Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups

David Mandell Freeman[⋆]

Stanford University, USA
dfreeman@cs.stanford.edu

Abstract. We develop an abstract framework that encompasses the key properties of bilinear groups of composite order that are required to construct secure pairing-based cryptosystems, and we show how to use prime-order elliptic curve groups to construct bilinear groups with the same properties. In particular, we define a generalized version of the subgroup decision problem and give explicit constructions of bilinear groups in which the generalized subgroup decision assumption follows from the decision Diffie-Hellman assumption, the decision linear assumption, and/or related assumptions in prime-order groups.

We apply our framework and our prime-order group constructions to create more efficient versions of cryptosystems that originally required composite-order groups. Specifically, we consider the Boneh-Goh-Nissim encryption scheme, the Boneh-Sahai-Waters traitor tracing system, and the Katz-Sahai-Waters attribute-based encryption scheme. We give a security theorem for the prime-order group instantiation of each system, using assumptions of comparable complexity to those used in the composite-order setting. Our conversion of the last two systems to prime-order groups answers a problem posed by Groth and Sahai.

Keywords: pairing-based cryptography, composite-order groups, cryptographic hardness assumptions.

1 Introduction

Bilinear groups of composite order are a tool that has been used in the last few years to solve many problems in cryptography. The concept was introduced by Boneh, Goh, and Nissim [3], who applied the technique to the problems of private information retrieval, online voting, and universally verifiable computation. Subsequent authors have built on their work to create protocols such as non-interactive zero-knowledge proofs [13, 14], ring and group signatures [6, 20], attribute-based encryption [5, 16], traitor tracing schemes [4], and hierarchical IBE [17, 21].

[⋆] Research supported by a National Science Foundation International Research Fellowship, with additional support from the Office of Multidisciplinary Activities in the NSF Directorate for Mathematical and Physical Sciences.

Bilinear groups of composite order are pairs of abelian groups $(\mathbb{G}, \mathbb{G}_t)$, each of composite order $N = pq$, equipped with a nondegenerate bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$. Cryptosystems using bilinear groups of composite order usually base their security on variants of the *subgroup decision assumption*. Informally, this assumption says that given an element $g \in \mathbb{G}$, there is no efficient algorithm to determine whether g has order p. In particular, the assumption implies that it is infeasible to factor the group order N.

While the subgroup decision assumption is a useful tool for constructing secure protocols, it presents significant obstacles to implementing these protocols in practice. The only known instantiations of composite-order bilinear groups use elliptic curves (or more generally, abelian varieties) over finite fields. Since the elliptic curve group order n must be infeasible to factor, it must be at least (say) 1024 bits. On the other hand, the size of a prime-order elliptic curve group that provides an equivalent level of security is 160 bits [1]. As a result, group operations and especially pairing computations are prohibitively slow on composite-order curves: a Tate pairing on a 1024-bit composite-order elliptic curve is roughly 50 times slower than the same pairing on a comparable prime-order curve [18], and this performance gap will only get worse at higher security levels.

In short, requiring that the group order be infeasible to factor negates the principal advantage of elliptic curve cryptography over factoring-based systems, namely, that there is no known subexponential-time algorithm for computing discrete logarithms on an elliptic curve, while there is such an algorithm for factoring. Thus for efficient implementations we seek versions of protocols that use only prime-order elliptic curve groups. Developing these protocols is the main goal of this paper. In particular, we do the following:

- We **develop an abstract framework** that encompasses the key properties of bilinear groups of composite order, and we show how to use prime-order elliptic curves to construct bilinear groups with the same properties.
- We apply our framework and our prime-order construction to **create more efficient versions of cryptosystems** that originally used composite-order groups. Specifically, we consider:
    1. The Boneh-Goh-Nissim encryption scheme [3],
    2. The Boneh-Sahai-Waters traitor tracing system [4], and
    3. The Katz-Sahai-Waters attribute-based encryption scheme [16].

Our conversion of the last two systems to prime-order groups answers a problem posed by Groth and Sahai [14, Section 9], who themselves implicitly use our framework to construct non-interactive proof systems using either composite-order or prime-order groups.

Outline and Summary of Results. The starting point for our abstract framework is the fact that the subgroup decision assumption defined by Boneh, Goh, and Nissim depends only on the existence of a group G for which it is infeasible to determine if an element $g \in G$ lies in a given proper subgroup G_1 of G. This observation gives us a more general subgroup decision assumption in the language of abstract groups (see Section 2).

Our construction using prime-order groups is based on the observation, used implicitly by Cramer and Shoup [7] and articulated explicitly by Gjøsteen [12], that the decision Diffie-Hellman (DDH) assumption is a generalized subgroup decision assumption. Specifically, suppose we are given a cyclic group $\mathbb{G}$ and elements $g, g^a, g^b, g^c \in \mathbb{G}$. Then the DDH assumption for $\mathbb{G}$ says exactly that it is infeasible to determine whether (g^b, g^c) is in the cyclic subgroup of $\mathbb{G} \times \mathbb{G}$ generated by (g, g^a). Thus any protocol that requires two groups $G_1 \subset G$ in which the generalized subgroup decision assumption holds can be instantiated using $G = \mathbb{G}\times\mathbb{G}$ and $G_1 = \langle (g_1, g_2) \rangle$, where $\mathbb{G}$ is a cyclic group in which the DDH assumption holds and g_1, g_2 are random elements of $\mathbb{G}$.

More generally, we can use $G = \mathbb{G}^n$ for any $n \geq 2$ and let G_1 be a rank-k subgroup for any $1 \leq k < n$. In this case the subgroup decision assumption in G follows from the *k-Linear assumption* in $\mathbb{G}$, which generalizes the DDH assumption. In particular, the 1-Linear assumption is DDH, while the 2-Linear assumption is the *decision linear assumption.* This more general construction makes explicit a relationship noticed by several previous authors (e.g., [14, 21]), namely, that functionality that can be achieved in composite-order groups under the subgroup decision assumption can also be achieved in prime-order groups under either the DDH or the decision linear assumption.

If the group $\mathbb{G}$ is equipped with a pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$, then applying $\hat{e}$ componentwise defines a pairing on $G = \mathbb{G}^n$. However, such a "symmetric" pairing (which only exists on supersingular elliptic curves) can be used to solve DDH in $\mathbb{G}$, so in this case our DDH-based construction is not secure. To get around this problem we use the fact that on ordinary (i.e., non-supersingular) elliptic curves there are two distinguished subgroups, denoted $\mathbb{G}_1$ and $\mathbb{G}_2$, in which DDH is believed to be infeasible for sufficiently large group orders. We can thus apply our construction twice to produce groups $G = \mathbb{G}_1^n$, $H = \mathbb{G}_2^n$, $G_t = \mathbb{G}_t^m$ (for some m), and an "asymmetric" pairing $e : G\times H \to G_t$. If the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$, then the subgroup decision assumption holds in G and H. (If using the d-Linear assumption with $d \geq 2$, we can remain in the symmetric setting.)

While the security of composite-order group protocols depends on (variants of) the subgroup decision assumption, the correctness of these protocols depends on the groups having certain additional properties. In some cases, the groups G, H, G_t must be equipped with projection maps π_1, π_2, π_t that map them onto proper subgroups and commute with the pairing. In other cases, the groups must decompose into subgroups $G \cong \prod G_i$ and $H \cong \prod H_i$ such that the pairing restricted to $G_i \times H_j$ is trivial whenever $i \neq j$. In Section 3 we define these properties in our abstract framework and show how to instantiate them in the product groups $\mathbb{G}_1^n$, $\mathbb{G}_2^n$.

Sections 2 and 3 give us the framework and the tools necessary to convert composite-order group protocols to prime-order groups. Section 4 analyzes the efficiency gains realized in terms of the number of bits needed to represent group elements. For example, at a security level equivalent to 80-bit AES, ciphertexts in the Boneh-Goh-Nissim cryptosystem can be up to three times smaller when instantiated using our prime-order construction than in the original composite-order system. At the 256-bit security level the improvement can be as large as a factor of 12.

