpha_\mu)$, sender's input by $\beta = (\beta_1, \ldots, \beta_\nu)$, and receiver's output by $\delta = (\delta_1, \ldots, \delta_\lambda)$. 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. 5. Larger inputs and outputs can be handled straightforwardly. The query phase consists of sending the elements $\mathsf{Enc}_{\mathsf{pk}}(\alpha_i)$ and the transfer phase consists of sending the elements $\mathsf{Enc}_{\mathsf{pk}}(\delta_j)$. We assume that α, β and δ have already been modified to allow efficient circuit evaluation. E.g., in the case of $\mathrm{GT}_\mu(y)$, every α_i is a bit. Assume that the input α of an honest receiver belongs to some public 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 protocol [Cha04] being one of the few exceptions. We assume that if $\alpha \notin \mathsf{Valid}$ then for any input value of an honest the sender, $f(\alpha, \beta)$ is defined to be a uniformly random value from $\{0,1\}^\ell$.

Let Π be an AH two-message protocol for f with λ outputs from $\{0,1\}^\ell$, where ℓ is as defined in Thm. 6. Let Π^{cds} be an AH two-message protocol for $\mathsf{CDS}_\ell^{\mathcal{S}}$. The basic idea is to forked-compose—without using any CPIR—an instantiation of Π with an instantiation of Π^{cds} on the same inputs α and β. Assume that the query phases of Π and Π^{cds} are the same; this is possible since in the $\mathsf{CDS}_\ell^{\mathsf{Valid}}$ protocol of Thm. 5, the receiver learns a secret $t \in \{0,1\}^{\ell\lambda}$ iff $\alpha \in \mathsf{Valid}$, and the sender learns the corresponding ciphertexts $P_i = \mathsf{Enc}_{\mathsf{pk}}(\alpha_i)$. Thus, in the

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

Receiver's first message:
For $i \in [\mu]$: Let P_i be receiver's first Π-message on input α_i. // I.e., $P_i = \mathsf{Enc}_{\mathsf{pk}}(\alpha)$.
Send $(P_1, \ldots, P_\mu)$ to the sender.
Sender'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 $\Delta'_j \leftarrow \Delta_j \cdot \mathsf{Enc}_{\mathsf{pk}}(t_j)$.
Send $(\Delta'_1, \ldots, \Delta'_\lambda; \{c_{i1}\}, \ldots, \{c_{i\lambda}\})$ to the receiver.
Recovery:
For $j \in [\lambda]$: The receiver recovers t_j from $(\alpha, \mathsf{q}, \{c_{ij}\})$ by using the recovery algorithm of Π^{cds}.
She recovers δ'_j from $(\alpha; \mathsf{q}; \Delta'_j)$ 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 f in malicious model by using additive CDS transformation

transfer phase, the sender can use the ciphertexts P_i as input to an AH two-message protocol Π that evaluates f. Finally, the sender 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 $\Delta_i \cdot \mathsf{Enc}_{\mathsf{pk}}(t_i)$ to the receiver. Thus, the receiver can peel off the masks t_i and recover the outputs iff her inputs are in the correct range.

Theorem 7. *Fix an AH public-key cryptosystem* $\mathsf{PKC} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. *Assume the PKI model, and fix a secret and public key pair* $(\mathsf{sk}, \mathsf{pk})$. *Let* Π^{cds} *be an* $(\varepsilon, \tau; \sigma_1)$*-relaxed-secure AH two-message protocol for* $\mathsf{CDS}_\ell^{\mathsf{Valid}}$. *Let* Π *be an* $(\varepsilon, \tau; \sigma_2)$*-secure AH two-message protocol for computing* f *in the semihonest model, such that* Π^{cds} *and* Π *have a common algorithm for computing receiver's first message. Then Prot. 2 is an* $(\varepsilon, \tau - O(1); \sigma_1 + \sigma_2 + \sigma_3)$*-relaxed-secure AH two-message protocol for computing* f*, where* $\sigma_3 = 2^\ell \lambda / n$.

