Secure Blind Decryption*

Matthew Green

Johns Hopkins University
Information Security Institute
3400 N. Charles Street; Baltimore, MD 21218, USA
mgreen@cs.jhu.edu

Abstract. In this work we construct public key encryption schemes that admit a protocol for *blindly* decrypting ciphertexts. In a blind decryption protocol, a user with a ciphertext interacts with a secret keyholder such that the user obtains the decryption of the ciphertext and the keyholder learns nothing about what it decrypted. While we are not the first to consider this problem, previous works provided only weak security guarantees against malicious users. We provide, to our knowledge, the first practical blind decryption schemes that are secure under a strong CCA security definition. We prove our construction secure in the standard model under simple, well-studied assumptions in bilinear groups. To motivate the usefulness of this primitive we discuss several applications including privacy-preserving distributed file systems and Oblivious Transfer schemes that admit *public* contribution.

1 Introduction

The past several years have seen a trend towards outsourcing data storage to remote data stores and cloud-based services. While much attention has been paid to securing this data, relatively little has been given to the problem of securing the data's access pattern. This is a real problem for some systems where users' access histories are more sensitive than the data itself, for example patent databases. Even in business there are many practical applications where users' access history is sensitive. For example, the data access patterns of a major corporation's executives could be worth millions of dollars to the right person, particularly in advance of a merger or acquisition.

To address these concerns, many recent works have proposed tools that allow users to transact online without sacrificing their privacy. These tools include (but are not limited to) efficient adaptive oblivious transfer protocols [15, 28, 29, 44], anonymous credential schemes [13, 4], and group signature schemes [16, 7]. One recent application for these tools is to the construction of *oblivious databases* that provide strong access control while preventing the operator from learning

* This work was supported in part by NSF Grant CNS-1010928 and by HHS Grant number 90TR0003/01. The contents of this publication are solely the responsibility of the authors and do not necessarily represent the official views of HHS or any other sponsor.

D. Catalano et al. (Eds.): PKC 2011, LNCS 6571, pp. 265–282, 2011.

which records its users access [20, 12]. Despite this progress, there are still many primitives that we do not know how to implement efficiently using the techniques available to us.

Blind Decryption. In this work we consider one such primitive, which we refer to as *blind decryption*. A blind decryption scheme is a public-key encryption (PKE) scheme that admits an efficient protocol for obliviously decrypting ciphertexts. In this protocol a User who possesses a ciphertext interacts with a Decryptor who holds the necessary secret key. At the conclusion of the protocol, the User obtains the plaintext while the Decryptor learns nothing about what it decrypted. Given that the fundamental purpose of a blind decryption protocol is to decrypt ciphertexts, it seems reasonable to analyze any such protocol with malicious adversaries in mind. Specifically, since such an adversary can implicitly use the blind decryption protocol to decrypt chosen ciphertexts, we will restrict out investigation to secure blind decryption schemes that retain their security even under (adaptive) chosen ciphertext attack.

Blind decryption has many applications to privacy-preserving protocols and systems. For example, blind decryption implies k-out-of-N oblivious transfer [11], which is important theoretically as well as practically for its applications to the construction of oblivious databases [15, 20, 12]. Moreover, blind decryption has practical applications to distributed cryptographic filesystems and for supporting rapid deletion [43].

We are not the first to consider the problem of constructing blind decryption schemes. The primitive was originally formalized by Sakura and Yamane [45] in the mid-1990s, but folklore solutions are thought to have predated that work by more than a decade. Despite an abundance of research in this area, most proposed constructions are insecure under adaptive chosen ciphertext attack [24, 49,41,23,47,42]. Several protocols have recently been proposed containing "blind decryption-like" techniques (see *e.g.,* the simulatable oblivious transfer protocols of [15, 29, 44, 30, 33]). However, these protocols use symmetric (or at least, non-public) encryption procedures, and it does not seem easy to adapt them to the public-key model.

Of course, blind decryption is an instance of secure multi-party computation (MPC) and can be achieved by applying general techniques (*e.g.,* [50, 26, 34]) to the decryption algorithm of a CCA-secure PKE scheme. However, the protocols yielded by this approach are likely to be quite inefficient, making them impractical for real-world applications.

Our Contributions. In this paper we present what is, to our knowledge, the first practical blind decryption scheme that is IND-CCA2-secure in the standard model. We prove our scheme secure under reasonable assumptions in bilinear groups. At the cost of introducing an optional Common Reference String, the protocol can be conducted in a single communication round.

To motivate the usefulness of this new primitive we consider several applications. Chief among these is the construction of privacy-preserving encrypted filesystems (and databases), where a central authority manages the decryption of many ciphertexts without learning users' access patterns. This is important

in situations where the access pattern might leak critical information about the information being accessed. Unlike previous attempts to solve this problem [15, 20, 12], our encryption algorithm is *public*, *i.e.,* users can encrypt new messages offline without assistance from a trusted party. By combining blind decryption with the new oblivious access control techiques of [20, 12] (which use anonymous credentials to enforce complex access control policies) we can achieve strong proactive access control without sacrificing privacy.

Of potential theoretical interest, blind decryption can be used as a building block in constructing adaptive k-out-of-N Oblivious Transfer protocols [15, 29, 44, 30, 33, 37]. In fact, it is possible to achieve a multi-party primitive that is more flexible than traditional OT, in that *any* party can commit messages to the message database (rather than just the Sender). We refer to this enhanced primitive as Oblivious Transfer with Public Contribution (OTPC). We discuss these applications in Section 5.

1.1 Related Work

The first blind decryption protocol is generally attributed Chaum [19], who proposed a technique for blinding an RSA ciphertext in order to obtain its decryption $c^d \ mod \ N$. Since traditional RSA ciphertexts are malleable and hence vulnerable to chosen ciphertext attack, this approach does not lead to a secure blind decryption scheme. Furthermore, standard encryption padding techniques [5] do not seem helpful.

Subsequent works [45, 24, 49] adapted Chaum's approach to other CPA-secure cryptosystems such as Elgamal. These constructions were employed within various protocols, including a 1-out-of-N Oblivious Transfer scheme due to Dodis *et al.* [24]. Unfortunately, since the cryptosystems underlying these protocols are not CCA-secure, security analyses of those protocols frequently required strong assumptions such as honest-but-curious adversaries[1]. Mambo, Sakurai and Okamoto [41] proposed to address chosen ciphertext attacks by signing the ciphertexts to prevent an adversary from mauling them. Their *transformable signature* could be blinded in tandem with the ciphertext. The trouble with this approach and other related approaches [15, 29, 30, 33, 44] is that the encryption scheme is no longer a PKE, since encryption now requires a knowledge of a secret signing key (furthermore, these transformable signatures were successfully cryptanalyzed [23]). Schnorr and Jakobsson [47] proposed a scheme secure under the weaker *one-more decryption attack* and used this to construct a PIR protocol. Unfortunately, their protocol is secure only for *random* messages, and furthermore cannot be extended to construct stronger primitives such as simulatable OT [15].

Recently, Green and Hohenberger [28] proposed a technique for blindly extracting decryption keys in an Identity-Based Encryption scheme. Subsequently, Ogata and Le Trieu [42] used this tool to obtain a weak blind decryption scheme

[1] For example, Dodis *et al.* [24] analyzed their 1-out-of-N oblivious transfer construction in the honest-but-curious model.

