Bivariate Polynomials Modulo Composites and their Applications

Dan Boneh and Henry Corrigan-Gibbs

Stanford University, Stanford CA 94305, U.S.A.

Abstract. We investigate the hardness of finding solutions to bivariate polynomial congruences modulo RSA composites. We establish necessary conditions for a bivariate polynomial to be one-way, second preimage resistant, and collision resistant based on arithmetic properties of the polynomial. From these conditions we deduce a new computational assumption that implies an efficient algebraic collision-resistant hash function. We explore the assumption and relate it to known computational problems. The assumption leads to (i) a new statistically hiding commitment scheme that composes well with Pedersen commitments, (ii) a conceptually simple cryptographic accumulator, and (iii) an efficient chameleon hash function.

Keywords: algebraic curves, bivariate polynomials, cryptographic commitments, Merkle trees

1 Introduction

In this paper, we investigate the cryptographic properties of bivariate polynomials modulo random RSA composites $N = pq$. We ask: for which integer polynomials $f \in \mathbb{Z}[x, y]$ does the function $f : \mathbb{Z}_N \times \mathbb{Z}_N \to \mathbb{Z}_N$ defined by f appear to be a one-way function, a second-preimage-resistant function, or a collision-resistant function? We say that a polynomial $f \in \mathbb{Z}[x, y]$ is one-way if the function $f : \mathbb{Z}_N \times \mathbb{Z}_N \to \mathbb{Z}_N$ defined by f is one-way (Section 3.1). We similarly define second-preimage-resistance (Section 3.2) and collision-resistance (Section 3.3) of polynomials $f \in \mathbb{Z}[x, y]$.

Using tools from algebraic geometry we develop a heuristic for deducing the cryptographic properties of a bivariate polynomial over $\mathbb{Z}_N$ from its arithmetic properties, namely from its properties as a polynomial over the rationals $\mathbb{Q}$. We give a number of necessary conditions for a bivariate polynomial to be one-way, second-preimage-resistant, or collision-resistant. We also provide examples of polynomials f that appear to satisfy each of these properties and we offer separations between these three classes.

Taking collision resistance as an example, we conjecture that a bivariate polynomial $f \in \mathbb{Z}[x, y]$ that defines an *injective* function $f : \mathbb{Q}^2 \to \mathbb{Q}$ gives a *collision resistant* function $f : \mathbb{Z}_N^2 \to \mathbb{Z}_N$ where N is a random RSA modulus of secret factorization (see Section 3.3). Constructing an explicit polynomial $f \in \mathbb{Z}[x, y]$ that is provably injective over the rationals is an open number theoretic problem [39]. However, even relatively simple low-degree polynomials appear to be injective over $\mathbb{Q}^2$. For example, Don Zagier [17, 45] conjectures that the polynomial $f_{\text{ZAG}}(x, y) := x^7 + 3y^7$, which we refer to as the *Zagier polynomial*, is injective over the rationals. Clearly, a collision in f_{ZAG} over the rationals would imply a collision modulo many RSA moduli N. Therefore, injectivity of f_{ZAG} over $\mathbb{Q}$ is necessary for its collision resistance modulo random RSA composites. For the applications in this paper we conjecture that f_{ZAG} is indeed collision resistant when taken modulo random RSA composites. To build confidence in this assumption we discuss potential collision-finding strategies and relate them to existing number theoretic problems.

Applications. The existence of low-degree collision-resistant bivariate polynomials such as f_{ZAG} can be very useful: it gives rise to very efficient instantiations of a number of cryptographic primitives.

First, we derive a statistically hiding commitment scheme which is computationally inexpensive to evaluate and composes naturally with Pedersen commitments. By "nesting" these new commitments inside of Pedersen commitments, we obtain an efficient zero-knowledge protocol for proving knowledge of an opening of a commitment which is nested inside of another commitment. Use of nested commitments reduces the length of transactions in an anonymous e-cash scheme [32] by roughly 70%.

Second, we demonstrate that the new commitment scheme, in conjunction with Merkle trees, can serve as a simple replacement for one-way accumulators. Though the communication complexity of our accumulator construction is asymptotically worse than that of strong-RSA accumulators [12]—$O(\log |S|)$ versus $O(1)$ for a set S being accumulated—our construction has the benefit of being conceptually simple and easy to implement.

Third, from the same collision-resistant polynomial, we derive a new chameleon hash function, signature scheme, claw-free permutation family, and a variable-length algebraic hash function.

2 Related Work

Polynomials in Cryptography. Multivariate polynomials in $\mathbb{Z}_N$ have a long history in cryptography. For example, the security of the Ong-Schnorr-Shamir signature scheme [35] followed from the hardness of finding solutions to a particular type of bivariate polynomial equation over $\mathbb{Z}_N$. Pollard and Schnorr later demonstrated a general attack against the hardness of finding solutions to such equations [37].

Shamir related the hardness of factoring certain multivariate polynomials modulo N to the problem of factoring the modulus N itself [42]. Schwenk and Eisfeld proposed encryption and signature schemes reliant on the hardness of finding roots of random *univariate* polynomials $f \in \mathbb{Z}[x]$ modulo a composite N, and they prove that this problem is as hard as factoring N [40].

Commitment Schemes. This work introduces a new statistically hiding commitment scheme based on low-degree polynomials. Commitment schemes are used widely in cryptography. Prior work has derived statistically hiding commitment schemes from the discrete log problem [36], the Paillier cryptosystem [16], and the RSA problem [3]. Verifying the correctness of opening a commitment in these existing schemes requires expensive modular exponentiations or elliptic curve scalar multiplications. Verifying an opening with our new commitment scheme requires just a few modular multiplications. By combining our new commitment scheme with traditional Pedersen commitments, we improve the communication efficiency of the Zerocoin decentralized e-cash construction [32].

Merkle Trees and Accumulators. Given a Pedersen commitment and a finite set of elements S, our commitment scheme leads to a simple zero-knowledge protocol for proving knowledge of an opening x of the commitment such that $x \in S$. The length of the proof is $\log |S|$. This technique, which uses Merkle trees [29], has applications to anonymous authentication [20] and credential systems [27] and it has the potential to replace traditional RSA one-way accumulators, introduced by Benaloh and De Mare [8] and revisited by Barić and Pfitzmann [4]. Bilinear maps [34] and class groups [28] also give rise to accumulators, under different cryptographic assumptions.

Camenisch and Lysyanskaya presented an efficient zero-knowledge protocol for proving that a value contained in a Pedersen commitment is also contained in a particular strong-RSA accumulator [12]. The Camenisch-Lysyanskaya accumulator produces a shorter proof of knowledge than ours does, but the conceptual simplicity and ease of implementation may make our Merkle-style proof more attractive for some applications.

Ben-Sasson et al. also develop a zero-knowledge proof of an opening of a commitment which is the leaf of a particular Merkle tree [6]. Their techniques require zk-SNARKs [7], which are relatively difficult to implement and rely on strong knowledge-of-exponent assumptions [23].

The "zero-knowledge sets" of Micali, Rabin, and Kilian solve an orthogonal problem from that of accumulators: a prover publishes a commitment to a set S and later can prove that $x \in S$ without leaking other information about S [31]. In contrast, we are interested in hiding the value x but allow the set of items S to be public.

3 Cryptographic Properties of Polynomials

We begin by surveying the cryptographic properties of integer polynomials modulo random RSA composites. Our goal is to relate the algebraic properties of polynomials to their cryptographic complexity. In particular, we identify families of integer polynomials that give rise to progressively stronger cryptographic primitives: one-way functions, second-preimage-resistant functions, and collision-resistant functions.

Notation. We write $x \overset{R}{\leftarrow} S$ to indicate that the variable x takes on a value sampled independently and uniformly at random from a finite set S. A function $f : \mathbb{Z} \to \mathbb{R}^+$ is *negligible* if it is smaller than $1/p(\lambda)$ for every polynomial $p()$ and all sufficiently large λ. We denote an arbitrary negligible function in λ as $\mathsf{negl}(\lambda)$. We use the notation $f(x) := x^2$ to indicate the definition of a new term.

