AICR · NOTE

The Uber Assumption and its Random Self-Reducibility

in Non-Bilinear and Type 1, 2, 3 Bilinear Groups

Abstract

We fix notation for prime-order groups without a pairing and for bilinear groups in each of the three Galbraith–Paterson–Smart types, formally define the Uber assumption in all four settings, and treat random self-reducibility in each. A single generic-computability lemma isolates, per setting, the space of exponents a reduction can realize; a single affine-reparametrization theorem then yields random self-reducibility whenever the source data are stable under the reparametrization and the challenge polynomial transforms covariantly. The four settings differ only through the product structure the pairing (and, in Type 2, the homomorphism ψ) makes available, and this difference is exactly what enlarges or shrinks the self-reducible subclass. We give a worked self-reducible assumption in each setting, show that no monomial target can ever separate two settings, and exhibit a non-monomial target that is self-reducible in Type 2 but for which, in Type 3, the affine reduction fails at every worst-case point — the largest the failure locus can be, since the admissible reparametrizations form a group and the failure fraction is therefore either $O(1/p)$ or exactly one. Because the admissibility conditions are polynomial conditions on the reparametrization, the search for one is mechanical, and we give a decision procedure that returns the maximal admissible group and either certifies the reduction or exhibits the locus of worst-case points at which it fails. The procedure needs nothing bespoke: the admissible group is the stabilizer of a subspace arrangement and of a point of a projective space, and the failure locus is the determinantal variety of the infinitesimal action, so the whole of it is standard computational algebra — a Gröbner basis, a Jacobian kernel, an ideal of maximal minors, a primary decomposition. A further section places the theorem against classical random self-reducibility. The affine condition is the one-query, linear case of the non-adaptive condition of Feigenbaum and Fortnow; Lipton's multi-query line interpolation, which the group encoding admits unchanged, solves the computational problem for every target here, including the separating one, but only against a high-advantage solver and never the decisional problem. The classical decoding machinery that would soften its threshold is not linear in the received word and so does not survive the encoding, which is why the separation stands in the decisional, negligible-advantage regime that cryptography works in.

1 Introduction

The discrete-logarithm problem is random self-reducible by the shift $g^x \mapsto g^x g^r$ [TW87]; the answer for a worst-case instance is recovered from an average-case solver by subtracting r. This note carries the underlying idea — injecting a uniform reparametrization of the secret and undoing it on the answer — across the standard group settings, and states it uniformly. Throughout, p is a prime and all groups have order p. We use implicit (bracket) notation in the style of [EHKRV13]. Section 10 relates the resulting condition to the classical form of the same idea, in which the undoing is done by

polynomial interpolation across several correlated queries rather than in closed form on one [Lipton91, FF93].

2 Group settings

2.1 Non-bilinear groups

Definition 1 (prime-order group). A *(non-bilinear) group generator* outputs $\mathcal{G} = (p, \mathbb{G}, g)$ where $\mathbb{G}$ is a cyclic group of prime order p, g a generator, and the group law and equality are computable in probabilistic polynomial time. For $a \in \mathbb{Z}_p$ write $[a] \coloneqq g^a$, so $[a] \cdot [b] = [a+b]$ and $[a]^c = [ca]$. There is no pairing.

2.2 Bilinear groups and the GPS taxonomy

Definition 2 (bilinear group). A *bilinear group generator* outputs $\mathcal{BG} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$ where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic of prime order p, g_1, g_2 generate $\mathbb{G}_1, \mathbb{G}_2$, all group laws and equalities are efficient, and $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable, non-degenerate bilinear map. For $a \in \mathbb{Z}_p$ write $[a]_1 \coloneqq g_1^a$, $[a]_2 \coloneqq g_2^a$, and $[a]_T \coloneqq e(g_1, g_2)^a$, so that

$$e([a]_1, [b]_2) = [ab]_T, \qquad [a]_i \cdot [b]_i = [a+b]_i, \qquad [a]_i^c = [ca]_i.$$

Following Galbraith, Paterson, and Smart [GPS08], bilinear groups come in three types, distinguished by the efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$:

Type 1. $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$ and e is symmetric, $e(u, v) = e(v, u)$. One source group.