In Section 5 we describe in detail the conversion procedure for the Boneh-Goh-Nissim cryptosystem, and in Section 6 we sketch the same for the Boneh-Sahai-Waters traitor tracing system and the Katz-Sahai-Waters attribute-based encryption scheme. (Details are in the full version of this paper [10].) In each case we describe the scheme in our general framework and convert the assumptions used in the security proofs to our more general setting. We then consider the system instantiated with our prime-order group construction and give security theorems in this setting. If the original system is secure under a simple assumption (e.g., subgroup decision) then the converted scheme is also secure under a simple assumption (e.g., DDH); if the original system uses a complex assumption (as in the Katz-Sahai-Waters system) then the corresponding assumption in prime-order groups is also complex.

We note that our conversion process is not “black box”: the security proof for each system must be analyzed to determine whether it carries over to our more general setting. For example, the recent IBE scheme of Lewko and Waters [17] uses explicitly in its security proof the fact that the group G has two subgroups of relatively prime order, and thus our techniques do not apply. However, we do expect that our framework can be used to convert to prime-order groups other cryptosystems originally built using composite-order groups.

2 Subgroup Decision Problems

The problem of determining whether a given element g of a finite group G lies in a specified proper subgroup G_1 was used as a hardness assumption for constructing cryptosystems long before Boneh, Goh, and Nissim defined their "subgroup decision problem." Gjøsteen [12] has undertaken an extensive survey of such problems, which he calls "subgroup membership problems." For example, the *quadratic residuosity problem* is a subgroup membership problem: if we let $N = pq$ be a product of two distinct primes and define the group G to be the group of elements of $\mathbb{Z}_N^*$ with Jacobi symbol 1, the problem is to determine whether a given element in G lies in the subgroup of squares in G.

Boneh, Goh, and Nissim [3] defined their problem for pairs of groups $(\mathbb{G}, \mathbb{G}_t)$ of composite order $N = pq$ for which there exists a nondegenerate bilinear map, or "pairing," $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$. The problem is to determine whether a given element $g \in \mathbb{G}$ is in the subgroup of order p. Note that if g' generates $\mathbb{G}$, then $e(g, g')$ is a challenge element for the same problem in $\mathbb{G}_t$; thus if the subgroup decision problem is infeasible in $\mathbb{G}$ then it is in $\mathbb{G}_t$ as well.

Our general notion of a subgroup decision problem extends Gjøsteen's work to the bilinear setting. We begin by defining an object that generates the groups we will work with. We assume that the two groups input to the pairing are not identical; this is known as an *asymmetric* pairing.

Definition 2.1. A *bilinear group generator* is an algorithm $\mathcal{G}$ that takes as input a security parameter λ and outputs a description of five abelian groups G, G_1, H, H_1, G_t, with $G_1 \subset G$ and $H_1 \subset H$. We assume that this description permits efficient (i.e., polynomial-time in λ) group operations and random sampling in each group. The algorithm also outputs an efficiently computable map (or "pairing") $e : G \times H \to G_t$ that is

- Bilinear: $e(g_1 g_2, h_1 h_2) = e(g_1, h_1)e(g_1, h_2)e(g_2, h_1)e(g_2, h_2)$ for all g_1, $g_2 \in G$, $h_1, h_2 \in H$; and
- Nondegenerate: for any $g \in G$, if $e(g, h) = 1$ for all $h \in H$, then $g = 1$ (and similarly with G, H reversed).

Our generalized subgroup decision assumption says that it is infeasible to distinguish an element in G_1 from a random element of G, and similarly for H. More precisely, we have the following definition. (The notation $x \overset{\text{R}}{\leftarrow} X$ means x is chosen uniformly at random from the set X.)

Definition 2.2. Let $\mathcal{G}$ be a bilinear group generator. We define the following distribution:

$$\mathbb{G} = (G, G_1, H, H_1, G_t, e) \overset{\text{R}}{\leftarrow} \mathcal{G}(\lambda),\ T_0 \overset{\text{R}}{\leftarrow} G,\ T_1 \overset{\text{R}}{\leftarrow} G_1.$$

We define the *advantage* of an algorithm $\mathcal{A}$ in solving the *subgroup decision problem on the left* to be

$$\text{SDP}_{\text{L}}\text{-Adv}[\mathcal{A}, \mathcal{G}] = \Big| \Pr[\mathcal{A}(\mathbb{G}, T_0) = 1] - \Pr[\mathcal{A}(\mathbb{G}, T_1) = 1] \Big|.$$

We say that $\mathcal{G}$ *satisfies the subgroup decision assumption on the left* if $\text{SDP}_{\text{L}}\text{-Adv}[\mathcal{A}, \mathcal{G}](\lambda)$ is a negligible function of λ for any polynomial-time algorithm $\mathcal{A}$. We define the *subgroup decision problem/assumption on the right* and $\text{SDP}_{\text{R}}\text{-Adv}[\mathcal{A}, \mathcal{G}]$ analogously, with $T_0 \overset{\text{R}}{\leftarrow} H$ and $T_1 \overset{\text{R}}{\leftarrow} H_1$. We say $\mathcal{G}$ *satisfies the subgroup decision assumption* if it satisfies both the left and right assumptions.

Example 2.3 ([3, Section 2.1]). Boneh, Goh, and Nissim construct a bilinear group generator $\mathcal{G}_{BGN}$ using supersingular elliptic curves of composite order. Let $\mathcal{E}(\lambda)$ be an algorithm that outputs a product $N = p_1 p_2$ of two distinct primes greater than 2^λ, a prime $q \equiv -1 \pmod{N}$, and a supersingular elliptic curve E over the finite field $\mathbb{F}_q$. Then $\#E(\mathbb{F}_q)$ is divisible by N, and we can construct $\mathcal{G}_{BGN}(\lambda)$ by running $\mathcal{E}(\lambda)$ and setting the output as follows:

- $G = H$ is the order-N subgroup of $E(\mathbb{F}_q)$;
- $G_1 = H_1$ is the order-p_1 subgroup of $E(\mathbb{F}_q)$;
- G_t is the order-N subgroup of $\mathbb{F}_{q^2}^*$; and
- $e : G \times G \to G_t$ is the modified N-Tate pairing on E [8, Sect. 2.1].

Each group is described by giving a generator.

It is believed that $\mathcal{G}_{BGN}$ satisfies the subgroup decision assumption when N is infeasible to factor. The construction can be extended to produce a group G whose order is a product of three or more primes, and the subgroup decision assumption is believed to hold in any nontrivial proper subgroup G_1 of G. Using the generic group analysis of Katz, Sahai, and Waters [16, Theorem A.2], one can show that any efficient generic algorithm to solve the subgroup decision problem for $\mathcal{G}_{BGN}$ can be used construct an efficient algorithm to factor N.

2.1 Product Groups, DDH, and d-Linear Assumptions

The primary motivation for our abstraction of composite-order group protocols is the observation that the decision Diffie-Hellman problem is also a subgroup decision problem [12, Section 4.5].

Let $\mathbb{G}$ be a finite cyclic group, and let $T = (g, g^a, g^b, g^c)$ be a 4-tuple of elements in $\mathbb{G}$. The *decision Diffie-Hellman (DDH) problem* is to determine whether $c \equiv ab \pmod{|g|}$; if this is infeasible then we say that $\mathbb{G}$ satisfies the *decision Diffie-Hellman assumption.* Now suppose we are given a DDH challenge T. Define G to be $\mathbb{G} \times \mathbb{G}$ and G_1 to be the cyclic subgroup of G generated by (g, g^a). Then the element $(g^b, g^c) \in G$ is in G_1 if and only if $c \equiv ab \pmod{|g|}$ — so solving the subgroup decision problem for $G_1 \subset G$ is exactly equivalent to solving DDH in $\mathbb{G}$.