In what follows, we let $\mathsf{RSAgen}(\lambda)$ denote a randomized algorithm that runs in time polynomial in λ. The algorithm generates two random $\mathsf{len}(\lambda)$-bit primes p and q and outputs $(p,\ q,\ N := p \cdot q)$. Here $\mathsf{len} : \mathbb{Z}^+ \to \mathbb{Z}^+$ is some fixed function that determines the size of the primes p and q as a function of λ.

Let $f \in \mathbb{Z}[x, y]$ be a bivariate polynomial. For $c \in \mathbb{Z}$ consider the curve $f(x, y) = c$. The *genus* of this curve is a standard measure of its "complexity:" conics have genus zero, elliptic curves have genus one, and so on (see, e.g. [2, 24]). We define the genus of a polynomial f as follows:

Definition 1. *The genus of a polynomial $f \in \mathbb{Z}[x, y]$ is defined as*

$$\max_{c \in \mathbb{Q}} \big(\text{genus}(\ f(x,y) = c\ )\big).$$

As we will see, the genus of a polynomial f has some relation to its cryptographic properties. While we focus on bivariate polynomials, most of the following discussion generalizes to multivariates.

We use the following terms throughout this section to describe relationships between curves. (For more precise definitions, see Hindry and Silverman [24, Sec. A.1.2].) A *rational map* from a curve C to another curve C' is a pair of rational functions g and h mapping points (x, y) on C to points $(g(x, y), h(x, y))$ on C'. A *birational map* from C to C' is a rational map which is a bijection between points on C and C' such that the map's inverse is also rational. Two curves C and C' and are *birationally equivalent* if there is a birational map from C to C'. An *automorphism* is a birational map from a curve to itself.

3.1 One-way Polynomials

One-way functions are the basis of much of cryptography. A function $g : X \to Y$ is *one-way* if, given the image $c = f(x)$ of a random point $x \in X$, it is hard to find an x' such that $f(x') = c$. We first ask: what polynomials give rise to one-way functions?

Definition 2. *A polynomial f in $\mathbb{Z}[x_1, ..., x_\ell]$ is* one-way *if for every p.p.t. algorithm $\mathcal{A}$ the following advantage is a negligible function of λ:*

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A},f}(\lambda) := \Pr[N \leftarrow \mathsf{RSAgen}(\lambda),\ \bar{x} \overset{R}{\leftarrow} (\mathbb{Z}_N)^\ell,\ c \leftarrow f(\bar{x})\ :& \\ f\big(\mathcal{A}(N, c)\big) = c \text{ in } \mathbb{Z}_N]&\ .\end{aligned}$$

Clearly linear polynomials are not one-way. A result of Pollard and Schnorr [37] shows that quadratic polynomials, indeed all genus zero polynomials, are not one-way.

Theorem 3. *A genus zero polynomial $f \in \mathbb{Z}[x, y]$ is not one-way.*

Proof sketch. For all $c \in \mathbb{Q}$ the curve $f(x, y) = c$ is of genus zero, or is a product of genus zero curves. A genus zero curve is birationally equivalent to a linear or quadratic curve $\tilde{f}(x, y) = 0$ [24, Theorem A.4.3.1]. If $\tilde{f}(x, y)$ is linear in one of the variables x or y then finding points on this curve is easy thereby breaking the one-wayness of f. This leaves the case where $\tilde{f}(x, y)$ is quadratic in both x and y. Let N be an output of $\mathsf{RSAgen}(\lambda)$. Let $\tilde{f} \in \mathbb{Z}[x, y]$ be a quadratic polynomial in x and y and let $c \in \mathbb{Z}_N$. There is an efficient algorithm that for most $c \in \mathbb{Z}_N$ finds an $(x_0, y_0) \in \mathbb{Z}_N^2$ such that $\tilde{f}(x_0, y_0) = c$ in $\mathbb{Z}_N$, breaking the one-wayness of f. See for example [9, Sec. 5.2] for a description of the algorithm. $\square$

Theorem 3 played an important role in analyzing the security of the Ong-Schnorr-Shamir signature scheme [35]. The scheme depended on the difficulty of finding solutions (x, y) to the equation:

$$x^2 + ay^2 = b \quad \text{in } \mathbb{Z}_N$$

for known constants $a, b \in \mathbb{Z}_N$. Since this equation defines a genus-zero curve, Theorem 3 shows that it is possible to efficiently find solutions without knowledge of the factors of N. Pollard and Schnorr demonstrated an attack against the scheme soon after its publication [37, 41].

One-way Polynomials. It is not known how to break the one-wayness of polynomials $f \in \mathbb{Z}[x, y]$ that are not genus zero. Thus, for example, even a simple polynomial such as $f(x, y) = y^2 - x^3$ may be one-way, although that would require further study.

3.2 Second Preimage Resistant Polynomials

A function $f : U \to V$ is *second preimage resistant* if, given $u \in U$, it is difficult to find a $u' \neq u \in U$ such that $f(u) = f(u')$. We define a similar notion for polynomials:

Definition 4. *A polynomial f in $\mathbb{Z}[x_1, ..., x_\ell]$ is* second preimage resistant *if, for every p.p.t. algorithm $\mathcal{A}$, the following advantage is a negligible function of λ:*

$$\mathsf{Adv}_{\mathcal{A},f}(\lambda) := \Pr[N \leftarrow \mathsf{RSAgen}(\lambda),\ \bar{x} \xleftarrow{R} (\mathbb{Z}_N)^\ell,\ \bar{x}' \leftarrow \mathcal{A}(N, \bar{x})\ :\ f(\bar{x}) = f(\bar{x}') \text{ in } \mathbb{Z}_N\ \wedge\ \bar{x} \neq \bar{x}']\ .$$

Since genus 0 polynomials are not one-way they are also not second preimage resistant. It is similarly straight-forward to show that no genus-one polynomial is second preimage resistant.

Proposition 5. *A genus one polynomial $f \in \mathbb{Z}[x, y]$ is not second preimage resistant.*

To see why, let $f \in \mathbb{Z}[x, y]$ be a polynomial such that $f(x, y) = c$ is a curve of genus one for all but finitely many $c \in \mathbb{Q}$. Then f is not second preimage resistant because of the group structure on elliptic curves. That is, let N be an output of $\mathsf{RSAgen}(\lambda)$. Choose a random pair $(x_0, y_0) \in \mathbb{Z}_N^2$ and set $c := f(x_0, y_0) \in \mathbb{Z}_N$. Then $P = (x_0, y_0)$ is a point on the curve $f(x, y) = c$ and so is the point $2P = P + P$ where addition refers to the elliptic curve group operation. With overwhelming probability $2P$ is not the point at infinity and therefore, given P as input, the adversary can output $2P$ as a second preimage for P. It follows that f is not second preimage resistant.

Even polynomials that give higher genus curves need not be second preimage resistant. For example, a hyperelliptic polynomial of genus $g \geq 2$ has the form $f(x, y) = y^2 - h(x) \in \mathbb{Z}[x, y]$ where $h \in \mathbb{Z}[x]$ is a polynomial of degree $2g + 1$ or $2g + 2$. The simple fact that $f(x_0, y_0) = f(x_0, -y_0)$ immediately gives a second preimage attack on these polynomials: given (x_0, y_0) the attacker outputs $(x_0, -y_0)$ as a second preimage.

The fact that all curves of genus two are hyperelliptic [24, Theorem A.4.5.1] leads to the following proposition:

Proposition 6. *A genus two polynomial $f \in \mathbb{Z}[x, y]$ is not second preimage resistant.*

This proposition, in combination with Theorem 3 and Proposition 5 means that all second preimage resistant polynomials must have genus at least three.

As outlined above, elliptic (genus one) and hyperelliptic (genus two) polynomials are not second preimage resistant because there are non-trivial automorphisms on the associated curves. We say that a polynomial $f \in \mathbb{Z}[x, y]$ is *automorphism free* if, for all but finitely many $c \in \mathbb{Q}$, the curve $f(x, y) = c$ has no automorphisms over $\mathbb{Q}$, apart from the trivial map $(x, y) \mapsto (x, y)$. It is natural to conjecture that every automorphism-free polynomial $f \in \mathbb{Z}[x, y]$ is second preimage resistant.

Poonen constructs a large family of automorphism-free polynomials, in arbitrarily many variables and of arbitrarily large degree [38]. For example, he proves that the polynomial $f(x, y) = x^3 + xy^3 + y^4$ is automorphism-free over the rationals [38].