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 σ_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.

Optimizations. 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 sender-security 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 sender-secure as long as prf is secure.

An example application: private scalar product protocol. In a *private scalar product protocol* protocol, the receiver has a Boolean vector α of dimension μ and the sender has a Boolean vector β of the same dimension. Receiver's private output is δ, such that $\delta = \sum_{i=1}^{\mu} \beta_i \alpha_i$, and the sender has no private output. It is simple to compute this functionality in the semihonest model [GLLM04,WY04]: Assume that c_i is a random encryption of α_i. Then, the sender sends $\Delta = \sum_{i=1}^{\mu} c_i^{\beta_i} \cdot \mathsf{Enc}_{\mathsf{pk}}(0)$ to the receiver. The receiver decrypts Δ. It is straightforward to apply the CDS transformation to get a protocol that is sender-secure 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).

9 Implementing Protocols in The Standard Model

Recall that in the PKI model, a trusted key generator TKG generates a key pair $(\mathsf{sk}, \mathsf{pk})$ by executing Gen, transferring $(\mathsf{sk}, \mathsf{pk})$ to the receiver and pk to the sender. The presence of PKI is normal in e-voting or in many applications of oblivious transfer, where a correct pk has been already defined by an upper level protocol that for efficiency reasons lets several different subprotocols to use the same key. In fact, in several places we already explicitly used a parallel composition of two or more AH protocols that use the same key. Because this key can be reused in other homomorphic protocols, this model is weaker than the common reference string protocol. Moreover, the constructed protocols (including the ones that use the new CDS transform) are in most of the cases considerably more efficient than the up-to-date non-interactive proofs of knowledge in the stronger common reference string model.

We can get rid of the PKI model by letting the receiver to execute once, separately and in an isolated manner (i.e., no other messages of different protocols are sent by the receiver at the same time), with every sender a zero-knowledge proof of knowledge that pk is valid and that he knows the corresponding secret key. This is followed by the real protocol. In the security proof, the simulator extracts the secret key by rewinding and thereafter continues to work as previously. Since we require statistical sender-security—and thus can use an unbounded simulator—then it is actually sufficient to have a zero-knowledge proof that the key is correct: the simulator just computes the secret key corresponding to the (correct) public key. Note that it is not relevant whether the receiver computes the public key with a correct distribution since for the proof we only need the existence of the secret key.

Next, we propose another possible solution that works in standard model, does not need extra rounds but needs an extra amount of computations by an honest sender. Namely, the extractor, described by Thm. 3, guarantees the security if $\mathsf{spf}(n)$ is large. (Some extra care might be needed, see [Kal05]. Note that the known AH cryptosystems like the Paillier remain homomorphic even if n is incorrectly formed.) Thus, the public key verification can just consist of verifying that $\mathsf{spf}(n) \geq p$ for some suitably large p. If p is not too large then this can be done efficiently by using Lenstra's Elliptic Curve Method [Len87] that works in time $\exp((\sqrt{2}+o(1))\sqrt{\ln p \cdot \ln\ln p})$ [Zim06a,ZD06]. If we want sender's computation to be polynomial in $\log n$ then we have to take $\ln^{\varepsilon} n = \exp((\sqrt{2}+o(1))\sqrt{\ln p \cdot \ln\ln p})$ or $\big((\ln\varepsilon + \ln\ln n)/(\sqrt{2}+o(1))\big)^2 = \ln p \cdot \ln\ln p$, or $p = 2^{(1/\sqrt{2}+o(1))\ln^2 \ln n}$.