Type 2. $\mathbb{G}_1 \neq \mathbb{G}_2$, with an efficiently computable isomorphism $\psi\colon \mathbb{G}_2 \to \mathbb{G}_1$ (so $\psi([a]_2) = [a]_1$), but none known in the reverse direction.

Type 3. $\mathbb{G}_1 \neq \mathbb{G}_2$, with no efficiently computable isomorphism in either direction.

Type 3 is the most efficient and most used in practice; Type 1 is the most permissive algebraically. The only structural difference for what follows is which group elements can be moved between $\mathbb{G}_1$ and $\mathbb{G}_2$, and hence which products the pairing can form.

3 The generic-computability calculus

Fix a setting and an instance consisting of encodings of finite polynomial tuples $R \subseteq \mathbb{Z}_p[\vec{X}]$ in $\mathbb{G}_1$, $S \subseteq \mathbb{Z}_p[\vec{X}]$ in $\mathbb{G}_2$, and $T \subseteq \mathbb{Z}_p[\vec{X}]$ in $\mathbb{G}_T$ (in a single-group setting only R and, if applicable, T occur). Each tuple is assumed to contain the constant 1, since the generators are public. For $\vec{X} = (X_1, \ldots, X_n)$ and a secret $\vec{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_p^n$, the instance is the collection of $[r_i(\vec{x})]_1$, $[s_j(\vec{x})]_2$, $[t_k(\vec{x})]_T$.

For a group $\gamma \in \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$ (or $\mathbb{G}$) let $\mathcal{L}_\gamma \subseteq \mathbb{Z}_p[\vec{X}]$ denote the polynomials q for which $[q(\vec{x})]_\gamma$ is computable in probabilistic polynomial time from the instance and

public parameters, uniformly in $\vec{x}$. Write $A \cdot B = \{ab : a \in A, b \in B\}$ and let $\langle\cdot\rangle$ be $\mathbb{Z}_p$-linear span.

Lemma 1 (computable exponents). *With the conventions above,*

$$\begin{aligned}
\textit{non-bilinear:}\quad & \mathcal{L}_{\mathbb{G}} = \langle R\rangle;\\
\textit{Type 1}\ (\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G},\ \textit{source tuple } R):\quad & \mathcal{L}_{\mathbb{G}} = \langle R\rangle,\quad \mathcal{L}_{\mathbb{G}_T} = \langle R\cdot R \ \cup\ T\rangle;\\
\textit{Type 2}\ (\psi : \mathbb{G}_2 \to \mathbb{G}_1):\quad & \mathcal{L}_1 = \langle R \cup S\rangle,\quad \mathcal{L}_2 = \langle S\rangle,\\
& \mathcal{L}_T = \langle (R\cup S)\cdot S \ \cup\ T\rangle;\\
\textit{Type 3:}\quad & \mathcal{L}_1 = \langle R\rangle,\quad \mathcal{L}_2 = \langle S\rangle,\quad \mathcal{L}_T = \langle R\cdot S \ \cup\ T\rangle.
\end{aligned}$$

The inclusions "$\supseteq$" hold unconditionally: every listed polynomial's encoding is computable by group operations and pairings. The reverse inclusions hold in the generic bilinear group model, where these are exactly the computable encodings [BBG05, Boyen08].