A Historical Aside: q-Way Preimage Resistance. A stronger notion of preimage resistance for a function $f : U \to V$, called *q-way preimage resistance*, states that given a random $v \in V$ and random points $u_1, \ldots, u_q$ in U such that $v = f(u_1) = \cdots = f(u_q)$, it is difficult to find a new point $u \in U \setminus \{u_1, \ldots, u_q\}$ such that $f(u) = v$.

As before, one can define a similar property for polynomials. That is, a polynomial f in $\mathbb{Z}[x, y]$ is *q-way preimage resistant* if, for a random RSA moduli N and a random $c \in \mathbb{Z}_N$, given q points on the curve $f(x, y) = c$ in $\mathbb{Z}_N$, it is hard to find another point on this curve.

Kilian and Petrank [25] proposed an authentication scheme whose security is based on the q-way preimage resistance of the polynomial $f_{\text{KP}}(x, y) = x^e - y^e$, for some small odd e, say $e = 17$. In their scheme, q is the total number of users in the system. Naor [33] refers to the computational assumption that f_{KP} is q-way preimage resistant as the *Difference RSA Assumption*. We note that the polynomial f_{KP} is not even second preimage resistant because there is a non-trivial automorphism $(x, y) \mapsto (-y, -x)$ on the curve. In other words, for any point (x_0, y_0) we have that $f_{\text{KP}}(x_0, y_0) = f_{\text{KP}}(-y_0, -x_0)$. This bad symmetry appears to violate the security properties needed for the Kilian-Petrank identification scheme, but the scheme can be modified to resist such attacks.

Camenisch and Stadler [14, Sec. 6] used a similar assumption to construct group signatures. They need the polynomial $f_{\text{CS}}(x, y) = x^{e_1} + ay^{e_2}$ to be q-way preimage resistant for some small e_1 and e_2. They propose using $e_1 = 5$ and $e_2 = 3$. We observe in that next section that the polynomial $f(x, y) = x^5 + y^3$ is not collision resistant. Nevertheless, it may be second preimage resistant.

3.3 Collision-Resistant Polynomials

A function $f : U \to V$ is *collision resistant* if it is difficult to find a pair $u \neq u' \in U$ such that $f(u) = f(u')$. We define a similar notion for polynomials:

Definition 7. *For a polynomial f in $\mathbb{Z}[x_1, ..., x_\ell]$ and an integer N, we say that $\bar{x}, \bar{y} \in (\mathbb{Z}_N)^\ell$ are an N-*collision for *f if $f(\bar{x}) = f(\bar{y})$ in $\mathbb{Z}_N$ and $\bar{x} \neq \bar{y}$.*

Definition 8. *A polynomial f in $\mathbb{Z}[x_1, ..., x_\ell]$ is* collision resistant *if for every p.p.t. algorithm $\mathcal{A}$ the following advantage is a negligible function of λ:*

$$\mathsf{Adv}_{\mathcal{A},f}(\lambda) := \Pr\left[\ N \leftarrow \mathsf{RSAgen}(\lambda) \quad : \quad \mathcal{A}(N) \text{ is an } N\text{-collision for } f\ \right] .$$

In the previous two subsections, we observed that polynomials $f \in \mathbb{Z}[x, y]$ which are of genus $g \leq 2$ or which are hyperelliptic, are not second preimage resistant and thus are not collision resistant.

Even polynomials that *are* second preimage resistant are not necessarily collision resistant. For example, in Section 3.2 we suggested that the polynomial $f(x, y) = x^3 + xy^3 + y^4$ may be second preimage resistant. However, it is certainly not collision resistant, since for any $r \in \mathbb{Q}$, the points $(r^4, 0)$ and $(0, r^3)$ constitute a collision.

Attacking Collision Resistance Over the Rationals. Suppose that a polynomial $f \in \mathbb{Z}[x_1, \ldots, x_\ell]$ has a *rational* collision. That is, there are rational points $\bar{x}_0 \neq \bar{x}_1$ in $\mathbb{Q}^\ell$ such that $f(\bar{x}_0) = f(\bar{x}_1)$. Then, for most[1] RSA moduli N, the points $\bar{x}_0$ and $\bar{x}_1$ give a collision for f in $\mathbb{Z}_N$. This breaks the collision resistance of f when the security parameter λ is sufficiently large. Indeed, for sufficiently large λ the attack algorithm can construct the fixed rational points $\bar{x}_0$ and $\bar{x}_1$ by exhaustive search and obtain collisions for f for most RSA moduli output by $\mathsf{RSAgen}(\lambda)$.

[1] The points $\bar{x}_0$ and $\bar{x}_1$ give a collision in $\mathbb{Z}_N$ whenever N is relatively prime to their denominators and $\bar{x}_0 \neq \bar{x}_1 \bmod N$. This holds with overwhelming probability for sufficiently large λ.

The discussion above shows that if a polynomial $f \in \mathbb{Z}[x_1, \ldots, x_\ell]$ has a *rational* collision then f is not collision resistant. We summarize this in the following proposition.

Proposition 9. *If a polynomial $f \in \mathbb{Z}[x_1, \ldots, x_\ell]$ is collision resistant then the function $f : \mathbb{Q}^\ell \to \mathbb{Q}$ must be injective.*

If $f \in \mathbb{Z}[x_1, \ldots, x_\ell]$ defines an injective function from $\mathbb{Q}^\ell$ to $\mathbb{Q}$ then f is said to be an *injective polynomial.* Proposition 9 shows that the search for collision-resistant polynomials must begin with the search for an injective polynomial over the rationals.

Injective Polynomials. Even the *existence* of bivariate injective polynomials is currently an open problem. Poonen [39] shows that they exist under certain number theoretic conjectures. Moreover, Poonen [39, Lemma 2.3] shows that if $f \in \mathbb{Z}[x, y]$ has only a *finite* number of rational collisions then one can use f to construct an injective polynomial $g \in \mathbb{Z}[x, y]$ by pre-composing f with a suitable polynomial map. In other words, an “almost” injective polynomial can be converted to an injective one.

Although proving that a particular polynomial is injective over $\mathbb{Q}$ is currently out of reach, there are simple polynomials that appear to have this property. In particular, Don Zagier[2] conjectures that the polynomial $f_{\text{ZAG}}(x, y) := x^7 + 3y^7$ (the “Zagier polynomial”) defines an injective function from $\mathbb{Q}^2$ to $\mathbb{Q}$. As indirect evidence, Cornelissen [17, Remarque 10] and Poonen [39, Remark 1.7] remark that the four-variate generalization of the *abc*-conjecture [11] implies that $f(x, y) = x^e + 3y^e$ is injective over the rationals for “sufficiently large” odd integers e. Experimentally, we have confirmed that there are no rational collisions in f_{ZAG} for rationals with height less than 100.

ℓ-Variate Injective Polynomials over $\mathbb{Q}$ from Merkle-Damgård. Given a bivariate injective polynomial over $\mathbb{Q}$, it is possible to construct ℓ-variate injective polynomials over $\mathbb{Q}$ for every $\ell > 2$ using the Merkle-Damgård construction for collision-resistant hash functions [19, 30]. For example, applying one step of Merkle-Damgård to f_{ZAG} shows that if f_{ZAG} is injective then so is the following three-variate polynomial:

$$g(x, y, z) = (x^7 + 3y^7)^7 + 3z^7 \ .$$

Injective Polynomials and Collision Resistance. Proposition 9 states that, for a polynomial f to be collision resistant over $\mathbb{Z}_N$, f must be injective over the rationals. The following conjecture asserts the converse: injectivity over the rationals is *sufficient* for collision resistance.

Conjecture 10. *If $f \in \mathbb{Z}[x_1, \ldots, x_\ell]$ is injective over $\mathbb{Q}$ then f is collision resistant.*

[2] Gunther Corneliseen attributes to Don Zagier the suggestion that $f(x, y) = x^7 + 3y^7$ is collision-free over the rationals [17, Remarque 10].

This conjecture is based on the intuition that the only *efficient* way to find collisions in f over $\mathbb{Z}_N$ is to find collisions in f over $\mathbb{Q}$. Since collisions over $\mathbb{Q}$ do not exist it may be difficult to find collisions over $\mathbb{Z}_N$.