In concrete numbers, for example, assume that ECM is "efficient" for 88-bit $\mathsf{spf}(n)$ and that $\sigma = 2^{-40}$, where σ is as in Thm. 4; the latter choice of σ is most probably sufficient in practice. Then, in the case of the DIE protocol, one has $\ell = 47$, which is sufficient for several applications. We verified this approach by using the suggested optimal parameters from [Zim06b], on an AMD Athlon 64 3000+ processor by using the GMP-ECM software. As an example, if $n = pq$, where q is an 88-bit prime and q is an $(1024-88)$-bit prime then one has to run the ECM algorithm on expected 206 curves with bounds $B1 = 50\,000$ and $B2 = 5\,000\,000$. Testing on one curve with these parameters takes ≈ 2.5 seconds, and thus testing that $\mathsf{spf}(n) \geq 2^{89}$ takes an expected 9 minutes. On the other hand, if q is an 66-bit prime then it takes 77 expected curves with bounds $B1 = 11\,000$ and $B2 = 1\,100\,000$. On the same platform, testing one curve with these parameters takes ≈ 0.66 seconds, and thus testing that $\mathsf{spf}(n) \geq 2^{67}$ takes less than expected 51 seconds. Given the advances in the ECM [Zim06a], we would expect the quoted timings to decrease dramatically over the next few years.

Acknowledgements. We would like to thank Phil Carmody, Yuval Ishai, 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 6848.

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.

[BL88] Josh Benaloh and Jerry Leichter. Generalized Secret Sharing and Monotone Functions. In S. Goldwasser, editor, *Advances in Cryptology—CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 27–35, Santa Barbara, California, USA, 21–25 August 1988. Springer-Verlag, 1990.

[CCKM00] Christian Cachin, Jan Camenisch, Joe Kilian, and Joy Müller. One-Round Secure Computation and Secure Autonomous Mobile Agents. In Ugo Montanari, José D. P. Rolim, and Emo Welzl, editors, *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000*, volume 1853 of *Lecture Notes in Computer Science*, pages 512–523, Geneva, Switzerland, July 9–15, 2000. Springer-Verlag.

[CFGP05] Olivier Chevassut, Pierre-Alain Fouque, Pierrick Gaudry, and David Pointcheval. Key Derivation and Randomness Extraction. Technical Report 2005/061, IACR, March 19 2005.

[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.

[Elg85] Taher Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 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 Private 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.

[Kal05] Yael Tauman Kalai. Smooth Projective Hashing and Two-Message Oblivious Transfer. In Ronald Cramer, editor, *Advances in Cryptology — EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 78–95, Aarhus, Denmark, May 22–26, 2005. Springer-Verlag.

[KM05a] 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.

[KM05b] Robert König and Ueli M. Maurer. Generalized Strong Extractors and Deterministic Privacy Amplification. In Nigel P. Smart, editor, *Cryptography and Coding, 10th IMA International Conference*, pages 322–339, Cirencester, UK, December 19–21, 2005.

[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.

[Len87] Hendrik W. Lenstra, Jr. Factoring integers with Elliptic Curves. *Annals of Mathematics*, 126(2):649–673, 1987.

[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.

[ZD06] Paul Zimmermann and Bruce Dodson. 20 Years of ECM. In ?, editor, *The Algorithmic Number Theory Symposium*, volume ? of *Lecture Notes in Computer Science*, pages ?–?, Berlin, Germany, 23–28 July 2006. Springer-Verlag. ISBN 3-540-67695-3.

[Zim06a] Paul Zimmermann. 20 Years of ECM. Technical Report 5834, INRIA, February 2006. Available at `http://www.inria.fr/rrrt/rr-5834.html`.

[Zim06b] Paul Zimmermann. Optimal Parameters for ECM. Available at `http://www.loria.fr/~zimmerma/records/ecm/params.html`, as of May, 2006, 2006.

A On Polynomial Security

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 $k = \Omega(\log^{3-o(1)} \kappa)$.