(by encrypting ciphertexts under a random identity, then blindly extracting the appropriate secret key). The resulting protocol is efficient, but the ciphertexts are malleable and thus vulnerable to adaptive chosen ciphertext attack.

1.2 Intuition

Ideally the development of a blind decryption scheme would begin with an existing CCA-secure PKE, and would only require us to develop an efficient two-party protocol for computing the decryption algorithm. Indeed, the literature provides us with many candidate PKE constructions that can be so adapted if we are willing to accept the costs associated with general multi-party computation techniques [50, 26, 34].

However, in this work we are interested in protocols that are both secure and *practical*. This rules out inefficient gate-by-gate decryption protocols, limiting us to a relatively small collection of techniques that can be used to build efficient protocols. This toolbox includes primitives such as homomorphic commitment schemes, which we might combine with zero knowledge proofs for statements involving algebraic relations among cyclic group elements, *e.g.,* [46, 31]. While these techniques have been deployed successfully to construct other privacy-preserving protocols, there are strict limitations on what they can accomplish.

To illustrate this point, let us review several of the most popular encryption techniques in the literature. Random oracle paradigms such as OAEP [5] and Fujisaki-Okamoto [25] seem fundamentally difficult to adapt, since these approaches require the decryptor to evaluate an ideal hash function on a partially-decrypted value *prior* to outputting a result. Even the more efficient standard-model CCA-secure paradigms such as Cramer-Shoup [22] and recent bilinear constructions (*e.g.,* [8, 10, 35]) require components that we cannot efficiently adapt. For example, when implemented in a group $\mathbb{G}$ of order p, the Cramer-Shoup scheme assumes a collision-resistant mapping $H : \mathbb{G} \times \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{Z}_p$. We know of no efficient two-party technique for evaluating such a function[2].

Our approach. Rather than adapt an existing scheme, we set out to design a new one. Our approach is based on the TBE-to-PKE paradigm proposed independently by Canetti *et al.* [18] and MacKenzie *et al.* [40]. This technique converts a Tag-Based Encryption (TBE) scheme into a CCA-secure public PKE with the assistance of a strongly unforgeable one-time signature (OTS). In this generic transform, encryption is conducted by first generating a keypair (vk, sk) for the OTS, encrypting the message using the TBE with vk as the tag, then signing the resulting ciphertext with sk. Intuitively the presence of the signature (which is verified at decryption time) prevents an adversary from mauling the ciphertext.

To blindly decrypt such a ciphertext, we propose the following approach: the User first commits to the ciphertext and vk using a homomorphic commitment or encryption scheme. She then efficiently proves knowledge of the associated signature for these committed values. If this proof verifies, the Decryptor

[2] Conceivably it might be possible to develop one, however it might be tied to the specific construction of $\mathbb{G}$ and thus be be quite inflexible.

may then apply the TBE decryption algorithm to the (homomorphically) committed ciphertext, secure in the knowledge that the commitment contains an appropriately-distributed value. Finally, the result can be opened by the User.

For this protocol to be efficient, we must choose our underlying primitives with care. Specifically, we must ensure that (1) the OTS verification key maps to the tag-space of the TBE, (2) and the TBE ciphertext maps to the message space of the OTS. Of course, the easiest way to achieve these goals is to use an OTS that directly signs the TBE ciphertext space, with a TBE whose tag-space includes the OTS verification keyspace. These primitives must admit efficient protocols for the operations we will conduct with them. Finally, we would like to avoid relying on complex or novel complexity assumptions in order to achieve these goals.

Our proposed construction is based on a variant of Cramer-Shoup that was adapted by Shacham [48] for security in bilinear groups. We first modify Shacham's construction into a TBE with the following ciphertext structure. Let $\alpha \in \mathbb{Z}_p^*$ be an arbitrary ciphertext tag and $m \in \mathbb{G}$ a message to be encrypted. Given a public key $g, g_1, g_2, g_3, h_1, h_2, c_1, c_2, d_1, d_2 \in \mathbb{G}$ an encryptor selects random elements $r_1, r_2 \in \mathbb{Z}_p^*$ and outputs the ciphertext:

$$(u_1, u_2, u_3, e, v, vk) = (g_1^{r_1}, g_2^{r_2}, g_3^{r_1+r_2}, m \cdot h_1^{r_1} h_2^{r_2}, (c_1 d_1^{\alpha})^{r_1} \cdot (c_2 d_2^{\alpha})^{r_2}, g^{\alpha})$$

An important feature of this construction is that the decryptor does *not* need to know the tag value α[3]. Therefore, in constructing our PKE we can "dual-purpose" α as both the ciphertext tag *and* as the secret key of a one-time signature (OTS) scheme. Specifically, our encryption process will select a random α, encrypt the message using the TBE with α as the tag, and finally sign the resulting elements (u_1, u_2, u_3, e, v) under α. The resulting ciphertext contains $(u_1, u_2, u_3, e, v, vk)$ along with the signature on those values.

The remaining challenge is therefore to construct an efficient OTS that can sign multiple bilinear group elements, yet admits an efficient proof-of-knowledge for a signature on committed elements. To address this we propose a new multi-block one-time "F-signature" that we believe may be of independent interest[4]. Interestingly, our signing algorithm does not actually operate on elements of $\mathbb{G}$, but rather signs message vectors of the form $(m_1, \ldots, m_n) \in \mathbb{Z}_p^{*n}$ (for some arbitrary vector length n). Once a message is signed, however, the signature can be *verified* given the tuple $(g^{m_1}, \ldots, g^{m_n}) \in \mathbb{G}^n$, rather than the original message

[3] This differs from many other candidate TBE and IBE schemes, *e.g.,* Boneh and Boyen's IBE [6] and Kiltz's TBE [35] where the tag/identity is an element of $\mathbb{Z}_p^*$ and *must* be provided at decryption time (or in the case of IBE, when a secret key is extracted). This requirement stems from the nature of those schemes' security proofs.

[4] F-signature is a contraction of F-unforgeable signature, which is a concept proposed by Belinkiy *et al.* [4], and later developed by Green and Hohenberger [30]. In this paradigm, the signing algorithm operates on a message m, but there exists a signature verification algorithm that can operate given only $F(m)$ for some one-way function F.

vector. Strictly speaking, this construction does not meet our requirements—an encryptor won't always know the discrete logarithm base g of (u_1, u_2, u_3, e, v). Our key insight is to show that encryptors can produce an identically distributed "workalike" signature even when the discrete logarithms are not known. We prove that, in the context of our encryption scheme, no adversary can forge these workalike signatures. Our signature construction is presented independently in Appendix 2.4.

2 Technical Preliminaries

2.1 Bilinear Groups and Cryptographic Assumptions

Let λ be a security parameter. We define $\mathsf{BMsetup}$ as an algorithm that, on input 1^λ, outputs the parameters for a bilinear mapping as $\gamma = (p, \mathbb{G}, \mathbb{G}_T, e, g \in \mathbb{G})$, where g generates $\mathbb{G}$, the groups $\mathbb{G}, \mathbb{G}_T$ each have prime order p, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. For $\langle g \rangle = \langle h \rangle = \mathbb{G}$ the efficiently-computable mapping e must be both *non-degenerate* ($\langle e(g,h) \rangle = \mathbb{G}_T$) and *bilinear* (for $a, b \in \mathbb{Z}_p^*$, $e(g^a, h^b) = e(g,h)^{ab}$).