We only state Conjecture 10 to stimulate further research on this topic. The conjecture is not needed for this paper. For the applications described in this paper, we only need the collision resistance of an explicit low-degree polynomial in $\mathbb{Z}[x, y]$. Nevertheless, if Conjecture 10 is true it would give a clean characterization of collision resistant polynomials in terms of their arithmetic properties.

For the applications in paper, the following assumption suffices.

Assumption 11. *The Zagier polynomial $f_{\text{ZAG}}(x, y) = x^7 + 3y^7 \in \mathbb{Z}[x, y]$ is collision resistant.*

We see that breaking Assumption 11 would either: (a) resolve a 15-year open number theoretic problem by showing that f_{ZAG} is non-injective, or (b) find $\mathbb{Z}_N$ collisions that are not rational collisions. We next review two potential avenues for attacks of type (b) and discuss why they do not apply.

Attack Strategy I: Related Non-Injective Polynomials over $\mathbb{Q}$. One potential avenue for attacking the collision resistance of f_{ZAG} in $\mathbb{Z}_N$ is to look for a polynomial $h \in \mathbb{Z}[x, y]$ such that

$$g(x, y) := f(x, y) + N \cdot h(x, y)$$

is not injective over $\mathbb{Q}$. If (x_0, y_0) and (x_1, y_1) in $\mathbb{Q}^2$ are a rational collision for g then by reducing this pair modulo N we may[3] obtain a $\mathbb{Z}_N$ collision for $f(x, y)$. We say that h is "useful" if there exists a rational collision for $g(x, y)$ that gives a $\mathbb{Z}_N$ collision for $f(x, y)$. It is easy to show that there are many useful polynomials h: every $\mathbb{Z}_N$ collision for $f(x, y)$ gives a useful polynomial h. However, we do not know how to construct a useful h just given f and N. Furthermore, even if efficiently constructing a useful h is possible, the attack algorithm will need to find a rational collision on the resulting g and this may not be feasible in polynomial time.

Attack Strategy II: Algebraic Extensions. Another avenue for attacking the collision resistance of f_{ZAG} in $\mathbb{Z}_N$ is via algebraic extensions. Let g be an irreducible polynomial in $\mathbb{Z}[x]$ and consider the number field $\mathbb{K} = \mathbb{Q}[x]/(g)$. Suppose the adversary constructs g so that it knows an efficiently computable map $\rho : \mathbb{K} \to \mathbb{Z}_N$ (this can be done by choosing the polynomial g so that the adversary knows a zero of g in $\mathbb{Z}_N$). Now, even if f_{ZAG} is injective as a function $\mathbb{Q}^2 \to \mathbb{Q}$, it may not be injective as a function $\mathbb{K}^2 \to \mathbb{K}$. For example, f_{ZAG} is not injective over the extension $\mathbb{K} = \mathbb{Q}[\sqrt[7]{3}]$: the points $(\sqrt[7]{3}, 0)$ and $(0, 1)$ are a collision. If the adversary could find a collision of f_{ZAG} in $\mathbb{K}^2$ this collision may lead

[3] If (x_0, y_0) and (x_1, y_1) happens to reduce to the same point modulo N or if one of the denominators is not relatively prime to N then this rational collision for g does not give a $\mathbb{Z}_N$ collision for f.

to a $\mathbb{Z}_N$ collision for f_{ZAG}. However, for a random RSA modulus N, it is not known how to efficiently construct an extension $\mathbb{K}$ such that (i) $f_{\text{ZAG}} : \mathbb{K}^2 \to \mathbb{K}$ is not injective, and (ii) the adversary has an efficiently computable map $\rho : \mathbb{K} \to \mathbb{Z}_N$.

Assumption 11 merits further analysis and we hope that this work will stimulate further research on this question.

Non-Collision Resistant Polynomials. Simple variations of Zagier's polynomial are trivially not injective and therefore not collision resistant. For example, the polynomials

$$f_1(x,y) = x^7 + y^7 \quad \text{and} \quad f_2(x,y) = x^7 + 2y^7$$

in $\mathbb{Z}[x,y]$ are not collision resistant. The polynomial f_1 is not injective because for all $x_0 \neq y_0$ in $\mathbb{Z}$ the points (x_0, y_0) and (y_0, x_0) are a collision for f_1. The polynomial f_2 is not collision resistant because for all $t \neq 0$ in $\mathbb{Z}$ the points $(-t, 0)$ and $(t, -t)$ are a collision for f_2.

Similarly, polynomials of the form $f(x,y) = x^{e_1} + by^{e_2} \in \mathbb{Z}[x,y]$ for some $b \in \mathbb{Z}$ where $\gcd(e_1, e_2) = 1$ are not injective and therefore not collision resistant. To see why observe that if the equation $\alpha e_1 - \beta e_2 = 1$ has integer solutions (α_0, β_0) and (α_1, β_1) then $(t^{\alpha_0}, t^{\beta_1})$ and $(t^{\alpha_1}, t^{\beta_0})$ are a collision for f.

Random Self-Reduction. Finally, we mention that the collision finding problem for the family of polynomials $\{x^e + ay^e\}_{a \in \mathbb{Z}_N}$ has a random self reduction. Given a collision-finding algorithm $\mathcal{A}(N, a)$ that outputs a $\mathbb{Z}_N$ collision in $x^e + ay^e$ for a non-negligible fraction of choices of $a \in \mathbb{Z}_N$, it is possible to construct a collision-finding algorithm $\mathcal{B}(N, a)$ that finds collisions for *every* choice of a with high probability. On input (N, a) Algorithm $\mathcal{B}$ chooses a random $r \leftarrow \mathbb{Z}_N$, and calls $\mathcal{A}(N, r^e a)$. When $\mathcal{A}$ outputs the collision (x_0, y_0), (x_1, y_1), algorithm $\mathcal{B}$ obtains the following collision on the original curve: (x_0, ry_0), (x_1, ry_1). If $\mathcal{A}$ fails then $\mathcal{B}$ can try again with a fresh random choice of $a \in \mathbb{Z}_N$. After an expected polynomial number of iterations algorithm $\mathcal{B}$ will find a collision for the given polynomial $x^e + ay^e$.

4 A Nestable Commitment Scheme From Polynomials Over $\mathbb{Z}_N$

Having argued that it is infeasible to find collisions in the function $f_{\text{ZAG}}(x,y) = x^7 + 3y^7 \bmod N$ (Assumption 11), we now turn to the cryptographic applications of this new computational assumption. In this section, we demonstrate that the collision-resistance of f_{ZAG} leads to a commitment scheme where the procedure for verifying that a commitment was opened correctly uses only low-degree polynomials. The new commitment scheme is statistically hiding and its computational binding property is based on Assumption 11.

The commitment scheme composes naturally with zero-knowledge proofs of knowledge involving Pedersen commitments. In particular, given a Pedersen commitment C to one of our low-degree commitments, there is a succinct zero-knowledge protocol which proves knowledge of *an opening of an opening* of C. We call the inner commitment scheme *nestable*, since it can be efficiently nested inside of a Pedersen commitment. We discuss applications of nestable commitments in Sections 4.4 and 5.

4.1 Commitments

A commitment scheme is a tuple of efficient algorithms (Setup, Commit, Open), with the following functionalities:

$\mathsf{Setup}(\lambda) \rightarrow \mathsf{pp}$. The Setup routine is a randomized algorithm that runs in time polynomial λ and returns public parameters pp. These parameters define a message space $\mathcal{M}$, a space of random blinding values $\mathcal{R}$, and a space of commitments $\mathcal{C}$. The following algorithms take the public parameters pp as an implicit argument.

$\mathsf{Commit}(m) \rightarrow (c, r)$. Given a message $m \in \mathcal{M}$, return a commitment $c \in \mathcal{C}$ and a random blinding value $r \in \mathcal{R}$ used to open the commitment.

$\mathsf{Open}(c, m, r) \rightarrow \{0, 1\}$. Given a commitment c, a message m, and a blinding value r, return "1" if (m, r) is a valid opening of c and "0" otherwise.