Now we consider the same construction in the bilinear setting: let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$ be finite cyclic groups, and let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t$ be a nondegenerate bilinear map. Then we can define $G = \mathbb{G}_1^2$, $H = \mathbb{G}_2^2$, and $G_t = \mathbb{G}_t^2$, and choose random elements of G and H to generate G_1 and H_1 respectively. We can define a nondegenerate pairing $e : G \times H \to \mathbb{G}_t$ by taking any invertible matrix $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right) \in \mathrm{Mat}_2(\mathbb{F}_p)$ and setting

$$e((g_1, g_2), (h_1, h_2)) := e(g_1, h_1)^a e(g_1, h_2)^b e(g_2, h_1)^c e(g_2, h_2)^d.$$

We can define a pairing mapping to $G_t = \mathbb{G}_t^m$ by choosing different coefficients a, b, c, d to define each component of the output. With this setup, if the DDH assumption holds in $\mathbb{G}_1$ and $\mathbb{G}_2$, then the subgroup decision assumption holds for $G_1 \subset G$ and $H_1 \subset H$.

More generally, we consider a bilinear group generator $\mathcal{G}_k^n$ that produces two groups $G = \mathbb{G}_1^n$ and $H = \mathbb{G}_2^n$ and random rank-k subgroups $G_1 \subset G$ and $H_1 \subset H$. In this situation the natural analogue of the DDH problem is the *k-Linear problem*, introduced by Hofheinz and Kiltz [15] and Shacham [19]. The 1-Linear problem is simply DDH, while the 2-Linear problem is called the *decision linear problem* and was originally proposed by Boneh, Boyen, and Shacham [2] as a reasonable analogue for DDH in a group with a bilinear map.

The following definition and theorem formalize the relationship between subgroup decision problems and d-Linear problems. We will use the following notation: if we have a group $\mathbb{G}$ of order p, an element $g \in \mathbb{G}$, and a vector $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{F}_p^n$, then we define $g^{\vec{x}} := (g^{x_1}, \ldots, g^{x_n}) \in \mathbb{G}^n$.

Definition 2.4. A bilinear group generator $\mathcal{P}$ is *prime-order* if the groups G, G_1, H, H_1, G_t all have prime order $p > 2^\lambda$. Then we have $G = G_1$ and $H = H_1$, and we denote the three distinct groups by $\mathbb{G}_1 = G$, $\mathbb{G}_2 = H$, and $\mathbb{G}_t = G_t$. We let $\hat{\mathbb{G}}$ denote the output $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, e)$ of $\mathcal{P}(\lambda)$.

Let $d \geq 1$ be an integer. If $\mathcal{A}$ is an algorithm that takes as input $2d + 2$ elements of $\mathbb{G}_1$, we define the *advantage* of $\mathcal{A}$ in solving the d-

Linear problem in $\mathbb{G}_1$, denoted d-$\mathrm{Lin}_{\mathbb{G}_1}$-$\mathrm{Adv}[\mathcal{A},\mathcal{P}]$, to be

$$\Bigg| \Pr\left[\mathcal{A}(\hat{\mathbb{G}}, g_1,\dots,g_d, g_1^{r_1},\dots,g_d^{r_d}, h, h^{r_1+\cdots+r_d}) = 1 : \substack{\hat{\mathbb{G}} \xleftarrow{R} \mathcal{P},\ g_1,\dots,g_d \xleftarrow{R} \mathbb{G}_1,\\ r_1,\dots,r_d \xleftarrow{R} \mathbb{F}_p}\right]$$
$$- \Pr\left[\mathcal{A}(\hat{\mathbb{G}}, g_1,\dots,g_d, g_1^{r_1},\dots,g_d^{r_d}, h, h^{s}) = 1 : \substack{\hat{\mathbb{G}} \xleftarrow{R} \mathcal{P},\ g_1,\dots,g_d \xleftarrow{R} \mathbb{G}_1,\\ r_1,\dots,r_d,s \xleftarrow{R} \mathbb{F}_p}\right] \Bigg|,$$

and similarly for d-$\mathrm{Lin}_{\mathbb{G}_2}$-$\mathrm{Adv}[\mathcal{A},\mathcal{P}]$. We say that $\mathcal{G}$ *satisfies the d-Linear assumption in* $\mathbb{G}_1$ if d-$\mathrm{Lin}_{\mathbb{G}_1}$-$\mathrm{Adv}[\mathcal{A},\mathcal{G}](\lambda)$ is a negligible function of λ for any polynomial-time algorithm $\mathcal{A}$ (and similarly for $\mathbb{G}_2$). The *decision Diffie-Hellman (DDH) assumption* is the 1-Linear assumption. The *decision linear assumption* is the 2-Linear assumption.

Some previous authors (e.g., [14]) have called the assumption that DDH is infeasible in both $\mathbb{G}_1$ and $\mathbb{G}_2$ the *symmetric external Diffie-Hellman assumption*, or SXDH. For clarity in our arguments, we prefer to call the problems DDH in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.

Theorem 2.5. *Let $\mathcal{P}$ be a prime-order bilinear group generator. For integers n,k with $n \geq 2$ and $1 \leq k < n$, define $\mathcal{G}_k^n$ to be a bilinear group generator that on input λ does the following:*

1. *Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, \hat{e}) \xleftarrow{R} \mathcal{P}(\lambda)$.*
2. *Let $G = \mathbb{G}_1^n$, $H = \mathbb{G}_2^n$, $G_t = \mathbb{G}_t^m$ for some m.*
3. *Choose generators $g \xleftarrow{R} \mathbb{G}_1$, $h \xleftarrow{R} \mathbb{G}_2$.*
4. *Choose random $\vec{x}_i, \vec{y}_i \xleftarrow{R} \mathbb{F}_p^n$ for $i = 1,\dots,k$, such that the sets $\{\vec{x}_i\}$ and $\{\vec{y}_i\}$ are each linearly independent.*
5. *Let G_1 be the subgroup of G generated by $\{g^{\vec{x}_1},\dots,g^{\vec{x}_k}\}$ and H_1 be the subgroup of H generated by $\{h^{\vec{y}_1},\dots,h^{\vec{y}_k}\}$*
6. *Choose nonzero $n \times n$ matrices $A_\ell = (a_{ij}^{(\ell)})$ for $\ell = 1,\dots,m$.*
7. *Define $e : G \times H \to G_t$ by $e((g_1,\dots,g_n),(h_1,\dots,h_n))_\ell := \prod e(g_i,h_j)^{a_{ij}^{(\ell)}}$.*
8. *Output the tuple $\Gamma_k^n = (G, G_1, H, H_1, G_t, e)$.*

If $\mathcal{P}$ satisfies the k-Linear assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{G}_k^n$ satisfies the subgroup decision assumption. Specifically, for any adversary $\mathcal{A}$ that solves the subgroup decision problem on the left for $\mathcal{G}_k^n$, there exists an adversary $\mathcal{B}$ that solves the k-Linear problem in $\mathbb{G}_1$ for $\mathcal{P}$, with

$$\mathrm{SDP_L}\text{-}\mathrm{Adv}[\mathcal{A},\mathcal{G}_k^n] \leq (n-k) \cdot k\text{-}\mathrm{Lin}_{\mathbb{G}_1}\text{-}\mathrm{Adv}[\mathcal{B},\mathcal{P}].$$

An analogous statement holds for $\mathcal{A}$ solving the subgroup decision problem on the right for $\mathcal{G}_k^n$ and $\mathcal{B}$ solving the k-Linear problem in $\mathbb{G}_2$ for $\mathcal{P}$.