Justification. The group law realizes all $\mathbb{Z}_p$-linear combinations, giving $\langle\cdot\rangle$ in each group. The pairing sends a computable $\mathbb{G}_1$-exponent $a \in \mathcal{L}_1$ and a computable $\mathbb{G}_2$-exponent $b \in \mathcal{L}_2$ to $ab \in \mathcal{L}_T$; pairing against the generator ($b = 1$ or $a = 1$) shows $\mathcal{L}_1, \mathcal{L}_2 \subseteq \mathcal{L}_T$. In Type 1 both pairing arguments range over $\langle R\rangle$, producing all products $R \cdot R$. In Type 2, ψ moves each $\mathbb{G}_2$ encoding into $\mathbb{G}_1$, so $\mathcal{L}_1$ gains S, and the pairable products become $\mathcal{L}_1 \cdot \mathcal{L}_2 = (R \cup S)\cdot S$; the missing products $R \cdot R$ would require an element of $\mathbb{G}_2$ with an R-exponent, which ψ cannot supply. In Type 3 neither group maps to the other, so $\mathcal{L}_1 = \langle R\rangle$, $\mathcal{L}_2 = \langle S\rangle$, and only cross products $R \cdot S$ arise. The generic-model converse is the master lemma of [BBG05, Boyen08]. □

The single distinction across the four rows is the set of $\mathbb{G}_T$-products: none (no pairing), $R \cdot R$ (Type 1), $R \cdot S \cup S \cdot S$ (Type 2), $R \cdot S$ (Type 3). Everything below is a consequence of this.

4 The Uber assumption

Definition 3 (bilinear Uber problem). Fix a bilinear setting, source tuples R (in $\mathbb{G}_1$), S (in $\mathbb{G}_2$), T (in $\mathbb{G}_T$), each containing 1, a challenge polynomial $f \in \mathbb{Z}_p[\vec{X}]$, and a challenge group $\mathrm{ch} \in \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$; write $\mathcal{L}_\star := \mathcal{L}_{\mathrm{ch}}$ (Lemma 1). For a secret $\vec{x} \in \mathbb{Z}_p^n$ the instance is

$$\mathsf{Inst}(\vec{x}) = \big(\{[r_i(\vec{x})]_1\}_i,\ \{[s_j(\vec{x})]_2\}_j,\ \{[t_k(\vec{x})]_T\}_k\big).$$

- *Computational* $(R, S, T, f; \mathrm{ch})$-Uber: given $\mathsf{Inst}(\vec{x})$, output $[f(\vec{x})]_\star$.
- *Decisional* $(R, S, T, f; \mathrm{ch})$-Uber: given $\mathsf{Inst}(\vec{x})$ and Z, decide whether $Z = [f(\vec{x})]_\star$ (real) or $Z \leftarrow_{\$} \mathrm{ch}$ (random).

The *average case* draws $\vec{x} \leftarrow_{\$} \mathbb{Z}_p^n$; the *worst case* quantifies over all $\vec{x}$. For a decision algorithm $\mathcal{A}$ put

$$\delta(\mathcal{A}; \vec{x}) = \Pr\big[\mathcal{A}(\mathsf{Inst}(\vec{x}), [f(\vec{x})]_\star) = 1\big] - \Pr\big[\mathcal{A}(\mathsf{Inst}(\vec{x}), U) = 1\big], \quad U \leftarrow_{\$} \mathrm{ch},$$

and $\delta_{\mathrm{avg}}(\mathcal{A}) = \mathbb{E}_{\vec{x}\leftarrow_{\$}\mathbb{Z}_p^n}[\delta(\mathcal{A};\vec{x})]$; for a search algorithm put $\mathsf{Succ}(\mathcal{A};\vec{x}) = \Pr[\mathcal{A}(\mathsf{Inst}(\vec{x})) = [f(\vec{x})]_\star]$ and $\mathsf{Succ}_{\mathrm{avg}}(\mathcal{A}) = \mathbb{E}_{\vec{x}}[\mathsf{Succ}(\mathcal{A};\vec{x})]$. The assumption asserts that δ_{avg} (resp. $\mathsf{Succ}_{\mathrm{avg}}$) is negligible for all probabilistic polynomial-time $\mathcal{A}$.

The four settings are Definition 3 with the computable spaces of Lemma 1. We record each explicitly.