For correctness, we require that, for all $m \in \mathcal{M}$:

$$\Pr[\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda); (c, r) \leftarrow \mathsf{Commit}(m) : \mathsf{Open}(c, m, r) = 1] \geq 1 - \mathsf{negl}(\lambda).$$

A statistically hiding commitment scheme must satisfy two security properties:

– **Statistically Hiding.** For any two messages m_0 and m_1 in $\mathcal{M}$, a commitment to m_0 is statistically indistinguishable from a commitment to m_1.
– **Computationally Binding.** For any p.p.t. adversary $\mathcal{A}$, the adversary has negligible advantage in producing two different valid openings of the same commitment. More precisely,

$$\begin{aligned}&\Pr[\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda); (c, m, r, m', r') \leftarrow \mathcal{A}(\mathsf{pp}) :\\ &\quad \mathsf{Open}(c, m, r) = 1 \land \mathsf{Open}(c, m', r') = 1 \land (m, r) \neq (m', r')] \leq \mathsf{negl}(\lambda).\end{aligned}$$

4.2 Construction

The public parameters for our new commitment scheme consist only of an RSA modulus N, for which no one knows the factorization. To commit to a value $m \in \mathbb{Z}_N^*$, the committer samples a random blinding value r from $\mathbb{Z}_N^*$ and computes the value of f_{ZAG} at the point (m, r).

The construction of the new commitment scheme follows.

$\mathsf{Setup}(\lambda) \rightarrow N$. The value N is an RSA modulus—the product of two random $\mathsf{len}(\lambda)$-bit primes p and q such that $\gcd(p-1, q-1, 7) = 1$. The commitment space $\mathcal{C}$ is $\mathbb{Z}_N$. The message space $\mathcal{M}$ and the space of blinding values $\mathcal{R}$ are $\mathbb{Z}_N^*$.

$\mathsf{Commit}(m) \rightarrow (c, r)$. Choose a random blinding value $r \leftarrow \mathbb{Z}_N^*$ and set $c \leftarrow m^7 + 3r^7$ in $\mathbb{Z}_N$. Return r as the commitment secret.

$\mathsf{Open}(c, m, r) \rightarrow \{0, 1\}$. Output "1" if $m, r \in \mathbb{Z}_N^*$ and if $c = m^7 + 3r^7$ in $\mathbb{Z}_N$. Output "0" otherwise.

Security Properties. The following theorem summarizes the security properties of the scheme.

Theorem 12. *The commitment scheme is statistically hiding and computationally binding under Assumption 11.*

Proof. Statistical hiding follows from a standard argument given in Appendix A. Computational binding follows directly from the collision resistance of f_{ZAG} over $\mathbb{Z}_N$. One issue is Setup algorithm generates a random N such that $\gcd(\phi(N), 7) = 1$ whereas Assumption 11 imposes no such restriction on N. Nevertheless, Assumption 11 implies the collision resistance of f_{ZAG} for this modified distribution of N: By way of contradiction, assume there were an algorithm $\mathcal{A}$ which finds collisions in f_{ZAG} with non-negligible probability ϵ when $\gcd(\phi(N), 7) = 1$. Since algorithm RSAgen in Assumption 11 generates such N with probability about $(5/6)^2 = 25/36$ it follows that $\mathcal{A}$ will find collisions in with probability at least $(25/36)\epsilon$ when N is sampled as in algorithm RSAgen, violating Assumption 11.

Efficiency. Generating and verifying standard Pedersen commitments requires two modular exponentiations (or elliptic curve scalar multiplications). In contrast, our scheme requires only a few modular *multiplications*. On a workstation with a 3.20 GHz processor, for example, computing 10,000 Pedersen commitments in a subgroup of order $\approx 2^{256}$ modulo a 2048-bit prime takes 16.54 seconds. Computing the same number of commitments using this new scheme takes 0.925 seconds—a factor of 17.9$\times$ speed-up.

4.3 Nestable Commitments

We say that a commitment scheme $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Open})$ is *nestable* if, given Pedersen commitments to a message m, randomness r, and a commitment c, there is an succinct zero-knowledge proof of knowledge of values m, r, and c, such that $c = \mathsf{Commit}(m, r)$. In other words, there is a succinct protocol for proving knowledge of an *opening of an opening* of a Pedersen commitment. For our purposes, a *succinct* zero-knowledge protocol is one in which proof length is $k|c|$ bits long, where k is a constant which does not depend on the security parameter.

We adopt the notation of Camenisch and Stadler [13] for specifying zero-knowledge proof-of-knowledge protocols. For example, $\mathsf{PoK}\{x, y : X = g^x \vee Y = g^x\}$ indicates a protocol in which the prover and verifier share public values g, X, and Y, and the prover demonstrates knowledge of either a value x such that $X = g^x$ *or* a value y such that $Y = g^y$.

Given Pedersen commitments

$$C_m = g^m h^{s_m} \qquad C_r = g^r h^{s_r} \qquad C_c = g^c h^{s_c}$$

a *nestable* commitment scheme has a succinct zero-knowledge protocol which proves knowledge of the statement:

$$\mathsf{PoK}\{m, r, c, s_m, s_r, s_c : C_m = g^m h^{s_m} \wedge C_r = g^r h^{s_r} \wedge C_c = g^{\mathsf{Commit}(m,r)} h^{s_c}\}.$$

For the commitment scheme outlined above, $\mathsf{Commit}(m, r) = m^7 + 3r^7 \bmod N$, so the proof of knowledge protocol is:

$$\mathsf{PoK}\{m, r, c, s_m, s_r, s_c : C_m = g^m h^{s_m} \wedge C_r = g^r h^{s_r} \wedge C_c = g^{m^7+3r^7} h^{s_c}\}.$$

The group $\mathbb{G} = \langle g \rangle = \langle h \rangle$ used for the proof must be a group of *composite* order N, where N is the RSA modulus used in the commitment scheme. As usual for Pedersen commitments, no one should know the discrete logarithm $\log_g h$ in $\mathbb{G}$. For example, $\mathbb{G}$ might be the order-N subgroup of the group $\mathbb{Z}_p^*$ for a prime $p = 2kN + 1$, where k is a small prime. Alternatively, $\mathbb{G}$ could be an elliptic curve group of order N.

The fact that the verification equation for our commitment scheme is a fixed low-degree polynomial means that this proof can be executed succinctly using standard techniques [14]. This proof requires only one challenge and 20 elements of $\mathbb{G}$. If N is a 2048-bit modulus, then the proof is roughly 5 KB in length.

In contrast, nesting Pedersen commitments inside of other Pedersen commitments does not lead to succinct proofs of knowledge. The shortest proofs of knowledge for nested Pedersen commitments require a number of group elements that is linear in the security parameter [15, Sec. 5.3.3], whereas our proof requires only a constant number of group elements.

Being able to prove knowledge of an opening of a commitment which is itself nested inside of a commitment proves useful in constructing distributed e-cash schemes (Section 4.4) and set membership proofs (Section 5).

4.4 Application Sketch: Anonymous Bitcoins

The Zerocoin scheme for anonymizing Bitcoin transactions requires a proof of knowledge of an opening of an opening of a commitment [32]. For this purpose, Zerocoin uses Pedersen commitments nested inside of Pedersen commitments, which requires a proof-of-knowledge of the form: $\mathsf{PoK}\{m, r, s : \hat{c} = \hat{g}^{(g^m h^r)} \hat{h}^s\}$. The number of group elements exchanged in this proof is linear in the security parameter, since the proof uses single-bit challenges.

By using our nestable commitment scheme for the "inner" commitment, we reduce the number of group elements from linear to constant in the security parameter. This reduces the length of anonymous coin transactions in the Zerocoin scheme by roughly 70% (down to 12.0 KiB from 39.4 KiB when using a 2048-bit RSA modulus). When instantiated with our nestable commitments, Zerocoin maintains its unconditional privacy property and maintains double-spending prevention under Assumption 11.

5 Succinct Set Membership Proofs

A *cryptographic accumulator*, first defined by Benaloh and De Mare [8], is a primitive which allows a prover to accumulate large set of values $S = \{x_1, \dots, x_n\}$ into a single short value A. For every value x_i in the accumulator, there is an accompanying short witness w_i. By exhibiting a valid (x_i, w_i) pair, a prover can convince a verifier that the value x_i was actually accumulated into A. Informally, the security property of the accumulator requires that it be difficult to find a valid value-witness pair (x^*, w^*) such that $x^* \notin S$.