Proof sketch. We sketch the proof for $n = k + 1$; the general case is proved in the full paper [10]. Let $(\hat{\mathbb{G}}, u_1, \ldots, u_k, v_1, \ldots, v_k, y, z)$ be a k-Linear challenge in $\mathbb{G}_1$. Let $\vec{x}_i = (x_{i,1}, \ldots, x_{i,n})$ be the vectors chosen in Step (4) above. Choose $\vec{b} \xleftarrow{\text{R}} \mathbb{F}_p^k$, and let G_1 be the subgroup generated by

$$\left\{(u_i^{x_{i,1}}, \ldots, u_i^{x_{i,k}}, v_i^{1/b_i})\right\}_{i=1}^{k}.$$

Now consider $T = (y^{\sum b_i x_{i,1}}, \ldots y^{\sum b_i x_{i,k}}, z) \in \mathbb{G}_1^n$, where each sum in the exponent runs over $i = 1$ to k. Write $v_i = u_i^{r_i}$, $z = y^c$. If $c = \sum_i r_i \pmod{p}$ then T is uniformly distributed in G_1, while if c is random then T is uniformly distributed in $\mathbb{G}_1^n$. It follows that any algorithm that has advantage ε in solving the subgroup decision problem for $\mathcal{G}_k^n$ can solve the k-Linear problem in $\mathbb{G}_1$ with advantage at least ε. $\square$

Since the d-Linear assumption implies the $(d+1)$-Linear assumption for all $d \geq 1$ [15, Lemma 3], if $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{G}_k^n$ satisfies the subgroup decision assumption for any $n \geq 1$ and $1 \leq k < n$. The converse holds when $k = 1$; the proof is in the full paper.

If we view all of the groups in the above construction as $\mathbb{F}_p$-vector spaces, then we see that the subgroup decision problem is a decisional version of the *vector decomposition problem* [22, 11], in which the adversary is given a decomposition $G \cong G_1 \times G_2$ and an element $x \in G$ and asked to find $y \in G_1$ and $z \in G_2$ such that $x = yz$.

The nondegeneracy of the pairing e defined on $\mathcal{G}_k^n$ will depend on the matrices A_ℓ and must be checked in each case. However, if $m = 1$ then e is nondegenerate if and only if A_1 is invertible.

3 Pairings on Product Groups

In our construction of the bilinear group generator $\mathcal{G}_k^n$ from the prime-order bilinear group generator $\mathcal{P}$, we took the pairing e on the product groups to be any nontrivial linear combination of the componentwise pairings on the underlying prime-order group. However, the correctness proofs for protocols built in composite-order groups all use the fact that the pairings have some extra structure that arbitrary linear combinations are unlikely to have. We now investigate this structure further and determine how to replicate it in our product group context.

Projecting Pairings. The cryptosystem of Boneh, Goh, and Nissim works by taking elements $g \in G$ and $h \in G_1$ and encrypting a message m as $C = g^m h^r$, where r is random. The h term is a "blinding term" used to hide the part of the ciphertext that contains the message. Decryption is achieved by "projecting" the ciphertext away from the blinding term and taking a discrete logarithm to recover m. Specifically, when g has order $N = p_1 p_2$ and h has order p_1, the decryption can be achieved by first computing C^{p_1} to remove the h term, and then taking the discrete logarithm to the base g^{p_1} to recover m. The functionality of the cryptosystem requires that we can do this procedure either before or after the pairing; i.e., that we can construct and remove blinding terms in G_t. The following definition incorporates this concept into our abstract framework.

Definition 3.1. Let $\mathcal{G}$ be a bilinear group generator (Def. 2.1). We say $\mathcal{G}$ is *projecting* if it also outputs a group $G_t' \subset G_t$ and group homomorphisms π_1, π_2, π_t mapping G, H, G_t to themselves, respectively, such that

1. G_1, H_1, G_t' are contained in the kernels of π_1, π_2, π_t, respectively, and
2. $e(\pi_1(g), \pi_2(h)) = \pi_t(e(g,h))$ for all $g \in G$, $h \in H$.

Example 3.2. The bilinear group generator $\mathcal{G}_{BGN}$ of Example 2.3 is projecting: we let G_t' be the order-p_1 subgroup of G_t, let $\pi_1 = \pi_2$ be exponentiation by p_1, and let π_t be exponentiation by $(p_1)^2$.

Given a prime-order bilinear group generator $\mathcal{P}$, we wish to modify the bilinear group generator $\mathcal{G}_k^n$ constructed in Theorem 2.5 so it is projecting. To do so, we interpret the generation of G_1 and H_1 in terms of matrix actions, and we define the pairing e using a *tensor product* of matrices.

We begin by defining the projection maps π_1 and π_2. Let $G = \mathbb{G}_1^n$ and let g be a generator of $\mathbb{G}_1$. For $i = 1, \ldots, n$, let $\vec{e}_i$ be the unit vector with a 1 in the ith place and zeroes elsewhere. To construct the projection map π_1, we first observe that if G_1' is the subgroup of G generated by $g^{\vec{e}_1}, \ldots, g^{\vec{e}_k}$, then any element of G_1' has 1's in the last $n-k$ coordinates, so we can define a projection map π_1' whose kernel is G_1' by

$$\pi_1'(g_1, \ldots, g_n) := (1, \ldots, 1, g_{n-k+1}, \ldots, g_n).$$

Next we observe that the elements $g^{\vec{x}_1}, \ldots, g^{\vec{x}_k}$ produced by $\mathcal{G}_k^n$ can be viewed as coming from a (right) action of an $n \times n$ matrix on the elements $g^{\vec{e}_1}, \ldots, g^{\vec{e}_k}$. More precisely, for $\mathbf{g} = (g_1, \ldots, g_n) \in G$ and a matrix $M = (a_{ij}) \in \mathrm{Mat}_n(\mathbb{F}_p)$, we define $\mathbf{g}^M$ by

$$\mathbf{g}^M := \left(\prod_{i=1}^n g_i^{a_{i1}}, \ldots, \prod_{i=1}^n g_i^{a_{in}}\right).$$

With this definition, we have $(g^{\vec{x}})^M = g^{(\vec{x}M)}$.

Now let M be an invertible matrix whose first k rows are the vectors $\vec{x}_i$. Then $g^{\vec{x}_i} = g^{\vec{e}_i M}$. If we define U_k to be the matrix with 1's in the last $n-k$ diagonal places and zeroes elsewhere, then the map π'_1 is given by $\pi'_1(\mathbf{g}) = \mathbf{g}^{U_k}$. Thus we can construct a projection map π_1 on G_1 by applying M^{-1} to map to G'_1, using π'_1 to project, and acting by M to map back to G_1; that is, $\pi_1(\mathbf{g}) = \mathbf{g}^{M^{-1}U_k M}$. We define π_2 analogously on H by computing an invertible matrix M' whose first k rows are the $\vec{y}_i$ produced by $\mathcal{G}_k^n$.

We now define the pairing e, the subgroup G'_t, and the projection map π_t. Recall that the *tensor product* of two n-dimensional vectors $\vec{x}, \vec{y}$ is

$$\vec{x} \otimes \vec{y} = (x_1\vec{y}, \ldots, x_n\vec{y}) = (x_1y_1, \ldots, x_1y_n, \ldots, x_ny_1, \ldots, x_ny_n).$$

We define $e : G \times H \to G_t := \mathbb{G}_t^{n^2}$ by $e(g^{\vec{x}}, h^{\vec{y}}) := \hat{e}(g,h)^{\vec{x}\otimes\vec{y}}$. That is, to pair $\mathbf{g} \in G$ and $\mathbf{h} \in H$, we take all the n^2 componentwise pairings $e(g_i, h_j)$ and write them in lexicographical order. In this case the A_ℓ of Theorem 2.5 are the n^2 matrices with a 1 in entry (i,j) and zeroes elsewhere; it is easy to see that these A_ℓ definite a nondegenerate pairing e

Defining the pairing in this manner allows us to define the map π_t abstractly as the tensor product of the maps π_1 and π_2. In terms of the matrices we have defined, we have

$$\pi_t(\mathbf{g}_t) = \mathbf{g}_t^{(M^{-1}\otimes M'^{-1})(U_k\otimes U_k)(M\otimes M')},$$

where $\otimes$ indicates the tensor product (or *Kronecker product*) of matrices: if $A = (a_{ij})$ and $B = (b_{ij})$ are two $n \times n$ matrices, then $A \otimes B$ is the $n^2 \times n^2$ matrix which, when divided into $n \times n$ blocks, has the (i,j)th block equal to $a_{ij}B$.