The Decision Linear Assumption (DLIN) [7]**.** Let $\mathbb{G}$ be a group of prime order $p \in \Theta(2^\lambda)$. For all p.p.t. adversaries $\mathcal{A}$, the following probability is $1/2$ plus an amount negligible in λ: $\Pr[f, g, h, z_0 \xleftarrow{R} \mathbb{G}; a, b \xleftarrow{R} \mathbb{Z}_p^*; z_1 \leftarrow h^{a+b}; d \xleftarrow{R} \{0,1\}; d' \leftarrow \mathcal{A}(f, g, h, f^a, g^b, z_d) : d = d']$.

The Flexible Diffie-Hellman Assumption (FDH) [36,30]**.** Let $\mathbb{G}$ be a group of prime order $p \in \Theta(2^\lambda)$. For all p.p.t. adversaries $\mathcal{A}$, the following probability is negligible in λ: $\Pr[g, g^a, g^b; a, b \xleftarrow{R} \mathbb{Z}_p^*; (w, w') \leftarrow \mathcal{A}(g, g^a, g^b) : w \neq 1 \wedge w' = w^{ab}]$.

This assumption was previously described as the 2-out-of-3 CDH assumption by Kunz-Jacques and Pointcheval [36]. We adopt the name Flexible Diffie-Hellman for consistency with recent work [39,30]. To instill confidence in this assumption, Green and Hohenberger [30] showed that a solver for the Flexible Diffie-Hellman problem implies a solver for a related decisional problem, the Decisional 3-Party Diffie-Hellman assumption (3DDH) which has been used several times in the literature [38, 9, 32, 30].

2.2 Proofs of Knowledge

We use several standard results for proving statements about the satisfiability of one or more pairing-product equations. For variables $\{\mathcal{X}\}_{1\ldots n} \in \mathbb{G}$ and constants $\{\mathcal{A}\}_{1\ldots n} \in \mathbb{G}, a_{i,j} \in \mathbb{Z}_p^*$, and $t_T \in \mathbb{G}_T$, these equations have the form:

$$\prod_{i=1}^{n} e(\mathcal{A}_i, \mathcal{X}_i) \prod_{i=1}^{n} \prod_{j=1}^{n} e(\mathcal{X}_i, \mathcal{X}_j)^{a_{i,j}} = t_T$$

The proof-of-knowledge protocols in this work can be instantiated using one of two approaches. The first approach is to use the interactive zero-knowledge

proof technique of Schnorr [46], with extensions due to *e.g.,* [21, 14, 17, 2, 15]. Note that this may require that the proofs be executed sequentially (indeed, this requirement is explicit in our security definitions). For details, see the work of Adida *et al.* [2], which provides a taxonomy of interactive proof techniques for pairing-based statements.

Alternatively, the proofs can be instantiated using the Groth-Sahai proof system [31] which permits efficient non-interactive proofs of the satisfiability of multiple pairing product equations. In the general case these proofs are witness indistinguishable. However a subset of special cases (including where $t_T = 1$) may be conducted in zero-knowledge[5]. The Groth-Sahai system can be instantiated under the Decision Linear assumption in the Common Reference String model.

We refer the reader to the cited works for formal security definitions of ZK and WI proof systems. In our security analysis we will assume some generic instantiation Π_{ZK} that is secure under the Decision Linear assumption in $\mathbb{G}$. Either of the techniques mentioned above can satisfy this requirement. When referring to WI and ZK proofs we will use the notation of Camenisch and Stadler [16]. For instance, $\mathsf{WIPoK}\{(g,h) : e(g,h) = T \ \wedge \ e(g,v) = 1\}$ denotes a witness indistinguishable proof of knowledge of elements g and h that satisfy both $e(g,h) = T$ and $e(g,v) = 1$. All values not in enclosed in ()'s are assumed to be known to the verifier.

2.3 Linear Encryption

Our blind decryption protocol employs a multiplicatively homomorphic scheme that encrypts elements of $\mathbb{G}$. We instantiate this scheme with the Linear Encryption scheme of Boneh, Boyen and Shacham [7] which is semantically secure under the Decision Linear assumption. Ciphertexts in this scheme have the form $(c_1, c_2, c_3) \in \mathbb{G}^3$, and the homomorphic operation is simple pairwise multiplication. Exponentiation by a scalar z can be performed as c_1^z, c_2^z, c_3^z. To re-randomize a ciphertext one multiplies it by $\mathsf{LE.Enc}(pk, 1)$. Our protocols also require an efficient ZK proof-of-knowledge of the plaintext m underlying a ciphertext C, which we denote by $\mathsf{ZKPoK}\{(m) : C \in \mathsf{LE.Enc}(pk, m)\}$. We refer the reader to the full version [27] for formal algorithm descriptions.

2.4 A One-Time *F*-Signature on Multiblock Messages

Our constructions require a strongly unforgeable one-time F-signature scheme that signs messages of the form $(m_1, \ldots, m_N) \in \mathbb{Z}_p^{*n}$ (for arbitrary values of n), but can *verify* signatures given only a function of the messages, specifically, $(g_1^{m_1}, \ldots, g_n^{m_n}) \in \mathbb{G}^n$ for fixed $g_1, \ldots, g_n \in \mathbb{G}$. Note that $g_1, \ldots, g_n$ need not be distinct.

[5] In many cases it is easy to re-write pairing products equation as a composition of multiple distinct equations having $t_T = 1$ (see [31]). Although we do not explicitly perform this translation in our protocols, we note that it can be applied to all of the ZKPoKs used in our constructions.

To construct FS, we adapt a weakly-unforgeable signature due to Green and Hohenberger [30] to admit multi-block messages, while simplifying the scheme into a one-time signature. The latter modification has the incidental effect of strengthening the signature to be strongly unforgeable. Let us now describe FS:

FS.KG. On input group parameters γ, a vector length n, select g, g_1, ..., g_n, v, d, u_1, ..., $u_n \xleftarrow{R} \mathbb{G}$ and $a \xleftarrow{R} \mathbb{Z}_p^*$. Output $vk = (\gamma, g, g^a, v, d, g_1, \ldots, g_n, u_1, \ldots, u_n, n)$ and $sk = (vk, a)$.

FS.Sign. Given sk and a message vector $(m_1, \ldots, m_n) \in \mathbb{Z}_p^{*n}$, first select $r \xleftarrow{R} \mathbb{Z}_p^*$ and output the signature $\sigma = ((\prod_{i=1}^n u_i^{m_i} \cdot v^r \ d)^a, \ g_1^{am_1}, \ \ldots, \ g_n^{am_n}, \ u_1^{m_1}, \ \ldots, \ u_n^{m_n}, \ r)$.

FS.Verify. Given $pk, (g_1^{m_1}, \ldots, g_n^{m_n})$, parse $\sigma = (\sigma_1, e_1, \ldots, e_n, f_1, \ldots, f_n, r)$, output 1 if the following check holds: $e(\sigma_1, g) = e(\prod_{i=1}^n f_i \cdot v^r d, g^a) \ \wedge \ \{e(g_i^{m_i}, g^a) = e(e_i, g) \ \wedge \ e(g_i^{m_i}, u_i) = e(g_i, f_i)\}_{i \in [1,n]}$.

Note that verification is a pairing product equation. Thus we can efficiently prove knowledge of a signature using the techniques described in Section 2.2. We denote such a proof by *e.g.,* $\mathsf{WIPoK}\{(\sigma) : \mathsf{Verify}(vk, (g^{m_1}, \ldots, g^{m_n}), \sigma) = 1\}$. Note that vk or the messages may reside within a commitment. In the full version of this paper [27] we provide details on these proofs of knowledge, as well as definitions of security and a proof that FS is strongly unforgeable under the Flexible Diffie-Hellman assumption.