Benaloh and De Mare give one example application of this primitive: the administrator of a club can accumulate the names of the members of the club into an accumulator A, distribute a witness to each member, and publish the accumulator value A. The value A is a concise representation of the club's membership list. A person can prove membership in the club by revealing her name x_i and the witness w_i to a verifier.

Camenisch and Lysyanskaya extend the basic strong-RSA accumulator primitive to allow for zero-knowledge proofs of accumulator membership [12]. That is, a prover can convince a verifier that the prover "knows" a valid value-witness pair (x, w) for a particular accumulator A, without revealing x or w. This augmented primitive allows for privacy-preserving authentication: a club member can prove that she is *some* member of the club defined by a membership list A without revealing *which* member she is.

We provide a construction that offers the same functionality as the Camenisch-Lysyanskaya scheme with the cost of requiring slightly larger proofs—of length $O(\log |S|)$ instead of length $O(1)$. The benefit of our construction is its simplicity: compared with the Camenisch-Lysyanskaya proof, which requires a nuanced security analysis, ours is relatively straightforward.

5.1 Definitions

A *cryptographic accumulator* is a tuple of algorithms (Setup, $\mathsf{Accumulate}$, $\mathsf{Witness}$, Verify) with the following functionalities:

$\mathsf{Setup}(\lambda) \to \mathsf{pp}$. Given a security parameter λ as input, output the public parameters pp. The other functions take pp as an implicit input. Setup runs in time polynomial in λ.

$\mathsf{Accumulate}(S = \{x_1, \dots, x_n\}) \to A$. Accumulate the n items in the set S into an accumulator value A.

$\mathsf{Witness}(S, x) \to w$ or $\perp$. If $x \notin S$, return $\perp$. Otherwise, return a witness w that x was accumulated in $\mathsf{Accumulate}(S)$. To be useful, the length of w should be short (constant or logarithmic) in the size of S.

$\mathsf{Verify}(A, x, w) \to \{0, 1\}$. Return "1" if the value-witness pair (x, w) is valid for the accumulator A. Return "0" otherwise.

Camenisch and Lysyanskaya, following Barić and Pfitzmann [4], define an accumulator as *secure*, if for all polynomial-time adversaries $\mathcal{A}$:

$$\Pr[\mathsf{pp} \leftarrow \mathsf{Setup}(\lambda);\ (S, x^*, w^*) \leftarrow \mathcal{A}(\mathsf{pp});\ x^* \notin S; \\ A \leftarrow \mathsf{Accumulate}(S) : \mathsf{Verify}(A, x^*, w^*) = 1] \leq \mathsf{negl}(\lambda).$$

If an accumulator satisfies this definition, then it is infeasible for an adversary to prove that a value x was accumulated in a value A if it was not.

Zero-Knowledge Proof of Knowledge of an Accumulated Value. In many applications, it is useful for a prover to be able to convince a verifier that the prover knows *some* value inside of an accumulator without revealing *which* value the prover knows. Such a proof protocol should satisfy the standard properties of soundness, completeness, and zero-knowledgeness [15, Sec. 2.9]. Camenisch and Lysyanskaya construct one such proof-of-knowledge protocol for the strong-RSA accumulator [12] and we exhibit a protocol for a Merkle-tree-style accumulator in Section 5.3.

5.2 Construction

Given a collision-resistant hash function $H : D \times D \to D$, which operates on a domain D such that $S \subseteq D$, it is possible to construct a simple accumulator using Merkle trees. For example, given a set $S = \{x_1, x_2, x_3, x_4\}$, the accumulator value A is the value $A \leftarrow H(H(x_1, x_2), H(x_3, x_4))$. A witness w_i that an element x_i is in the accumulator is the set of $O(\log |S|)$ nodes along the Merkle tree needed to verify a path from x_i to the root (labeled A).

The limitation of this accumulator construction is that it no longer admits simple zero-knowledge proofs of knowledge of (x, w) pairs, unless H has a very special form. For instance, if H is a standard cryptographic hash function (e.g., SHA-256), there is no straightforward zero-knowledge protocol for proving knowledge in zero knowledge of a preimage under H. By instantiating H with the function $H(x, y) = x^7 + 3r^7 \bmod N$, as we demonstrate in the following section, it is possible to execute this zero-knowledge proof succinctly.

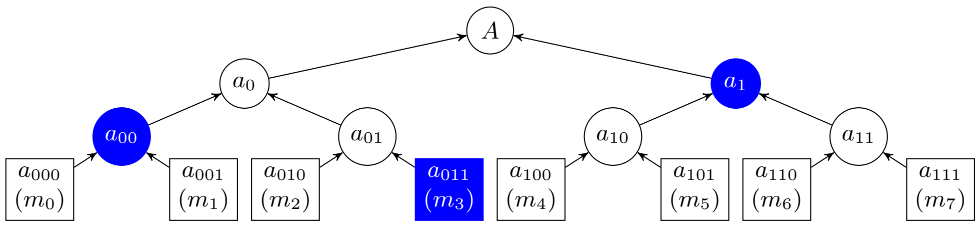

Fig. 1. A perfect Merkle tree with eight leaves rooted at A. The shaded nodes are a witness to the fact that m_2 is accumulated in A. The tree invariant is $a_i = H(a_{i0}, a_{i1})$.

We first recall the standard construction of Merkle trees [29] and then describe the zero-knowledge proof construction. The construction from a general collision-resistant hash function family $\{\mathcal{H}_\lambda\}_{\lambda=1}^{\infty}$ follows.

$\mathsf{Setup}(\lambda) \to H$. Given a security parameter λ as input, sample a λ-secure collision-resistant hash function H from $\mathcal{H}_\lambda$. Setup runs in time polynomial in λ.

$\mathsf{Accumulate}(S = \{x_1, \dots, x_n\}) \to A$. If $|S|$ is not a power of two, insert "dummy" elements into S (e.g., by duplicating the first element of S) until $|S|$ is a power of two. Construct a perfect Merkle tree of depth $d = \log_2 |S|$ using the hash function H with the members of S as its leaves and return the root A. Figure 1 depicts an example tree of depth three.

$\mathsf{Witness}(S, x) \to w$ or $\perp$. If $x \notin S$, return $\perp$. Otherwise, let the path from A to the message x be: $P = (A, a_{b_1}, a_{b_1 b_2}, a_{b_1 b_2 b_3}, \ldots, a_{b_1 \ldots b_d})$, where a_{i0} is the left child of node a_i, a_{i1} is the right child of node a_i, and d is the number of edges between the root and leaf labeled x in the tree. The first component of the witness is the list of siblings of the nodes in the path P: $w_\alpha = (a_{\bar{b}_1}, a_{b_1 \bar{b}_2}, a_{b_1 b_2 \bar{b}_3}, \ldots, a_{b_1 \ldots \bar{b}_d})$. The second component of the witness is a bit vector indicating where x is located in the tree: $w_\beta = (b_1, b_2, \ldots, b_{d-1}, b_d)$. The witness is $w = (w_\alpha, w_\beta)$.

$\mathsf{Verify}(A, x, w) \to \{0, 1\}$. Interpret the witness as (w_α, w_β) such that $w_\alpha = (w_1, \ldots, w_d)$ and $w_\beta = (b_1, \ldots, b_d)$. To verify the witness, let $t_d = x$ and recompute the intermediate nodes of the tree from the leaf back to the root. Specifically, compute test nodes t_i for $i = d - 1, \ldots, 0$:

$$t_i = \begin{cases} H(t_{i+1}, w_{i+1}) : \text{if } b_i = 0 \\ H(w_{i+1}, t_{i+1}) : \text{if } b_i = 1 \end{cases}$$

Return "1" if $A = t_0$ and "0" otherwise.

5.3 Proof of Knowledge of an Accumulated Value

When instantiated with a general hash function H, the Merkle-tree accumulator of the prior section does not admit a succinct proof of knowledge of an accumulated value. When instantiated with our new hash function $H(x, y) = x^7 + 3y^7 \bmod N$, however, there *is* a succinct proof of knowledge that the prover knows an opening of a Pedersen commitment C_m such that some leaf of the accumulator Merkle tree has label m. The proof requires a group $\mathbb{G} = \langle g \rangle = \langle h \rangle$ of order N, as in Section 4.3. The proof length is $\log |S|$, for a set S of elements accumulated.