Given this framework, we see that the constructions of Groth and Sahai [14, Section 5] are exactly the above with $(n,k) = (2,1)$ and $(3,2)$. We now give explicit details for the first case.

Example 3.3. Let $\mathcal{P}$ be a prime-order bilinear group generator. Define $\mathcal{G}_P$ to be a bilinear group generator that on input λ does the following:

1. Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, \hat{e}) \xleftarrow{\text{R}} \mathcal{P}(\lambda)$, and let $G = \mathbb{G}_1^2$, $H = \mathbb{G}_2^2$, $G_t = \mathbb{G}_t^4$.
2. Choose generators $g \xleftarrow{\text{R}} \mathbb{G}_1$, $h \xleftarrow{\text{R}} \mathbb{G}_2$, and let $\gamma = e(g,h)$.
3. Choose random $a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2 \xleftarrow{\text{R}} \mathbb{F}_p$, such that $a_1d_1 - b_1c_1 = a_2d_2 - b_2c_2 = 1$.
4. Let G_1 be the subgroup of G generated by (g^{a_1}, g^{b_1}), let H_1 be the subgroup of H generated by (h^{a_2}, h^{b_2}), and let G'_t be the subgroup of G_t generated by
$$\{\gamma^{(a_1a_2, a_1b_2, b_1a_2, b_1b_2)}, \gamma^{(a_1c_2, a_1d_2, b_1c_2, b_1d_2)}, \gamma^{(c_1a_2, d_1b_2, c_1a_2, d_1b_2)}\}.$$
5. Define $e : G \times H \to G_t$ by
$$e((g_1,g_2),(h_1,h_2)) := (\hat{e}(g_1,h_1), \hat{e}(g_1,h_2), \hat{e}(g_2,h_1), \hat{e}(g_2,h_2)).$$

6. Let $A = \begin{pmatrix} -b_1c_1 & -b_1d_1 \\ a_1c_1 & a_1d_1 \end{pmatrix}$, $B = \begin{pmatrix} -b_2c_2 & -b_2d_2 \\ a_2c_2 & a_2d_2 \end{pmatrix}$, and define

$$\begin{aligned} \pi_1((g_1, g_2)) &:= (g_1, g_2)^A = (g_1^{-b_1c_1} g_2^{a_1c_1}, g_1^{-b_1d_1} g_2^{a_1d_1}) \\ \pi_2((h_1, h_2)) &:= (h_1, h_2)^B = (h_1^{-b_2c_2} h_2^{a_2c_2}, h_1^{-b_2d_2} h_2^{a_2d_2}) \\ \pi_t((\gamma_1, \gamma_2, \gamma_3, \gamma_4)) &:= (\gamma_1, \gamma_2, \gamma_3, \gamma_4)^{A \otimes B} \end{aligned}$$

7. Output the tuple $(G, G_1, H, H_1, G_t, G_t', e, \pi_1, \pi_2, \pi_t)$.

It is easy (though tedious) to check that $\mathcal{G}_P$ is a projecting bilinear group generator. We note that the groups output by $\mathcal{G}_P$ can be described simply by giving $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t$ and the pairs (g^{a_1}, g^{b_1}), (h^{a_2}, h^{b_2}). In particular, the group G_t' is generated by elements of the form $e(\mathbf{g}, \mathbf{h}_1)$ and $e(\mathbf{g}_1, \mathbf{h})$ with $\mathbf{g} \in G$, $\mathbf{g}_1 \in G_1$, $\mathbf{h} \in H$, and $\mathbf{h}_1 \in H_1$. This is important since in applications the maps π_1, π_2, π_t will be "trapdoor" information used as the system's secret key.

Proposition 3.4. *If $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{G}_P$ satisfies the subgroup decision assumption.*

Proof. Since g is uniform in $\mathbb{G}_1$ and a_1, b_1, c_1 are uniformly random in $\mathbb{F}_p$, imposing the condition $a_1d_1 - b_1c_1 = 1$ does not introduce any deviations from uniformity in the generation of G_1 (and similar for G_2). We can thus apply Theorem 2.5 with $n = 2$, $k = 1$. □

Cancelling Pairings. The traitor-tracing scheme of Boneh, Sahai, and Waters [4], the predicate encryption scheme of Katz, Sahai, and Waters [16], and many other schemes based on bilinear groups of composite order use in an essential manner the fact that if two group elements g, h have relatively prime orders, then $e(g, h) = 1$. This property implies, for example, that we can use the two subgroups generated by g and h to encode different types of information, and the two components will remain distinct after the pairing operation. The following definition incorporates this concept into our framework.

Definition 3.5. Let $\mathcal{G}$ be a bilinear group generator (Definition 2.1). We say that $\mathcal{G}$ is *r-cancelling* if it also outputs groups $G_2, \ldots, G_r \subset G$ and $H_2, \ldots, H_r \subset H$, such that

1. $G \cong G_1 \times \cdots \times G_r$ and $H \cong H_1 \times \cdots \times H_r$,
2. $e(g_i, h_j) = 1$ whenever $g_i \in G_i$, $h_j \in H_j$, and $i \neq j$.

Example 3.6. The bilinear group generator $\mathcal{G}_{BGN}$ of Example 2.3 is 2-cancelling: we set $G_2 = H_2$ to be the order-p_2 subgroup of $E(\mathbb{F}_p)$. An analogous r-cancelling generator can be built by making the group order N a product of r distinct primes.

Given a prime-order bilinear group generator $\mathcal{P}$, we now show how to modify the bilinear group generator $\mathcal{G}_1^n$ constructed in Theorem 2.5 so it is n-cancelling. We define the pairing $e : G \times H \to G_t := \mathbb{G}_t$ to be

$$e((g_1, \ldots, g_n), (h_1, \ldots, h_n)) := \textstyle\prod_{i=1}^n \hat{e}(g_i, h_i), \tag{3.1}$$

so we have $e(g^{\vec{x}}, h^{\vec{y}}) = e(g, h)^{\vec{x}\cdot\vec{y}}$, where $\cdot$ indicates the vector dot product; this pairing is necessarily nondegenerate.

If $\mathcal{G}_1^n$ is n-cancelling, then the subgroups G_i, H_i are all cyclic of order p. Thus we need to choose generators $g^{\vec{x}_i}$ of G_i and $h^{\vec{y}_i}$ of H_i such that $\vec{x}_i \cdot \vec{y}_j = 0$ if and only if $i = j$. This is straightforward: we first choose any set of n linearly independent $\vec{x}_i$; then the equation $\vec{x}_i \cdot \vec{y}_j = 0$ for all $i \neq j$ gives a linear system n variables of rank $n-1$, so there is a one-dimensional solution space in $\mathbb{F}_p^n$. If we choose $\vec{y}_j$ in this space then with high probability we have $\vec{x}_j \cdot \vec{y}_j \neq 0$; if this is not the case then we can start again with a different set of $\vec{x}_i$. We illustrate with concrete examples for $n = 2$ and 3. We use the notation $\langle X \rangle$ to indicate the cyclic group generated by X.