Workalike signatures. Our blind decryption constructions make use of the "workalike" algorithms (WAKG, WASign). While the public outputs of these algorithms are identically distributed those of KG and Sign, the WASign algorithm operates on messages of the form $(g_1, \ldots, g_n) \in \mathbb{G}^n$. We stress that (WAKG, WASign, Verify) is *not* a secure signature scheme on arbitrary group elements, but can be used securely under the special conditions of our constructions..

FS.WAKG. Select $x_1, \ldots, x_n \xleftarrow{R} \mathbb{Z}_p^*$ and set $(u_1, \ldots, u_n) = (g^{x_1}, \ldots, g^{x_n})$. Compute the remaining elements as in KG and set $sk = (vk, a, x_1, \ldots, x_n)$.

FS.WASign. Given a message vector $(h_1, \ldots, h_n) \in \mathbb{G}^n$, first select $r \xleftarrow{R} \mathbb{Z}_p^*$ and output the signature $\sigma = ((\prod_{i=1}^n h_i^{x_i} \cdot v^r d)^a, h_1^a, \ldots, h_n^a, h_1^{x_1}, \ldots, h_n^{x_n}, r)$.

3 Definitions

Notation: Let $\mathcal{M}$ be the message space and $\mathcal{C}$ be the ciphertext space. We write $P(\mathcal{A}(a), \mathcal{B}(b)) \rightarrow (c, d)$ to indicate the protocol P is between parties $\mathcal{A}$ and $\mathcal{B}$, where a is $\mathcal{A}$'s input, c is $\mathcal{A}$'s output, b is $\mathcal{B}$'s input and d is $\mathcal{B}$'s output. We will define $\nu(\cdot)$ as a negligible function.

Definition 1 (Blind Decryption Scheme). A public-key blind decryption scheme consists of a tuple of algorithms (KG, Enc, Dec) and a protocol BlindDec.

$\mathsf{KG}(1^\lambda)$. On input a security parameter λ, the key generation algorithm KG outputs a public key pk and a secret key sk.

$\mathsf{Enc}(pk, m)$. On input a public key pk and a message m, Enc outputs a ciphertext C.

$\mathsf{Dec}(pk, sk, C)$. On input pk, sk and a ciphertext C, Dec outputs a message m or the error symbol $\perp$.

The two-party protocol $\mathsf{BlindDec}$ is conducted between a user $\mathcal{U}$ and a decryptor $\mathcal{D}$:

$\mathsf{BlindDec}(\{\mathcal{U}(pk, C)\}, \{\mathcal{D}(pk, sk)\}) \rightarrow (m, \mathsf{nothing})$. On input pk and a ciphertext C, an honest user $\mathcal{U}$ outputs the decryption m or the error symbol $\perp$. The decryptor $\mathcal{D}$ outputs $\mathsf{nothing}$ or an error message.

We now present the standard definition of adaptive chosen ciphertext security for public key encryption.

Definition 2 ($\mathsf{IND\text{-}CCA2}$). A public key encryption scheme $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is $\mathsf{IND\text{-}CCA2}$ secure if every *p.p.t.* adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has advantage $\leq \nu(\lambda)$ in the following experiment.

$$
\begin{array}{l}
\mathsf{IND\text{-}CCA2}(\Pi, \mathcal{A}, \lambda) \\
\quad (pk, sk) \leftarrow \mathsf{KG}(1^\lambda) \\
\quad (m_0, m_1, z) \leftarrow \mathcal{A}_1^{\mathcal{O}_{dec}(pk, sk, \cdot)}(pk) \text{ s.t. } m_0, m_1 \in \mathcal{M} \\
\quad b \leftarrow \{0,1\}; c^* \leftarrow \mathsf{Enc}(pk, m_b) \\
\quad b' \leftarrow \mathcal{A}_2^{\mathcal{O}'_{dec}(pk, sk, \cdot)}(c^*, z) \\
\quad \text{Output } b'
\end{array}
$$

Where $\mathcal{O}_{dec}$ is an oracle that, on input a ciphertext c, returns $\mathsf{Dec}(pk, sk, c)$ and $\mathcal{O}'_{dec}$ operates identically but returns $\perp$ whenever $c = c^*$. We define $\mathcal{A}$'s advantage in the above game by:

$$|\mathbf{Pr}\,[\, b = b' \,] - 1/2|$$

Additional security properties. A secure blind decryption scheme must possess the additional properties of *leak-freeness* and *blindness*. Intuitively, leak-freeness [28] ensures that an adversarial User gains no more information from the blind decryption protocol than she would from access to a standard decryption oracle. Blindness prevents a malicious Decryptor from learning *which* ciphertext a User is attempting to decrypt. Let us now formally state these properties.

Definition 3 (Leak-Freeness [28]). A protocol $\mathsf{BlindDec}$ associated with a PKE scheme $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is *leak free* if for all *p.p.t.* adversaries $\mathcal{A}$, there exists an efficient simulator $\mathcal{S}$ such that for every value λ, no *p.p.t.* distinguisher D can distinguish the output of Game Real from Game Ideal with non-negligible advantage:

Game Real: Run $(pk, sk) \leftarrow \mathsf{KG}(1^\lambda)$ and publish pk. As many times as D wants, $\mathcal{A}$ chooses a ciphertext C and atomically executes the $\mathsf{BlindDec}$ protocol with $\mathcal{D}$:

$\mathsf{BlindDec}(\{\mathcal{U}(pk, C)\}, \{\mathcal{D}(pk, sk)\})$. $\mathcal{A}$'s output (which is the output of the game) includes the list of ciphertexts and decrypted plaintexts.

Game Ideal: A trusted party runs $(pk, sk) \leftarrow \mathsf{KG}(1^\lambda)$ and publishes pk. As many times as D wants, $\mathcal{S}$ chooses a ciphertext C and queries the trusted party to obtain the output of $\mathsf{Dec}(pk, sk, C)$, if $C \in \mathcal{C}$ and $\perp$ otherwise. $\mathcal{S}$'s output (which is the output of the game) includes the list of ciphertexts and decrypted plaintexts.

In the games above, $\mathsf{BlindDec}$ and Dec are treated as atomic operations. Hence D and $\mathcal{A}$ (or $\mathcal{S}$) may communicate at any time except during the execution of those protocols. Additionally, while we do not explicitly specify that auxiliary information is given to the parties, this information must be provided in order to achieve a sequential composition property.

Definition 4 (Ciphertext Blindness). Let $\mathcal{O}_\mathcal{U}(pk, C)$ be an oracle that, on input a public key and ciphertext, initiates the User's portion of the $\mathsf{BlindDec}$ protocol, interacting with an adversary. A protocol $\mathsf{BlindDec}(\mathcal{U}(\cdot,\cdot)\ \mathcal{A}(\cdot,\cdot))$ is Blind secure if every *p.p.t.* adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ has advantage $\leq \nu(\lambda)$ in the following game.

$$\begin{array}{l}\mathsf{Blind}(\mathsf{BlindDec}, \mathcal{A}, \lambda) \\ \quad (pk, C_0, C_1, z) \leftarrow \mathcal{A}_1(1^\lambda) \\ \quad b \leftarrow \{0,1\}; b' \leftarrow \mathcal{A}_2^{\mathcal{O}_\mathcal{U}(pk, C_b), \mathcal{O}_\mathcal{U}(pk, C_{b-1})}(z)\end{array}$$

We define $\mathcal{A}$'s advantage in the above game as: $|\Pr[b' = b] - 1/2|$. Note that a stronger notion of blindness is *selective-failure* blindness, which was proposed by Camenisch *et al.* [15]. While our constructions do not natively achieve this definition, in section 4.1 we discuss techniques for achieving this stronger definition.