The Setup algorithm outputs an RSA modulus $N \leftarrow \mathsf{RSAgen}(\lambda)$ such that $\gcd(\phi(N), 7) = 1$ and such that no one knows the factorization of N. The hash function H is $H(x, y) = x^7 + 3y^7 \bmod N$ and the accumulator domain D is $\mathbb{Z}_N^*$.

The high-level idea is that, if the prover wants to convince the verifier that a particular value m is accumulated in A, the prover commits to the values of all of the nodes in the Merkle tree along the path from the root to the leaf labeled m. The prover also commits to all of the witness values needed to recreate the path from the leaf labeled m down to the tree root. The prover can then convince the verifier in zero knowledge that these commitments together contain a path to *some* leaf in the tree, without revealing which one.

Assume that the prover has a value-witness pair (x, w) which convinces a verifier that x is accumulated in A. Denote the node values along the path from the root node, with value A, to the leaf node, with value x, in the Merkle tree as: $p = (p_0, p_1, \ldots, p_d)$. Note that $p_0 = A$ and $p_d = x$.

The prover now commits to every value p_i in this path *and* to the values of the left and right children of p_i in the Merkle tree. If the value of the left child is ℓ_i and the right child is r_i, the commitments are, for $i = 0, \ldots, d-1$:

$$P_i = g^{p_i} h^{s_i} \qquad L_i = g^{\ell_i} h^{s'_i} \qquad R_i = g^{r_i} h^{s''_i}$$

The prover opens P_0 by publishing (p_0, s_0) and the verifier ensures that $p_0 = A$ and that $P_0 = g^{p_0} h^{s_0}$.

The prover now can prove, for $i = 0, \ldots, d-1$, that each (P_i, L_i, R_i) tuple is well-formed using a standard discrete logarithm proof:

$$\mathsf{PoK}_\alpha\{\ell, r, s, s', s'' : P_i = g^{\ell^7 + 3r^7} h^s \wedge L_i = g^\ell h^{s'} \wedge R_i = g^r h^{s''}\}.$$

The prover then must prove that it knows an opening of the commitment P_{i+1} such that the opening is equal to an opening of either L_i or R_i. For $i = 0, \ldots, d-1$, the prover proves:

$$\mathsf{PoK}_\beta\{p, s, s_\ell, s_r : P_{i+1} = g^p h^s \wedge (L_i = g^p h^{s_\ell} \vee R_i = g^p h^{s_r})\}.$$

The complete proof is the set of commitment pairs $\{(P_i, L_i, R_i)\}_{i=0}^d$, the $2d$ proofs of knowledge, and the opening (p_0, r_0) of the root commitment P_0. The total length is $O(d) = O(\log |S|)$, since the tree has depth $d = \log |S|$ and each of the elements of the proof has length which is constant in $|S|$.

Security. The completeness and zero-knowledgeness properties follows from the properties of the underlying zero-knowledge proofs used and from the fact that Pedersen commitments are perfectly hiding.

To show soundness, we must demonstrate that if the verifier accepts, it can extract a value-witness pair (x^*, w^*) for the original Merkle tree with non-negligible probability by rewinding the prover. Starting at the root and working towards the leaves of the tree, we will be able to extract the prover's witness for each of the proofs of knowledge with non-negligible probability.

By induction on i, we can show that after d steps, the verifier will be able to extract the value-witness pair (x, w). The base case of the induction is $i = 0$ and the verifier can extract a preimage of A under H. From each of the i PoK_αs, the verifier extracts an element of the witness w_α (the preimage of p_i under H). From each of the i PoK_βs, the verifier extracts an element of the witness w_β (whether the next node in the path is the left or right child of p_i).

6 Claw-Free Functions, Signatures, and Chameleon Hashes

In this section, we describe a few other applications arising from the assumed collision-freeness of the Zagier polynomial.

Claw-Free Functions and Signatures. Assumption 11 immediately gives rise to a family of *trapdoor claw-free functions* [18]. For each RSA modulus N selected as in Section 4.2, we can define a function family:

$$\mathcal{F}_N := \{f_a \mid a \in \mathbb{Z}_N^*\} \quad \text{where} \quad f_a(x) = x^7 + 3a^7 \bmod N.$$

Following Damgård [18], a function family $\mathcal{F}_N$ is *claw free* if, given $\mathcal{F}_N$, it is difficult to find a "claw" (x, y, a, b) such that $f_a(x) = f_b(y)$. For all p.p.t. adversaries $\mathcal{A}$, we require that:

$$\Pr[\ N \leftarrow \mathsf{RSAgen}(\lambda),\ (x, y, a, b) \leftarrow \mathcal{A}(N)\ :\ f_a(x) = f_b(y)\ ] \leq \mathsf{negl}(\lambda).$$

The claw-freeness of $\mathcal{F}_N$ follows from Assumption 11, since a claw in $\mathcal{F}_N$ implies a collision in $f(x, y) = x^7 + 3y^7 \bmod N$. Additionally, the function family $\mathcal{F}_N$ is trapdoor claw-free, since anyone with knowledge of the factors of N can find claws easily by choosing (x, y, a) arbitrarily and solving for b.

This family $\mathcal{F}_N$ is not quite a family of trapdoor claw-free permutations, since the range of two functions f_a and f_b in $\mathcal{F}_N$ are not necessarily equal (i.e., $f_b^{-1}(f_a(x))$ is sometimes undefined). However, the fraction of choices of (a, b, x) for which this event occurs is negligible, so it is possible to treat $\mathcal{F}_N$ as if it were a family of trapdoor claw-free permutations. In particular, this function family leads to a signature scheme secure against adaptive chosen message attacks in the standard model by way of the Goldwasser-Micali-Rivest signature construction [22].

Chameleon Hash. This commitment scheme immediately gives rise to a new chameleon hash function. A chameleon hash, as defined by Krawczyk and Rabin, is a public hash function $H(m, r)$ with a secret "trapdoor" [26]. A chameleon hash function has three properties:

1. Without the trapdoor, it is difficult to find collisions in H. That is, it is hard to find colliding pairs (m, r) and (m', r') such that $H(m, r) = H(m', r')$.
2. Given the trapdoor, there is an efficient algorithm which takes (m, r, m') as input and outputs a value r' such that $H(m, r) = H(m', r')$.
3. For any pair of messages m and m' in the message space $\mathcal{M}$, the distributions $H(m, r)$ and $H(m', r')$ are statistically close if r and r' are chosen at random.

Chameleon hashes are useful in building secure signature schemes in the standard model [21], converting any signature scheme into an online/offline signature scheme [43], converting weakly unforgeable signatures schemes into strongly unforgeable ones [10, 44], and for a number of other applications [26]. Bellare and Ristov [5] demonstrate that any sigma protocol gives rise to a chameleon hash.

To derive a chameleon hash scheme from our commitment scheme, set the public key to the RSA modulus N, and the secret key to the factorization of N. The hash function H is then $H(m, r) = m^7 + 3r^7 \bmod N$. Without the factors of N, it is difficult to find collisions but anyone with knowledge of the factors of N (the "trapdoor") can find collisions.

Chameleon hashes based on Pedersen commitments require two modular exponentiations to evaluate, while ours requires just a few modular multiplications. Our chameleon hash requires a new computational assumption (Assumption 11), but outperforms the schemes surveyed in prior work [5].

7 Conclusion and Future Work

We have used arithmetic properties of bivariate polynomials over $\mathbb{Q}$ to reason about their cryptographic properties in the ring $\mathbb{Z}_N$. Using one particular low-degree polynomial, f_{ZAG}, we build a new statistically hiding commitment scheme, a conceptually simple cryptographic accumulator, and a computationally efficient chameleon hash function. To gain confidence in Conjecture 10 it would be interesting to prove it in the *generic ring* model [1]. We leave that for future work.

Acknowledgments. We are grateful to Bjorn Poonen for information about injective polynomials, to Steven Galbraith and Antoine Joux for comments on our cryptographic assumptions, and to Don Zagier for recounting his rationale for conjecturing the injectivity of $f(x, y) = x^7 + 3y^7$ over $\mathbb{Q}$. We thank Joe Zimmerman for helpful conversations about this work. This work was supported by DARPA, an NSF research grant, and an NSF Graduate Research Fellowship under Grant No. DGE-114747.