Example 3.7. Let $\mathcal{P}$ be a prime-order bilinear group generator. Define $\mathcal{G}_{3C}$ to be a bilinear group generator that on input λ does the following:

1. Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_t, \hat{e}) \overset{\mathrm{R}}{\leftarrow} \mathcal{P}(\lambda)$, and let $G = \mathbb{G}_1^3$, $H = \mathbb{G}_2^3$, $G_t = \mathbb{G}_t$.
2. Choose generators $g_1, g_2, g_3 \overset{\mathrm{R}}{\leftarrow} \mathbb{G}_1$, $h_1, h_2, h_3 \overset{\mathrm{R}}{\leftarrow} \mathbb{G}_2$.
3. Choose random $x, y, z, u, v, w \overset{\mathrm{R}}{\leftarrow} \mathbb{F}_p$, with $\begin{cases} -xv-xw-yu+yw+zu+zv\neq 0, \\ xv-xw-yu+yw+zu-zv\neq 0. \end{cases}$
4. Define the subgroups

$$G_1 = \langle (g_1, g_1^x, g_1^u) \rangle,\ G_2 = \langle (g_2, g_2^y, g_2^v) \rangle,\ G_3 = \langle (g_3, g_3^z, g_3^w) \rangle,$$
$$H_1 = \langle (h_1^{zv-yw}, h_1^{w-v}, h_1^{y-z}) \rangle,\ H_2 = \langle (h_2^{zu-xw}, h_2^{w-u}, h_2^{z-x}) \rangle,$$
$$H_3 = \langle (h_3^{yu-xv}, h_3^{v-u}, h_3^{x-y}) \rangle.$$

5. Define $e : G \times H \to G_t$ by (3.1) (with $n = 3$).
6. Output the tuple $(G, G_1, G_2, G_3, H, H_1, H_2, H_3, G_t, e)$.

It is straightforward to show that $\mathcal{G}_{3C}$ is a 3-cancelling bilinear group generator. The inequalities in Step (3) guarantee non-degeneracy of the pairing e. Note that choosing the elements g_1, g_2, g_3 independently uniform allows us to scale the vectors $\vec{x}_1 = (1, x, u)$, $\vec{x}_2 = (1, y, v)$, $\vec{x}_3 = (1, z, w)$ so their first components are 1 without losing uniformity.

Example 3.8. We define a 2-cancelling bilinear group generator $\mathcal{G}_{2C}$ by restricting the construction in Example 3.7 to the first two components. Explicitly, we have $G = \mathbb{G}_1^2$, $H = \mathbb{G}_2^2$, $G_t = \mathbb{G}_t$ and we set $u = 0$, $v = 0$, $w = 1$ to obtain

$$G_1 = \langle (g_1, g_1^x) \rangle, \ G_2 = \langle (g_2, g_2^y) \rangle, \ H_1 = \langle (h_1^{-y}, h_1) \rangle, \ H_2 = \langle (h_2^{-x}, h_2) \rangle.$$

We define $e : G \times H \to G_t$ by (3.1) and output $(G, G_1, G_2, H, H_1, H_2, G_t, e)$.

Example 3.9. If we have a symmetric pairing (i.e. $\mathbb{G}_1 = \mathbb{G}_2$), then for any $n > k > 1$ we can obtain an $(n - k + 1)$-cancelling bilinear group generator $\mathcal{G}_L(n, k)$ by doing the following:

1. Let $(p, \mathbb{G}, \mathbb{G}_t, \hat{e}) \overset{\text{R}}{\leftarrow} \mathcal{P}(\lambda)$, and let $G = H = \mathbb{G}^n$, $G_t = \mathbb{G}_t$.
2. Choose $\vec{x}_1, \ldots, \vec{x}_n \overset{\text{R}}{\leftarrow} \mathbb{F}_p^n$, such that $\{\vec{x}_i\}$ is linearly independent and for all $i > k$ we have $\vec{x}_i \cdot \vec{x}_j = 0$ if $i \neq j$, and $\vec{x}_i \cdot \vec{x}_j \neq 0$ if $i = j$.
3. Choose a generator $g \overset{\text{R}}{\leftarrow} \mathbb{G}$, and let $\gamma_i = g^{\vec{x}_i} \in G$.
4. Let $G_1 = \langle \gamma_1, \ldots, \gamma_k \rangle$, and $G_i = \langle \gamma_{i+k-1} \rangle$ for $2 \leq i \leq n - k + 1$.
5. Define e by (3.1) and output $(G, G_1, \ldots, G_{n-k+1}, G_t, e)$.

Proposition 3.10. *If $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then $\mathcal{G}_{3C}$ and $\mathcal{G}_{2C}$ satisfy the subgroup decision assumption. If $\mathbb{G}_1 = \mathbb{G}_2$ and $\mathcal{P}$ satisfies the k-linear assumption in $\mathbb{G}_1$, then $\mathcal{G}_L(n, k)$ satisfies the subgroup decision assumption.*

Proof. Recall that an SDP adversary is given only G, G_1, H, H_1, and not a description of any G_i or H_i for $i \geq 2$. Since in each case the generators of G_1 and H_1 are independent and uniform, the outputs of $\mathcal{G}_{3C}$, $\mathcal{G}_{2C}$, and $\mathcal{G}_L(n, k)$ are distributed identically to the output of $\mathcal{G}_k^n$ (for the appropriate values of n, k) so we may apply Theorem 2.5. □

4 Performance Analysis

Our primary motivation for converting composite-order group protocols to prime-order groups is to improve efficiency in implementations. This improvement results from the fact that we can use smaller prime-order groups than composite-order groups at equivalent security levels. We now examine this improvement concretely. Specifically, we compare the sizes of the groups G, H, and G_t produced by the bilinear group generator $\mathcal{G}_{BGN}$ (Example 2.3) with the four examples from Section 3 of bilinear group generators built from prime-order generators.

For the generators $\mathcal{G}_P$ (Example 3.3), $\mathcal{G}_{3C}$ (Example 3.7), and $\mathcal{G}_{2C}$ (Example 3.8) we take the prime-order bilinear group generator $\mathcal{P}$ to be an algorithm that produces a "pairing-friendly" ordinary elliptic curve E over a finite field $\mathbb{F}_q$. On such curves there are two "distinguished" subgroups $\mathbb{G}_1$ and $\mathbb{G}_2$ of order p in which the DDH problem is presumed to be infeasible, and such that the Tate pairing $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_t \subset \mathbb{F}_{q^k}^*$ is nondegenerate. Here k is the *embedding degree*, defined to be the smallest integer such that p divides the order of $\mathbb{F}_{q^k}^*$.

The ordinary elliptic curves E that give the best performance while providing discrete log security comparable to three commonly proposed levels of AES security are as follows. The group sizes follow the 2007 NIST recommendations [1]. Further details can be found in the full version of this paper [10, Appendix A]; descriptions of the elliptic curves are in [9].

80-bit security: A 170-bit MNT curve with embedding degree $k = 6$.
128-bit security: A 256-bit Barreto-Naehrig curve with $k = 12$.
256-bit security: A 640-bit Brezing-Weng curve with $k = 24$.

The advantage of the generator $\mathcal{G}_L$ is that we can use a prime-order group with a symmetric pairing, which only exists on supersingular elliptic curves. Thus in this case we take $\mathcal{P}$ to produce a supersingular curve over $\mathbb{F}_{3^m}$ with embedding degree $k = 6$. The fields that provide the best "match" for group orders at our three security levels are $\mathbb{F}_{3^{111}}$, $\mathbb{F}_{3^{323}}$, and $\mathbb{F}_{3^{1615}}$. Since 6 is the maximum possible embedding degree for supersingular curves, at high security levels the group $\mathbb{G}_1$ will be much larger than the group $\mathbb{G}_1$ on an equivalent ordinary curve.

Table 1 compares the sizes of the groups produced by all of our bilinear group generators at each of the three security levels. In all cases the groups G and H built using products of prime-order groups are much smaller than the groups G and H built using composite-order groups. The group G_t for the projecting generator $\mathcal{G}_P$ is twice as large as the composite-order G_t, due to the fact that elements of G_t are four elements of $\mathbb{F}_{q^k}$. However, the groups G_t for the cancelling generators $\mathcal{G}_{2C}$, $\mathcal{G}_{3C}$, $\mathcal{G}_L$ are half as large as the composite-order G_t.