Definition 5 (CCA2-secure Blind Decryption). A blind decryption scheme $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{BlindDec})$ is $\mathsf{IND\text{-}CCA2}$-secure if and only if: (1) $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is $\mathsf{IND\text{-}CCA2}$-secure, (2) $\mathsf{BlindDec}$ is leak free, and (3) $\mathsf{BlindDec}$ possesses the property of ciphertext blindness.

4 Constructions

We now present a blind decryption scheme BCS that is secure under the Decision Linear and Flexible Diffie-Hellman assumptions. BCS is based on a variant of Cramer-Shoup that was proposed by Shacham [48], with significant extensions to permit blind decryption.

The core algorithms. We now describe the algorithms $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, which are responsible for key generation, encryption and decryption respectively. BCS encrypts elements of $\mathbb{G}$, which may necessitate an encoding scheme from other message spaces (see *e.g.,* [3]).

$\mathsf{BCS.KG}(1^\lambda)$. First sample $\gamma = (p, \mathbb{G}, \mathbb{G}_T, \hat{e}, g \in \mathbb{G}) \leftarrow \mathsf{BMsetup}(1^\lambda)$. Choose $g, g_1, g_2, g_3, v', d', u'_1, \ldots, u'_5 \xleftarrow{R} \mathbb{G}$, and $x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3 \xleftarrow{R} \mathbb{Z}_p^*$ and compute:

$$\begin{array}{lll} c_1 \leftarrow g_1^{x_1} g_3^{x_3} & d_1 \leftarrow g_1^{y_1} g_3^{y_3} & h_1 \leftarrow g_1^{z_1} g_3^{z_3} \\ c_2 \leftarrow g_2^{x_2} g_3^{x_3} & d_2 \leftarrow g_2^{y_2} g_3^{y_3} & h_2 \leftarrow g_2^{z_2} g_3^{z_3} \end{array}$$

Output $pk = (\gamma, g, g_1, g_2, g_3, c_1, c_2, d_1, d_2, h_1, h_2, v', d', u'_1, \ldots, u'_5)$, $sk = (x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3)$.

$\mathsf{BCS.Enc}(pk, m \in \mathbb{G})$. Select $\alpha, r_1, r_2, c, \psi \xleftarrow{R} \mathbb{Z}_p^*$. Construct a FS keypair (vk_1, sk_1) and a second "workalike" keypair (vk_2, sk_2) as follows:

$$\begin{array}{ll} vk_1 \leftarrow (\gamma, g, g^\alpha, v', d', g_1, g_2, g_3, g, g, u'_1, \ldots, u'_5, 5) & vk_2 = (\gamma, g, g^\psi, v', d', g, g^c, 1) \\ sk_1 \leftarrow (vk_1, \alpha) & sk_2 = (vk_2, \psi, c) \end{array}$$

Next, compute the ciphertext $C = (u_1, u_2, u_3, e, v, vk, e_1, e_2, e_3, f_1, f_2, \sigma_1, \sigma_2)$ as:

$$\begin{array}{l} u_1 \leftarrow g_1^{r_1} \quad u_2 \leftarrow g_2^{r_2} \quad u_3 \leftarrow g_3^{r_1+r_2} \quad e \leftarrow m \cdot h_1^{r_1} h_2^{r_2} \quad v \leftarrow (c_1 d_1^\alpha)^{r_1} \cdot (c_2 d_2^\alpha)^{r_2} \\ vk \leftarrow g^\alpha \quad e_1 \leftarrow u_1^\alpha \quad e_2 \leftarrow u_2^\alpha \quad e_3 \leftarrow u_3^\alpha \quad f_1 \leftarrow g^c \quad f_2 \leftarrow g^\psi \\ \sigma_1 \leftarrow \mathsf{FS.Sign}\,(sk_1, (r1, r2, r1 + r2, c, \psi)) \quad \sigma_2 \leftarrow \mathsf{FS.WASign}\,(sk_2, e) \end{array}$$

$\mathsf{BCS.Dec}(pk, sk, C)$. Parse sk and C as above. Assemble $vk_1 \leftarrow (\gamma, g, vk, v', d', g_1, g_2, g_3, g, g, u'_1, \ldots, u'_5, 5)$ and $vk_2 \leftarrow (\gamma, g, f_2, v', d', g, f_1, 1)$. Now, verify the relations:

$$\begin{array}{r} \{\hat{e}(vk, u_i) = \hat{e}(e_i, g)\}_{i \in [1,3]} \wedge \\ \mathsf{FS.Verify}\,(vk_1, (u_1, u_2, u_3, f_1, f_2), \sigma_1) = 1 \ \wedge \ \mathsf{FS.Verify}\,(vk_2, (e), \sigma_2) = 1\} \end{array} \quad (1)$$

If this check fails, output $\perp$. Otherwise, parse $sk = (x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3)$ and select $z \xleftarrow{R} \mathbb{Z}_p^*$. Compute the decryption m' as:

$$m' = e \cdot \frac{(u_1^{x_1} e_1^{y_1} \cdot u_2^{x_2} e_2^{y_2} \cdot u_3^{x_3} e_3^{y_3})^z}{u_1^{z_1} u_2^{z_2} u_3^{z_3} \cdot v^z} \quad (2)$$

Ciphertexts consist of approximately 25 elements of $\mathbb{G}$ and two element of $\mathbb{Z}_p^*$. While at first glance these ciphertexts may seem large, note that the scheme can be instantiated in asymmetric bilinear settings such as the MNT group of elliptic curves, where group elements can be represented in as little as 170 bits at the 80-bit security level. In this setting we are able to achieve a relatively ciphertext size of approximately 5100 bits. While this is large compared to RSA, a 640-byte per file overhead is quite reasonable for many practical applications. Also note that in our description the KG algorithm samples a unique set of bilinear group parameters γ for each key; however, it is perfectly acceptable for many keyholders to share the same group parameters.

The Blind Decryption Protocol. The blind decryption protocol $\mathsf{BlindDec}$ with respect to BCS is shown in Figure 1. The protocol requires a multiplicatively

homomorphic IND-CPA-secure encryption scheme, which we instantiate using the Linear Encryption scheme (LE) of Boneh *et al.* [7].[6]

The protocol employs the homomorphic property of LE to construct a a two-party implementation of the Dec algorithm, with ZKPoKs used to ensure that both the User and Decryptor's contributions are correctly formed. Note that for security reasons it is critical that the Decryptor *re-randomize* the ciphertext that it sends back to the User in its portion of the protocol. In the LE scheme this can be accomplished by multiplying a ciphertext with a fresh encryption of the identity element.

SECURITY. Let Π_{ZK} be a zero-knowledge (and, implicitly, witness indistinguishable) proof system secure under the Decision Linear assumption (possibly in the Common Reference String model). In the following theorems we will show that if the Decision Linear and Flexible Diffie-Hellman assumptions hold in $\mathbb{G}$ then $\mathsf{BCS} = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{BlindDec})$ implemented with Π_{ZK} is a secure blind decryption scheme in the sense of Definition 5. To accomplish this we must show that: (1) the algorithms $(\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ comprise an IND-CCA2-secure encryption scheme, (2) the BlindDec protocol is leak-free, and (3) BlindDec achieves ciphertext blindness.