B Proof of Theorem 1

Proof. Computational receiver-privacy: A standard hybrid argument since the sender only sees μ different ciphertexts. *Statistical sender-security:* The universal simulator Sim does the following. First, it generates $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}$ and sends $(\mathsf{pk}, \mathsf{sk})$ to the malicious receiver and pk to the sender. After receiving $\mathfrak{q}$ from the receiver, it verifies that $\mathfrak{q}$ is correctly formed. If $\mathfrak{q}$ is malformed, Sim sends the halt symbol $\perp$ to TTP and provides no second message to the receiver. Otherwise, $\mathfrak{q} = (\mathsf{Enc}_{\mathsf{pk}}(\alpha_1^*; r_1), \ldots, \mathsf{Enc}_{\mathsf{pk}}(\alpha_\mu^*; r_\mu))$ for some α_i^*, r_i. Sim uses sk to obtain receiver's protocol inputs $\alpha^* = (\alpha_1^*, \ldots, \alpha_\mu^*)$, and sends α^* to TTP. After obtaining both α^* and sender's input β, TTP answers with $f^* := f(\alpha^*, \beta)$. Then Sim computes $\mathfrak{a}^* \leftarrow \Sigma(\mathsf{pk}, f^*)$, and outputs whatever the receiver outputs on the input $\mathfrak{a}^*$. Since the honest sender did not output anything, we have to consider only the output of the receiver. Because $\mathfrak{a}^* \overset{\sigma}{\sim} \mathcal{A}(\alpha, \beta)$ then receiver's output on $\mathfrak{a}^*$ is σ-close to his output on random message from $\mathcal{A}(\alpha, \beta)$. □

C Proof of Lemma 1

Proof. Computational receiver-privacy is straightforward, since receiver's message is same as in each protocol. *Statistical sender-security.* Let Sim_i be the universal non-rewinding simulator for Π_i. The simulator Sim^f for the forked composition does the following. It generates $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{Gen}$, sends $(\mathsf{sk}, \mathsf{pk})$ to the receiver and pk to the sender. Sim^f receives $\mathfrak{q}$, where $\mathfrak{q}$ is the first common message of all Π_i from the receiver. If the receiver value is malformed then Sim^f outputs $\perp$. Otherwise, Sim^f uses sk to obtain α^* from $\mathfrak{q}$. She sends α^* to TTP. After receiving $f_1(\alpha^*, \beta_1), \ldots, f_m(\alpha^*, \beta_m)$ TTP, Sim^f runs all canonical Sim_i with $f_i(\alpha^*, \beta_i)$, as message from TTP, and combines simulated replies $\mathfrak{a}_i^*$ into a single simulated reply $\mathfrak{a}^*$. The claim follows, as $\mathfrak{a}_i \overset{\sigma_i}{\sim} \mathfrak{a}_i^*$ for all $i \in [m]$. □

D Proof of Theorem 3

For Thm. 3 we 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.) The result is somewhat trivial as Lemma 2 essentially computes the statistical difference between $U(\mathbb{Z}_T) \mod p$ and $U(\mathbb{Z}_p)$ when $p \ll T$.

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

Proof. Fix $m \in \mathbb{Z}_v$. 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 $\gcd(p, v) = 1$, then v is a generator of $\mathbb{Z}_p$. Thus, we can partition $\mathbb{Z}_p$ into two sets $\mathcal{T}_0 = \{c \in \mathbb{Z}_p : \Pr[(m + U(v\mathbb{Z}_T)) \mod p = c] = a/T\}$ and $\mathcal{T}_1 = \{c \in \mathbb{Z}_p : \Pr[(m + U(v\mathbb{Z}_T)) \mod p = c] = (a+1)/T\}$. Then $(m + U(v\mathbb{Z}_T)) \mod p \overset{\varepsilon}{\sim} U(\mathbb{Z}_p)$, 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 $\varepsilon = \max\{((p-b)b)/(Tp), (b(p-b))/(Tp)\} = (b(p-b))/(Tp) \le p/(4T)$. From $T = \lfloor n/v \rfloor \ge (n-v)/v \ge n/(2v)$ it follows that $\varepsilon \le p/(4T) \le n/(4 \cdot \mathsf{spf}(n)T) \le v/(2 \cdot \mathsf{spf}(n))$. □