The last column indicates the number of elliptic curve pairings required to compute the pairing e for the specified generator. While the prime-order generators require more pairings than the composite-order generator $\mathcal{G}_{BGN}$, the sizes of the elliptic curve groups in this case are so much smaller that the pairings will be far more than four times faster. For example, the Tate pairing on a 1024-bit supersingular curve runs ≈ 50 times slower than the Tate pairing on a 170-bit MNT curve [18], so a pairing for $\mathbb{G}_P$ at the 80-bit security level will be roughly 12 times faster than a pairing for $\mathbb{G}_{BGN}$.

Table 1. Estimated bit sizes of group elements for bilinear group generators at three different security levels.

	80-bit AES			128-bit AES			256-bit AES			#Pai-
Bilinear group generator	G	H	G_t	G	H	G_t	G	H	G_t	rings
$\mathcal{G}_{BGN}$ (Example 2.3)	1024	1024	2048	3072	3072	6144	15360	15360	30720	1
$\mathcal{G}_P$ (Example 3.3)	340	680	4080	512	1024	12288	1280	5120	61440	4
$\mathcal{G}_{3C}$ (Example 3.7)	510	1020	1020	768	1536	3072	1920	7680	15360	3
$\mathcal{G}_{2C}$ (Example 3.8)	340	680	1020	512	1024	3072	1280	5120	15360	2
$\mathcal{G}_L(3,2)$ (Example 3.9)	528	528	1056	1536	1536	3072	7680	7680	15360	3
$\mathcal{G}_L(4,2)$ (Example 3.9)	704	704	1056	2048	2048	3072	10240	10240	15360	4

5 Application: The BGN Cryptosystem

Our first application of the framework developed above is to the public-key encryption scheme of Boneh, Goh, and Nissim [3]. This scheme has the feature that given two ciphertexts, anyone can create a new ciphertext that encrypts either the sum or the product of the corresponding plaintexts. The product operation can only be carried out once; the system is thus "partially doubly homomorphic."

Step 1 of the conversion process is to write the scheme in the abstract framework and transfer it to asymmetric groups. In the original BGN protocol any ciphertext may be paired with any other ciphertext, so in the asymmetric setting each computation in G must be duplicated in H. We must use a projecting pairing, as the decryption algorithm requires projection away from a certain subgroup.

$\mathsf{KeyGen}(\lambda)$**:** Let $\mathcal{G}$ be a projecting bilinear group generator (Definition 3.1). Compute $(G, G_1, H,\ H_1, G_t, G'_t, e, \pi_1, \pi_2, \pi_t) \leftarrow \mathcal{G}(\lambda)$. Choose $g \overset{\mathrm{R}}{\leftarrow} G$, $h \overset{\mathrm{R}}{\leftarrow} H$, and output the public key $PK = (G, G_1, H, H_1, G_t, e, g, h)$ and the secret key $SK = (\pi_1, \pi_2, \pi_t)$.

$\mathsf{Encrypt}(PK, m)$**:** Choose $g_1 \overset{\mathrm{R}}{\leftarrow} G_1$ and $h_1 \overset{\mathrm{R}}{\leftarrow} H_1$. (Recall that the output of $\mathcal{G}$ allows random sampling from G_1 and H_1.) Output the ciphertext $(C_A, C_B) = (g^m \cdot g_1,\ h^m \cdot h_1) \in G \times H$.

$\mathsf{Multiply}(PK, C_A, C_B)$**:** This algorithm takes as input two ciphertexts $C_A \in G$ and $C_B \in H$. Choose $g_1 \overset{\mathrm{R}}{\leftarrow} G_1$ and $h_1 \overset{\mathrm{R}}{\leftarrow} H_1$, and output $C = e(C_A, C_B) \cdot e(g, h_1) \cdot e(g_1, h) \in G_t$.

$\mathsf{Add}(PK, C, C')$**:** This algorithm takes as input two ciphertexts C, C' in one of G, H, or G_t. Choose $g_1 \overset{\mathrm{R}}{\leftarrow} G_1$, $h_1 \overset{\mathrm{R}}{\leftarrow} H_1$, and do the following:

1. If $C, C' \in G$, output $C \cdot C' \cdot g_1 \in G$.
2. If $C, C' \in H$, output $C \cdot C' \cdot h_1 \in H$.
3. If $C, C' \in G_t$, output $C \cdot C' \cdot e(g, h_1) \cdot e(g_1, h) \in G_t$.

Decrypt(SK, C)**:** The input ciphertext C is an element of G, H, or G_t.

1. If $C \in G$, output $m \leftarrow \log_{\pi_1(g)}(\pi_1(C))$.
2. If $C \in H$, output $m \leftarrow \log_{\pi_2(h)}(\pi_2(C))$.
3. If $C \in G_t$, output $m \leftarrow \log_{\pi_t(e(g,h))}(\pi_t(C))$.

It is clear that if C, C' are encryptions of m, m' respectively, then the Add algorithm gives a correctly distributed encryption of $m + m'$. Furthermore, it follows from the bilinear property of the pairing that if $C_A \in G$, $C_B \in H$ are the left and right halves of encryptions of m, m' respectively, then the Multiply algorithm gives a correctly distributed encryption of $m \cdot m'$. Since there is no pairing on G_t we can only perform the multiplication once.

Correctness of decryption of ciphertexts in G and H follows from the fact that G_1, H_1 are in the kernels of π_1, π_2, respectively. Correctness of decryption of ciphertexts in G_t follows from the "projecting" properties of $\mathcal{G}$; for example, we have $\pi_t(e(g, h_1)) = e(\pi_1(g), \pi_2(h_1)) = e(\pi_1(g), 1) = 1$.

Step 2 of the conversion process is to translate the security assumptions to asymmetric bilinear groups. In this case, semantic security of ciphertexts in G follows from the subgroup decision assumption on the left for $\mathcal{G}$. Intuitively, if $\mathcal{G}$ satisfies the subgroup decision assumption on the left, then an adversary cannot distinguish the real system from a "fake" system in which $g \in G_1$. Semantic security then follows from the fact that in the fake system the ciphertext element C_A will be a uniformly random element of G_1 and thus will contain no information about the message m. The same argument holds for ciphertexts in H, and semantic security of ciphertexts in G_t follows from semantic security in G and H. For further details see [3, Theorem 3.1].

Step 3 is to translate the assumption to prime-order groups. Since the security proof uses no intrinsic properties of the groups G and H, it carries over to our more general setting.

Theorem 5.1. *Let $\mathcal{P}$ be a prime-order bilinear group generator, and let $\mathcal{G}_P$ be the projecting bilinear group generator constructed from $\mathcal{P}$ in Example 3.3. If $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$, then the BGN cryptosystem instantiated with $\mathcal{G} = \mathcal{G}_P$ is semantically secure.*

When instantiated with either $\mathcal{G}_{BGN}$ or $\mathcal{G}_P$, decryption in the BGN system requires taking discrete logarithms in a group of large prime order. Thus to achieve efficient decryption the message space must be small (i.e., logarithmic in the group size). It is an open problem to find a projecting bilinear group generator $\mathcal{G}$ for which the subgroup decision assumption may hold and for which discrete logarithms can be computed in a subset of $\pi_1(G)$ whose size is a constant fraction of the full group order.

If we carry out the tensor product construction described in Section 3 for any k and $n \geq k+1$, we obtain an instantiation of the BGN cryptosystem whose security depends on the k-Linear assumption. Since ciphertexts will consist of n elements of $\mathbb{G}_1$ or $\mathbb{G}_2$ or n^2 elements of $\mathbb{G}_t$, these systems will be less efficient than the system constructed using $\mathcal{G}_P$, which has $(n,k) = (2,1)$. We do note, however, that if $k \geq 2$ we can use a group with a symmetric pairing, in which case the $\mathsf{Encrypt}$ algorithm needs only to output the ciphertext C_A.