Theorem 1. *If the Decision Linear and Flexible Diffie-Hellman assumptions hold in* $\mathbb{G}$*, then* (BCS.KG, BCS.Enc, BCS.Dec) *comprise an* IND-CCA2*-secure public-key encryption scheme secure in the standard model.*

Due to space concerns, we must leave a full proof of Theorem 1 to the full version of this work [27]. Here we will sketch the intuition behind the proof, which employs techniques from the Cramer-Shoup variant proposed by Shacham [48]. As in that scheme, our simulator knows the scheme's secret key, and can use it to answer decryption queries. The exceptions to this rule are certain queries related to the challenge ciphertext. Specifically, we must be careful with queries that are (*a*) "malformed", *i.e.,* the queried value $v \neq u_1^{x_1} e_1^{y_1} \cdot u_2^{x_2} e_2^{y_2} \cdot u_3^{x_3} e_3^{y_3}$, or that (*b*) embed the value vk^* from the challenge ciphertext.

Note that equation (2) of the Dec algorithm ensures that malformed ciphertexts decrypt to a random element of $\mathbb{G}$, so the first case is easily dealt with in our simulation. The adversary cannot maul the ciphertext due to the presence of the checksum v. Thus it remains to consider well-formed ciphertexts with $vk = vk^*$. We argue that the challenge ciphertext itself is the only ciphertext that will pass all of our checks.

Intuitively our simulation accomplishes this by setting $vk = g^{\alpha^*}$ as the public key of a strongly unforgeable OTS which is secure under the Flexible Diffie-Hellman assumption. In principle we use this key to sign the challenge ciphertext components $(u_1^*, u_2^*, u_3^*, e^*)$, which produces all of the remaining components of the ciphertext. When the adversary submits a decryption query with $vk = vk^*$

[6] In asymmetric bilinear groups where the Decisional Diffie-Hellman problem is hard, this can easily be replaced with Elgamal encryption, resulting in a significant efficiency improvement.

$\underline{\mathcal{U}(pk, C)}$ $\underline{\mathcal{D}(pk, sk)}$

1. Parse C as $(u_1,\ u_2,\ u_3,\ e,\ v,\ vk,\ e_1,\ e_2,\ e_3,\ f_1,\ f_2,\ \sigma_1,\ \sigma_2)$, and parse $pk = (\gamma,\ g,\ g_1,\ g_2,\ g_3,\ c_1,\ c_2,\ d_1,\ d_2,\ h_1,\ h_2,\ v',\ d',\ u'_1,\ \ldots,\ u'_5)$. Verify that C satisfies equation (1) of the Dec algorithm. If not, abort and output $\perp$.
2. Generate $(pk_{\mathcal{U}}, sk_{\mathcal{U}}) \leftarrow \mathsf{LE.KG}(\gamma)$ and select $\bar{z} \xleftarrow{R} \mathbb{Z}_p^*$. Compute: $\mathbf{c}_1 \leftarrow \mathsf{LE.Enc}(pk_{\mathcal{U}}, u_1^{\bar{z}})$, $\mathbf{c}_2 \leftarrow \mathsf{LE.Enc}(pk_{\mathcal{U}}, u_2^{\bar{z}})$, $\mathbf{c}_3 \leftarrow \mathsf{LE.Enc}(pk_{\mathcal{U}}, u_3^{\bar{z}})$, $\mathbf{c}_4 \leftarrow \mathsf{LE.Enc}(pk_{\mathcal{U}}, e_1^{\bar{z}})$, $\mathbf{c}_5 \leftarrow \mathsf{LE.Enc}(pk_{\mathcal{U}}, e_2^{\bar{z}})$, $\mathbf{c}_6 \leftarrow \mathsf{LE.Enc}(pk_{\mathcal{U}}, e_3^{\bar{z}})$, $\mathbf{c}_7 \leftarrow \mathsf{LE.Enc}(pk_{\mathcal{U}}, v^{\bar{z}})$ and set $vk_1 \leftarrow (\gamma, g, vk, v', d', g_1, g_2, g_3, g, g, u'_1, \ldots, u'_5, 5)$, $vk_2 \leftarrow (\gamma, g, f_2, v', d', g, f_1, 1)$
3. Send $pk_{\mathcal{U}}, \mathbf{c}_1, \ldots, \mathbf{c}_7$ to $\mathcal{D}$ and conduct the following PoK with $\mathcal{D}$:

$\mathsf{WIPoK}\{(u_1, u_2, u_3, v, vk, e_1, e_2, e_3, f_1, f_2, \sigma_1, \sigma_2, vk_1, vk_2, \bar{z}):$
$\mathbf{c}_1 = \mathsf{LE.Enc}(pk_{\mathcal{U}}, u_1^{\bar{z}}) \wedge \mathbf{c}_2 = \mathsf{LE.Enc}(pk_{\mathcal{U}}, u_2^{\bar{z}}) \wedge \mathbf{c}_3 = \mathsf{LE.Enc}(pk_{\mathcal{U}}, u_3^{\bar{z}}) \wedge$
$\mathbf{c}_4 = \mathsf{LE.Enc}(pk_{\mathcal{U}}, e_1^{\bar{z}}) \wedge \mathbf{c}_5 = \mathsf{LE.Enc}(pk_{\mathcal{U}}, e_2^{\bar{z}}) \wedge \mathbf{c}_6 = \mathsf{LE.Enc}(pk_{\mathcal{U}}, e_3^{\bar{z}}) \wedge$
$\mathbf{c}_7 = \mathsf{LE.Enc}(pk_{\mathcal{U}}, v^{\bar{z}}) \ \wedge\ \hat{e}(vk, u_i) = \hat{e}(e_i, g)\}_{i\in[1,3]} \wedge$
$\mathsf{FS.Verify}\,(vk_1, (u_1, u_2, u_3, f_1, f_2), \sigma_1) = 1 \ \wedge\ \mathsf{FS.Verify}\,(vk_2, (e), \sigma_2) = 1\}$

4. If the proof does not verify, abort.
5. Compute $\mathbf{c}' = \mathsf{LE.Enc}(pk_{\mathcal{U}}, 1)$, $\bar{z}' \xleftarrow{R} \mathbb{Z}_p^*$.
6. Using the homomorphic property of LE, compute: $\mathbf{c}'' \leftarrow \frac{(\mathbf{c}_1^{x_1}\mathbf{c}_4^{y_1}\cdot\mathbf{c}_2^{x_2}\mathbf{c}_5^{y_2}\cdot\mathbf{c}_3^{x_3}\mathbf{c}_6^{y_3})^{\bar{z}'}}{\mathbf{c}_1^{z_1}\mathbf{c}_2^{z_2}\mathbf{c}_3^{z_3}\cdot\mathbf{c}_7^{\bar{z}'}} \cdot \mathbf{c}'$.
7. Return $\mathbf{c}''$ and conduct the following proof:

$ZKPoK\{(x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3, \bar{z}', \mathbf{c}'):$
$\mathbf{c}' = \mathsf{LE.Enc}(pk, 1) \wedge$
$\mathbf{c}'' = \frac{(\mathbf{c}_1^{x_1}\mathbf{c}_4^{y_1}\cdot\mathbf{c}_2^{x_2}\mathbf{c}_5^{y_2}\cdot\mathbf{c}_3^{x_3}\mathbf{c}_6^{y_3})^{\bar{z}'}}{\mathbf{c}_1^{z_1}\mathbf{c}_2^{z_2}\mathbf{c}_3^{z_3}\cdot\mathbf{c}_7^{\bar{z}'}} \cdot \mathbf{c}'\}$

8. If the proof does not verify, abort and return $\perp$.
9. Compute $m' = e \cdot (\mathsf{LE.Dec}(sk, \mathbf{c}''))^{1/\bar{z}}$.