References

1. Aggarwal, D., Maurer, U.: Breaking RSA generically is equivalent to factoring. In: EUROCRYPT. pp. 36–53 (2009)
2. Ash, A., Gross, R.: Elliptic Tales: Curves, Counting, and Number Theory. Princeton University Press (2012)
3. Ateniese, G., de Medeiros, B.: Identity-based chameleon hash and applications. In: Financial Cryptography. pp. 164–180 (2004)
4. Barić, N., Pfitzmann, B.: Collision-free accumulators and fail-stop signature schemes without trees. In: EUROCRYPT. pp. 480–494 (1997)
5. Bellare, M., Ristov, T.: Hash functions from sigma protocols and improvements to VSH. In: Proceedings of ASIACRYPT 2008. pp. 125–142 (2008)
6. Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., Virza, M.: Zerocash: Decentralized anonymous payments from Bitcoin. In: IEEE Security and Privacy (2014)
7. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Succinct non-interactive zero knowledge for a von Neumann architecture. In: USENIX Security Symposium (2014)
8. Benaloh, J., De Mare, M.: One-way accumulators: A decentralized alternative to digital signatures. In: EUROCRYPT. pp. 274–285 (1993)
9. Boneh, D., Gentry, C., Hamburg, M.: Space-efficient identity based encryption without pairings. In: FOCS. pp. 647–657 (2007)
10. Boneh, D., Shen, E., Waters, B.: Strongly unforgeable signatures based on computational Diffie-Hellman. In: PKC 2006, pp. 229–240 (2006)

11. Browkin, J., Brzeziński, J.: Some remarks on the *abc*-conjecture. Mathematics of Computation 62(206), 931–939 (1994)
12. Camenisch, J., Lysyanskaya, A.: Dynamic accumulators and application to efficient revocation of anonymous credentials. In: CRYPTO. pp. 61–76 (2002)
13. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups. In: CRYPTO. pp. 410–424 (1997)
14. Camenisch, J., Stadler, M.: Proof systems for general statements about discrete logarithms. Tech. Rep. 260, Dept. of Computer Science, ETH Zurich (Mar 1997)
15. Camenisch, J.L.: Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem. Ph.D. thesis, Swiss Federal Institute of Technology Zürich (ETH Zürich) (1998)
16. Catalano, D., Gennaro, R., Howgrave-Graham, N., Nguyen, P.Q.: Paillier's cryptosystem revisited. In: ACM conference on Computer and Communications Security. pp. 206–214 (2001)
17. Cornelissen, G.: Stockage diophantien et hypothese abc généralisée. Comptes Rendus de l'Académie des Sciences-Series I-Mathematics 328(1), 3–8 (1999)
18. Damgård, I.B.: The Application of Claw Free Functions in Cryptography. Ph.D. thesis, Aarhus University (May 1988)
19. Damgård, I.B.: A design principle for hash functions. In: CRYPTO. pp. 416–427 (1989)
20. Dodis, Y., Kiayias, A., Nicolosi, A., Shoup, V.: Anonymous identification in ad hoc groups. In: EUROCRYPT 2004. pp. 609–626 (2004)
21. Gennaro, R., Halevi, S., Rabin, T.: Secure hash-and-sign signatures without the random oracle. In: EUROCRYPT. pp. 123–139 (1999)
22. Goldwasser, S., Micali, S., Rivest, R.L.: A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing 17(2), 281–308 (1988)
23. Groth, J.: Short pairing-based non-interactive zero-knowledge arguments. In: ASIACRYPT 2010, pp. 321–340 (2010)
24. Hindry, M., Silverman, J.H.: Diophantine geometry: an introduction, vol. 201. Springer (2000)
25. Kilian, J., Petrank, E.: Identity escrow. In: CRYPTO. pp. 169–185 (1998)
26. Krawczyk, H., Rabin, T.: Chameleon hashing and signatures. In: NDSS. pp. 143–154 (2000)
27. Lin, Z., Hopper, N.: Jack: Scalable accumulator-based Nymble system. In: Workshop on Privacy in the Electronic Society. pp. 53–62. ACM (2010)
28. Lipmaa, H.: Secure accumulators from euclidean rings without trusted setup. In: Applied Cryptography and Network Security. pp. 224–240 (2012)
29. Merkle, R.C.: A digital signature based on a conventional encryption function. In: CRYPTO. pp. 369–378 (1987)
30. Merkle, R.C.: One way hash functions and DES. In: CRYPTO. pp. 428–446 (1989)
31. Micali, S., Rabin, M., Kilian, J.: Zero-knowledge sets. In: FOCS. pp. 80–91 (2003)
32. Miers, I., Garman, C., Green, M., Rubin, A.D.: Zerocoin: Anonymous distributed e-cash from Bitcoin. In: IEEE Security and Privacy. pp. 397–411 (2013)
33. Naor, M.: On cryptographic assumptions and challenges. In: CRYPTO. pp. 96–109 (2003)
34. Nguyen, L.: Accumulators from bilinear pairings and applications. In: CT-RSA 2005, pp. 275–292 (2005)
35. Ong, H., Schnorr, C.P., Shamir, A.: An efficient signature scheme based on quadratic equations. In: STOC. pp. 208–216 (1984)

36. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: CRYPTO. pp. 129–140 (1991)
37. Pollard, J., Schnorr, C.: An efficient solution of the congruence. Information Theory, IEEE Transactions on 33(5), 702–709 (1987)
38. Poonen, B.: Varieties without extra automorphisms III: hypersurfaces. Finite fields and their applications 11(2), 230–268 (2005)
39. Poonen, B.: Multivariable polynomial injections on rational numbers. arXiv preprint arXiv:0902.3961v2 (Jun 2010)
40. Schwenk, J., Eisfeld, J.: Public key encryption and signature schemes based on polynomials over $\mathbb{Z}_n$. In: EUROCRYPT. pp. 60–71 (1996)
41. Shallit, J.: An exposition of Pollard's algorithm for quadratic congruences (Oct 1984)
42. Shamir, A.: On the generation of multivariate polynomials which are hard to factor. In: STOC. pp. 796–804. ACM (1993)
43. Shamir, A., Tauman, Y.: Improved online/offline signature schemes. In: Proceedings of CRYPTO 2001. pp. 355–367 (2001)
44. Steinfeld, R., Pieprzyk, J., Wang, H.: How to strengthen any weakly unforgeable signature into a strongly unforgeable signature. In: CT-RSA 2007, pp. 357–371 (2006)
45. Zagier, D.: Personal communication (Jun 2014)

A Proof of Statistical Hiding

This appendix presents a proof that the commitment scheme of Section 4.2 is statistically hiding. To demonstrate that the statistical hiding property holds, we show that for *any* message $m \in \mathbb{Z}_N^*$, the distribution of the value of a commitment c to m is statistically close to uniform.

The commitment c is generated by sampling a random value $r \leftarrow_R \mathbb{Z}_N^*$ and letting $c \leftarrow m^7 + 3r^7$. Since $r \in \mathbb{Z}_N^*$, and since $\gcd(7, \phi(N)) = 1$, the RSA function $f(x) = x^7 \bmod N$ defines a permutation on $\mathbb{Z}_N^*$. Thus, there are exactly $|\mathbb{Z}_N^*| = \phi(N)$ possible commitments to m, and each of these values occurs with equal probability.

Let the random variable C take on the value of the commitment to m and let U be a random variable uniformly distributed over $\mathbb{Z}_N$. Then:

$$\Pr[C = c_0] = \frac{1}{\phi(N)} \qquad ; \qquad \Pr[U = c_0] = \frac{1}{N}$$

The statistical distance between these distributions is:

$$\begin{aligned} \Delta(C,U) &= \frac{1}{2} \sum_{c_0 \in \mathbb{Z}_N} |\Pr[C = c_0] - \Pr[U = c_0]| \\ &= \frac{1}{2} \sum_{c_0 \in \mathbb{Z}_N} \left| \frac{N - \phi(N)}{N\phi(N)} \right| = \frac{(p+q-1)}{2\phi(N)} \leq \mathsf{negl}(\lambda). \end{aligned}$$