Proof (Thm. 3). Assume that m is chosen uniformly from $a\mathbb{Z}_n$ for some $a \in \mathbb{Z} \setminus \{0\}$. If $\gcd(a, n) = 1$ then $a\mathbb{Z}_n = \mathbb{Z}_n$ and the claim follows. Otherwise, $a = p$ is a non-trivial factor of n. Set $v := 2^\ell$. Therefore, as $\sharp p\mathbb{Z}_n = n/p$, $\Pr[\mathsf{Ext}(m, U(S)) \equiv y \pmod n] = \Pr[U(p\mathbb{Z}_n) + U(S) \equiv y \pmod n] = p/n \cdot \Pr[U(S) \equiv y \pmod p]$ for any $y \in \mathbb{Z}_n$. Lem. 2 assures that $d\left(U(p\mathbb{Z}_n + U(S)), U(\mathbb{Z}_n)\right) = d\left(U(S) \mod p, U(\mathbb{Z}_p)\right) \le 2^{\ell-1}/\mathsf{spf}(n) \le \varepsilon$. □

E Proof of Theorem 4

Proof. If pk is valid then $\mathsf{Dec}_{\mathsf{sk}}(\mathfrak{a}) \sim \mathsf{Ext}((\alpha^* - b)U(\mathcal{M}), U(2^\ell \cdot \mathbb{Z}_T)) + \beta \mod n$, where $\alpha^* := \mathsf{Dec}_{\mathsf{sk}}(\mathfrak{q})$. *Correctness* is straightforward, since if $\alpha^* = b$ then $\mathsf{Dec}_{\mathsf{sk}}(\mathfrak{a}) \sim \mathsf{Ext}(0 \cdot U(\mathcal{M}), U(2^\ell \cdot \mathbb{Z}_T)) + \beta = U(2^\ell \cdot \mathbb{Z}_T) + \beta$. Since always $\mathsf{Dec}_{\mathsf{sk}}(\mathfrak{a}) < n$ then $\mathsf{Dec}_{\mathsf{sk}}(\mathfrak{a}) \equiv \beta \mod 2^\ell$. According to Thm. 1 we now just must show that Prot. 1 is σ-simulatable. Construct the next $\Sigma_1(\mathsf{pk}, f(\alpha^*, \beta))$: If $f(\alpha^*, \beta) = 1$ (i.e., $\alpha^* = b$) then set $\hat{\mathfrak{a}} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\mathsf{Ext}(0, U(2^\ell \cdot \mathbb{Z}_T)) + \beta)$. Otherwise, set $\hat{\mathfrak{a}} \leftarrow \mathsf{Enc}_{\mathsf{pk}}(U(\mathcal{M}))$. Denote $V := \mathsf{Ext}((\alpha - b)U(\mathcal{M}), U(2^\ell \cdot \mathbb{Z}_T)) + \beta$. In the case of the honest sender, $\mathsf{Dec}_{\mathsf{sk}}(\hat{\mathfrak{a}}) \sim V$. Since $\mathcal{R}$ is a quasigroup then $\hat{\mathfrak{a}} \sim \mathsf{Enc}_{\mathsf{pk}}(V)$. Due to Thm. 3, $d\,(V, U(\mathcal{M})) \leq 2^\ell/(2 \cdot \mathsf{spf}(n)) \leq \sigma$. The claim follows. □

F Proof of Theorem 5

Proof (Sketch). Corollary of Thm. 2/4. The simulator Sim computes the secret key sk corresponding to pk. Then, she 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 sender's second messages. □

G Proof of Theorem 7

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, the receiver obtains $\widehat{t}_j \mod 2^\ell$ and consequently the correct end-result, as there are no modular wrappings. *Receiver-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 receiver-private than Π. *Statistical sender-security*: Clearly, Π' is a forked composition of two statistically sender-secure AH two-message protocols. Therefore, Π' is an $(\sigma_1 + \sigma_2)$-sender-secure 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 $\sigma_3/\lambda = d\,(U(\mathbb{Z}_{n-2^\ell}), U(\mathbb{Z}_n)) = 2^\ell/n$. □