Output m'. Output nothing.

Fig. 1. The Blind Decryption protocol $\mathsf{BlindDec}(\mathcal{U}(pk, C), \mathcal{D}(pk, sk)) \rightarrow (m', \mathsf{nothing})$. For compactness of notation we represent the homomorphic operation on two LE ciphertexts $\mathbf{c}_1, \mathbf{c}_2$ using simple multiplicative notation ($\mathbf{c}_1\mathbf{c}_2$), and exponentiation by a scalar z as $\mathbf{c}_1^z$.

we can be assured that the query is identical to the challenge ciphertext, as any other result would require the adversary to forge the OTS.

It remains to separately argue that the signatures σ_1, σ_2 are unforgeable. This is non-trivial, since the OTS operates on messages of the form $m_1, \ldots, m_n \in \mathbb{Z}_p^*$. In a separate simulation we could select the elements u_1^*, u_2^*, u_3^* such the simulator knows their discrete logarithm base g. Unfortunately, even this is not sufficient, since our simulator cannot always know the discrete logarithm of the value e^* which is based on a message chosen by the adversary. The core intuition of our proof is to give two separate simulations: in one the signing key α^* is known and we can *simulate* the signature, producing a correctly-distributed (but not

unforgeable) signature over arbitrary group elements. In the second simulation the signing key is unknown: the simulator chooses $(u_1^*, u_2^*, u_3^*, e^*)$ at random such that it knows the discrete logarithm (base g) of each value. Although the resulting ciphertext does not encrypt either m_0 *or* m_1, an adversary is unable to detect this condition under the Decision Linear assumption.

Theorem 2. *If the Decision Linear assumption holds in* $\mathbb{G}$ *and* Π_{ZK} *is secure under the Decision Linear assumption, then the* BCS *protocol* BlindDec *is leak-free.*

We present a proof sketch of Theorem 2 in the full version of this work [27]. Intuitively this proof is quite simple: we show that for any real-world adversary $\mathcal{A}$ we can construct an ideal-world adversary $\mathcal{S}$ that, whenever $\mathcal{A}$ initiates the BlindDec protocol, operates as follows: (1) $\mathcal{S}$ uses the extractor for the PoK system to obtain $\mathcal{A}$'s requested ciphertext, (2) queries this result to the trusted decryption oracle, (3) re-blinds and returns the correctly formulated result to the adversary, simulating the necessary ZK proofs. We show that under the Decision Linear assumption no *p.p.t.* distinguisher can differentiate the output of $\mathcal{S}$ playing the Ideal-World game from the output of $\mathcal{A}$ in the Real-World game except with negligible probability.

Theorem 3. *If the Decision Linear assumption holds in* $\mathbb{G}$ *and* Π_{ZK} *is secure under the Decision Linear assumption, then the* BCS *protocol* BlindDec *satisfies the property of Ciphertext Blindness (*Blind*).*

We sketch a proof of Theorem 3 in the full version of this work [27]. Intuitively, we show that an adversarial Decryptor who distinguishes the User's exection of the blind decryption protocol on two distinct (and adversarially-chosen) ciphertexts C_0 and C_1 must imply a distinguisher for the witness indistinguishable proof system, or a CPA adversary against the LE encryption scheme.

4.1 Extensions

Tag-Based Encryption. Tag-Based Encryption (TBE) allows encryptors to apply a tag (label) to each ciphertext. This tag is used during the decryption process. The BCS construction is in fact natively based on a TBE scheme, but this functionality is lost as part of the TBE-to-PKE transform we use. In the full version of this work [27] we show that with some minor extensions it is possible to retain the scheme's full TBE functionality.

Selective-failure blindness. Camenisch *et al.* [15] propose a stronger definition of blindness (for signature schemes) that they refer to as "selective-failure" blindness. Intuitively, this definition captures the notion that an adversarial Decryptor might attempt to induce failures in the protocol (*e.g.,* by generating malformed ciphertexts) in order to deprive the User of privacy. Unfortunately our protocols do not natively achieve this definition because the Decryptor can create ciphertexts with an improperly formed check value v. Unfortunately, due to the nature of our scheme this check cannot be verified independently by the

user. One potential solution to this problem is to add to each ciphertext a non-interactive proof that v is correctly formed. Such a proof could be constructed using the Fiat-Shamir heuristic in the random oracle model, or using the Groth-Sahai system in the Common Reference String model. Note that this approach would not require any changes to the blind decryption protocol. We elaborate on this approach in the full version of this work [27].

5 Applications

Blind decryption has applications to a number of privacy-preserving protocols. Several applications have already been proposed in the literature, *e.g.,* [43, 24]. Below we propose two specific applications motivated by our construction.

Privacy-preserving Distributed Filesystems. Many organizations are responding to the difficulty of securing data in a distributed network, where storage locations can include semi-trusted file servers, desktop computers and mobile devices. An increasingly popular approach is to employ cryptographic access control to restrict and monitor file access in these environments. In this approach (*e.g.,* [1]), access control is performed by encrypting files at rest; authorized users contact a centralized server in order to decrypt them when necessary.

A concern with this approach is that the server gains a great deal of information regarding users' access patterns. In some cases, knowing *which* content a user is accessing may by itself leak confidential information. For example, the pattern of file accesses by executives during a corporate merger might have enormous financial value to an investor. While it is desirable to centralize access control, it may also be important to restrict this centralized party from learning which information is being managed. While these goals seems contradictory, Coull *et al.* [20] and Camenisch *et al.* [12] recently showed how to construct sophisticated access control mechanisms using anonymous credentials. In these protocols a server provides strong, and even *history-dependent* access control without ever learning user's access pattern. Our blind decryption protocols are amenable to integration with these access control techniques. In particular, by extending BCS to include *encryption tags* as in Section 4.1, data can be explicitly categorized and policies can be defined around these categories.

Oblivious Transfer with Public Contribution. In an adaptively-secure k-out-of-N Oblivious Transfer protocol ($\mathsf{OT}^N_{k\times 1}$) a Receiver obtains up to k items from a Sender's N-item database, without revealing to the Sender *which* messages were transferred. There has been much recent interest in $\mathsf{OT}^N_{k\times 1}$ [15, 28, 29, 33, 44, 20, 12], as it is particularly well suited for constructing privacy-preserving databases in which the user's query pattern is cryptographically protected (this is critical in *e.g.,* patent and medical databases).

For practical reasons, there are situations in which it is desirable to distribute the authorship of records, particularly when database updates are performed offline. Unfortunately, existing $\mathsf{OT}^N_{k\times 1}$ protocols seem fundamentally incapable of supporting message contributions by third parties without the explicit cooperation of the Sender. Our blind decryption constructions admit new $\mathsf{OT}^N_{k\times 1}$

protocols that support *public contribution*. Intuitively, contributors simply encrypt their messages using the Enc algorithm under the Sender's public key and send the resulting ciphertexts directly to the Receiver. The Receiver can then obtain up to k decryptions by running BlindDec with the Sender. Proving this intuitive protocol secure under a strong simulation-based definition [15, 28] requires some additional components that are easily achieved using the techniques available to us.