6 More Applications

We conclude by summarizing several further applications of our framework to cryptosystems constructed using composite-order groups. Details can be found in the full version of this paper [10].

Traitor Tracing. Boneh, Sahai, and Waters [4] construct a traitor tracing system that is fully collusion resistant and has short ciphertexts. After reducing the construction of their system to construction of a primitive called *private linear broadcast encryption* (or PLBE), Boneh et al. devise a PLBE scheme using bilinear groups of composite order and show it secure under three assumptions in bilinear groups: the subgroup decision assumption, the *3-party Diffie-Hellman assumption*, and the *bilinear subgroup decision assumption*.

To apply our general framework to the Boneh et al. PLBE scheme, we first write the original system using asymmetric pairings on abstract groups, and then convert the three assumptions to this more general context. We instantiate the system using the 2-cancelling bilinear group generator $\mathcal{G}_{2C}$ of Example 3.8 and obtain the following security theorem.

Theorem 6.1. *Let $\mathcal{P}$ be a prime-order bilinear group generator, and let $\mathcal{G}_{2C}$ be the 2-cancelling bilinear group generator constructed from $\mathcal{P}$ in Example 3.8. If $\mathcal{P}$ satisfies the DDH assumption in $\mathbb{G}_2$ and the 3-party DDH assumptions in $\mathbb{G}_1$ and $\mathbb{G}_2$ (i.e., given g, g^a, g^b, g^c, no efficient adversary can distinguish g^{abc} from a random group element), then the Boneh-Sahai-Waters PLBE system is secure when instantiated with $\mathcal{G} = \mathcal{G}_{2C}$.*

Predicate Encryption. Katz, Sahai, and Waters [16] construct a predicate encryption scheme using bilinear groups whose order is a product N

of three distinct primes. The security of the system is based on two complex (yet constant-size) assumptions in composite-order bilinear groups, which we call Assumptions 1 and 2; both can be seen as variants of the subgroup decision problem.

To apply our general framework to this scheme, we write the scheme using a bilinear group generator with an asymmetric pairing and translate the security assumptions into this more general context. We then instantiate the system in two different ways, using the 3-cancelling bilinear group generators $\mathcal{G}_{3C}$ of Example 3.7 and $\mathcal{G}_L(4,2)$ of Example 3.9. When using $\mathcal{G}_{3C}$, translating the asymmetric versions of Assumptions 1 and 2 explicitly to this setting produces two new (constant-size) assumptions in prime-order groups; we call these Assumptions 3 and 4. (We also show that these assumptions hold in the generic group model.) When using $\mathcal{G}_L(4,2)$ we can use simpler assumptions at the expense of a less efficient system (cf. Table 1). We obtain the following security theorem for $\mathcal{G}_{3C}$; details for both cases appear in the full paper.

Theorem 6.2. *Let $\mathcal{P}$ be a prime-order bilinear group generator, and let $\mathcal{G}_{3C}$ be the* 3*-cancelling bilinear group generator constructed from $\mathcal{P}$ in Example 3.7. If $\mathcal{P}$ satisfies Assumptions 3 and 4, then the Katz-Sahai-Waters predicate encryption scheme is secure when instantiated with $\mathcal{G} = \mathcal{G}_{2C}$.*

Further Work. We expect that our framework can be used to create prime-order group instantiations of other cryptosystems that use composite-order bilinear groups. However, since our construction is not black box, the security proof of each cryptosystem must be checked to ensure that it is still valid in our more general framework. For example, the proof of the Lewko-Waters IBE system [17] uses in an essential way the fact that G has two subgroups with relatively prime order; thus our prime-order construction does not apply in this case. Lewko and Waters do give a version of their system in prime-order groups, with a different security proof under new assumptions. It remains an open problem to find a framework that incorporates both versions of the system.

Acknowledgments. The author thanks Dennis Hofheinz, Eike Kiltz, and Brent Waters for helpful discussions.

References

1. E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. “Recommendation for key management — Part 1: General (revised).” NIST Special Pub. 800-57 (2007).

2. D. Boneh, X. Boyen, and H. Shacham. "Short group signatures." In *Crypto 2004*, Springer LNCS **3152** (2004), 41–55.
3. D. Boneh, E.-J. Goh, and K. Nissim. "Evaluating 2-DNF formulas on ciphertexts." In *TCC 2005*, Springer LNCS **3378** (2005), 325–341.
4. D. Boneh, A. Sahai, and B. Waters. "Fully collusion resistant traitor tracing with short ciphertexts and private keys." In *Eurocrypt 2006*, Springer LNCS **4004** (2006), 573–592.
5. D. Boneh and B. Waters. "Conjunctive, subset, and range queries on encrypted data." In *TCC 2007*, Springer LNCS **4392** (2007), 535–554.
6. X. Boyen and B. Waters. "Full-domain subgroup hiding and constant-size group signatures." In *PKC 2007*, Springer LNCS **4450** (2007), 1–15.
7. R. Cramer and V. Shoup. "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack." *SIAM Journal on Computing* **33** (2003), 167–226.
8. S. Duquesne and T. Lange. "Pairing-based cryptography." In *Handbook of Elliptic and Hyperelliptic Curve Cryptography.* Chapman & Hall/CRC, Boca Raton, FL (2006), 573–590.
9. D. Freeman, M. Scott, and E. Teske. "A taxonomy of pairing-friendly elliptic curves." *Journal of Cryptology* **23** (2010), 224–280.
10. D. M. Freeman. "Converting pairing-based protocols from composite-order groups to prime-order groups." Cryptology ePrint Archive, Report 2009/540 (2009). Available at `http://eprint.iacr.org/2009/540`.
11. S. Galbraith and E. Verheul. "An analysis of the vector decomposition problem." In *PKC 2008*, Springer LNCS **4939** (2008), 308–327.
12. K. Gjøsteen. *Subgroup membership problems and public key cryptosystems.* Ph.D. dissertation, Norwegian University of Science and Technology (2004). Available at `http://ntnu.diva-portal.org/smash/get/diva2:121977/FULLTEXT01`.
13. J. Groth, R. Ostrovsky, and A. Sahai. "Perfect non-interactive zero knowledge for NP." In *Eurocrypt 2006*, Springer LNCS **4004** (2006), 339–358.
14. J. Groth and A. Sahai. "Efficient non-interactive proof systems for bilinear groups." In *Eurocrypt 2008*, Springer LNCS **4965** (2008), 415–432.
15. D. Hofheinz and E. Kiltz. "Secure hybrid encryption from weakened key encapsulation." In *Crypto 2007*, Springer LNCS **4622** (2007), 553–571.
16. J. Katz, A. Sahai, and B. Waters. "Predicate encryption supporting disjunctions, polynomial equations, and inner products." In *Eurocrypt 2008*, Springer LNCS **4965** (2008), 146–162. Full version at `http://eprint.iacr.org/2007/404`.
17. A. Lewko and B. Waters. "New techniques for dual system encryption and fully secure HIBE with short ciphertexts." In *TCC 2010*, Springer LNCS **5978** (2010), 455–479.
18. M. Scott. Personal communication (17 February 2009).
19. H. Shacham. "A Cramer-Shoup encryption scheme from the Linear assumption and from progressively weaker Linear variants." Cryptology ePrint Archive, Report 2007/074 (2007). Available at `http://eprint.iacr.org/2007/074`.
20. H. Shacham and B. Waters. "Efficient ring signatures without random oracles." In *PKC 2007*, Springer LNCS **4450** (2007), 166–180.
21. B. Waters. "Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions." In *Crypto 2009*, Springer LNCS **5677** (2009), 619–636.
22. M. Yoshida. "Inseparable multiplex transmission using the pairing on elliptic curves and its application to watermarking." Proc. 5th Conf. on Algebraic Geometry, Number Theory, Coding Theory and Cryptography, Univ. of Tokyo (2003). Available at `http://www.math.uiuc.edu/~duursma/pub/yoshida_paper.pdf`.