Acknowledgements. The author would like to thank Susan Hohenberger for her helpful comments.

References

1. Eruces Tricryption, `http://www.eruces.com/index.php`
2. Adida, B., Hohenberger, S., Rivest, R.L.: Ad-hoc group signatures from hijacked keypairs. In: DIMACS Workshop on Theft in E-Commerce (preliminary version) (April 2005)
3. Ateniese, G., Camenisch, J., de Medeiros, B.: Untraceable RFID tags via insubvertible encryption. In: CCS 2005, pp. 92–101. ACM Press, New York (2005)
4. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and Noninteractive Anonymous Credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (2008)
5. Bellare, M., Rogaway, P.: Optimal asymmetric encryption padding — how to encrypt with rsa. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 92–111. Springer, Heidelberg (1995)
6. Boneh, D., Boyen, X.: Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
7. Boneh, D., Boyen, X., Shacham, H.: Short Group Signatures. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 41–55. Springer, Heidelberg (2004)
8. Boneh, D., Katz, J.: Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 87–103. Springer, Heidelberg (2005)
9. Boneh, D., Sahai, A., Waters, B.: Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
10. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: CCS 2005, pp. 320–329. ACM Press, New York (2005)
11. Brassard, G., Crépeau, C., Robert, J.M.: All-or-Nothing Disclosure of Secrets. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 234–238. Springer, Heidelberg (1987)
12. Camenisch, J., Dubovitskaya, M., Neven, G.: Oblivious transfer with access control. In: CCS 2009, pp. 131–140. ACM, New York (2009)
13. Camenisch, J., Lysyanskaya, A.: Signature Schemes and Anonymous Credentials from Bilinear Maps. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 56–72. Springer, Heidelberg (2004)
14. Camenisch, J., Michels, M.: Proving in zero-knowledge that a number n is the product of two safe primes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 107–122. Springer, Heidelberg (1999)

15. Camenisch, J., Neven, G., Shelat, A.: Simulatable Adaptive Oblivious Transfer. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 573–590. Springer, Heidelberg (2007)
16. Camenisch, J., Stadler, M.: Efficient Group Signature Schemes for Large Groups. In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 410–424. Springer, Heidelberg (1997)
17. Camenisch, J.L.: Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem. PhD thesis, ETH Zürich (1998)
18. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg (2003)
19. Chaum, D.: Blind signatures for untraceable payments. In: CRYPTO 1982, pp. 199–203. Plenum Press, New York (1982)
20. Coull, S., Green, M., Hohenberger, S.: Controlling Access to an Oblivious Database Using Stateful Anonymous Credentials. In: Jarecki, S., Tsudik, G. (eds.) PKC 2009. LNCS, vol. 5443, pp. 501–520. Springer, Heidelberg (2009)
21. Cramer, R., Damgård, I., Schoenmakers, B.: Proof of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (1994)
22. Cramer, R., Shoup, V.: A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 13–25. Springer, Heidelberg (1998)
23. Damgård, I., Mambo, M., Okamoto, E.: Further study on the transformability of digital signatures and the blind decryption. In: SCIS 1997-33B (1997)
24. Dodis, Y., Halevi, S., Rabin, T.: A Cryptographic Solution to a Game Theoretic Problem. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 112–130. Springer, Heidelberg (2000)
25. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
26. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or a completeness theorem for protocols with honest majority. In: STOC 1987, pp. 218–229 (1987)
27. Green, M.: Secure blind decryption (full version) (December 2010), `http://eprint.iacr.org`, `http://spar.isi.jhu.edu/~mgreen/SecureBlindDecryption.pdf`
28. Green, M., Hohenberger, S.: Blind Identity-Based Encryption and Simulatable Oblivious Transfer. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 265–282. Springer, Heidelberg (2007)
29. Green, M., Hohenberger, S.: Universally Composable Adaptive Oblivious Transfer. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 179–197. Springer, Heidelberg (2008)
30. Green, M., Hohenberger, S.: Practical adaptive oblivious transfer from a simple assumption. In: TCC 2011. LNCS, vol. 6597. Springer, Heidelberg (to appear), `http://eprint.iacr.org/2010/109`
31. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
32. Hohenberger, S., Rothblum, G.N., Shelat, A., Vaikuntanathan, V.: Securely Obfuscating Re-encryption. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 233–252. Springer, Heidelberg (2007)

33. Jarecki, S., Liu, X.: Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 577–594. Springer, Heidelberg (2009)
34. Kilian, J.: Founding cryptography on oblivious transfer. In: STOC 1988, pp. 20–31 (1988)
35. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
36. Kunz-Jacques, S., Pointcheval, D.: About the Security of MTI/C0 and MQV. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 156–172. Springer, Heidelberg (2006)
37. Kurosawa, K., Nojima, R.: Simple Adaptive Oblivious Transfer without Random Oracle. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 334–346. Springer, Heidelberg (2009)
38. Laguillaumie, F., Paillier, P., Vergnaud, D.: Universally Convertible Directed Signatures. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 682–701. Springer, Heidelberg (2005)
39. Libert, B., Vergnaud, D.: Multi-use unidirectional proxy re-signatures. In: CCS 2008, pp. 511–520. ACM, New York (2008)
40. MacKenzie, P.D., Reiter, M.K., Yang, K.: Alternatives to non-malleability: Definitions, constructions, and applications (extended abstract). In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 171–190. Springer, Heidelberg (2004)
41. Mambo, M., Sakurai, K., Okamoto, E.: How to utilize the transformability of digital signatures for solving the oracle problem. In: Kim, K.-c., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 322–333. Springer, Heidelberg (1996)
42. Ogata, W., Le Trieu, P.: Blind HIBE and its application to blind decryption. In: SCIS, pp. 4D1–2 (2008)
43. Radia, P.: The ephemerizer: Making data disappear. Journal of Information System Security 1(1), 51–68 (2005)
44. Rial, A., Kohlweiss, M., Preneel, B.: Universally Composable Adaptive Priced Oblivious Transfer. In: Shacham, H., Waters, B. (eds.) Pairing 2009. LNCS, vol. 5671, pp. 231–247. Springer, Heidelberg (2009)
45. Sakurai, K., Yamane, Y.: Blind decoding, blind undeniable signatures, and their applications to privacy protection. In: Anderson, R. (ed.) IH 1996. LNCS, vol. 1174, pp. 257–264. Springer, Heidelberg (1996)
46. Schnorr, C.-P.: Efficient signature generation for smart cards. Journal of Cryptology 4(3), 239–252 (1991)
47. Schnorr, C.-P., Jakobsson, M.: Security of Signed ElGamal Encryption. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 73–89. Springer, Heidelberg (2000)
48. Shacham, H.: A Cramer-Shoup encryption scheme from the linear assumption and from progressively weaker linear variants. Cryptology ePrint Archive, Report 2007/074 (2007), `http://eprint.iacr.org/`
49. Teague, V.: Selecting Correlated Random Actions. In: Juels, A. (ed.) FC 2004. LNCS, vol. 3110, pp. 181–195. Springer, Heidelberg (2004)
50. Yao, A.: How to generate and exchange secrets. In: FOCS 1986, pp. 162–167 (1986)