Definition 4 (non-bilinear Uber). Setting $\mathcal{G} = (p, \mathbb{G}, g)$; source tuple R (in $\mathbb{G}$, containing 1); challenge $f \in \mathbb{Z}_p[\vec{X}]$ in $\mathbb{G}$. Instance $\mathsf{Inst}(\vec{x}) = \{[r_i(\vec{x})]\}_i$; goal, compute or distinguish $[f(\vec{x})]$. Computable space $\mathcal{L}_\star = \mathcal{L}_{\mathbb{G}} = \langle R\rangle$; no pairing, hence no products.

Definition 5 (Type-1 Uber). Setting $\mathcal{BG}$ of Type 1, $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$; source tuple R (in $\mathbb{G}$), target tuple T (in $\mathbb{G}_T$), each containing 1; challenge f in $\mathrm{ch} \in \{\mathbb{G}, \mathbb{G}_T\}$. Computable spaces $\mathcal{L}_{\mathbb{G}} = \langle R\rangle$ and $\mathcal{L}_{\mathbb{G}_T} = \langle R \cdot R \cup T\rangle$.

Definition 6 (Type-2 Uber). Setting $\mathcal{BG}$ of Type 2 with $\psi\colon \mathbb{G}_2 \to \mathbb{G}_1$; source tuples R (in $\mathbb{G}_1$), S (in $\mathbb{G}_2$), T (in $\mathbb{G}_T$); challenge f in $\mathrm{ch} \in \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$. Computable spaces $\mathcal{L}_1 = \langle R \cup S\rangle$, $\mathcal{L}_2 = \langle S\rangle$, $\mathcal{L}_T = \langle (R \cup S) \cdot S \cup T\rangle$.

Definition 7 (Type-3 Uber). Setting $\mathcal{BG}$ of Type 3; source tuples R (in $\mathbb{G}_1$), S (in $\mathbb{G}_2$), T (in $\mathbb{G}_T$); challenge f in $\mathrm{ch} \in \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\}$. Computable spaces $\mathcal{L}_1 = \langle R\rangle$, $\mathcal{L}_2 = \langle S\rangle$, $\mathcal{L}_T = \langle R \cdot S \cup T\rangle$.

Remark 1 (hardness). The master theorem of [BBG05, Boyen08] states, informally, that the decisional $(R, S, T, f; \mathrm{ch})$-Uber problem is hard in the generic bilinear group model if and only if $f \notin \mathcal{L}_\star$, that is, the challenge exponent is linearly independent of the computable ones. Random self-reducibility is a separate property and does not require hardness; the two interact through Lemma 1, as discussed in Section 12.

5 An affine random-self-reducibility theorem

Let $\mathrm{AGL}_n(\mathbb{Z}_p)$ be the affine group of maps $\tau_{A,\vec{b}}\colon \vec{x} \mapsto A\vec{x} + \vec{b}$ with $A \in \mathrm{GL}_n(\mathbb{Z}_p)$, acting on polynomials by $(q \circ \tau)(\vec{X}) = q(A\vec{X} + \vec{b})$. The following theorem is setting-agnostic: it refers only to the computable spaces of Lemma 1.

Theorem 1 (master reduction). *Consider any of the four settings, an instance with source tuples R, S, T and challenge $(f; \mathrm{ch})$, and computable spaces $\{\mathcal{L}_\gamma\}$ with $\mathcal{L}_\star = \mathcal{L}_{\mathrm{ch}}$. Suppose there is an efficiently sampleable distribution $\mathcal{D}$ on a subgroup $\mathcal{T} \le \mathrm{AGL}_n(\mathbb{Z}_p)$ such that, for every $\tau \in \mathcal{T}$,*

(C1) *$r_i \circ \tau \in \mathcal{L}_1$, $s_j \circ \tau \in \mathcal{L}_2$, and $t_k \circ \tau \in \mathcal{L}_T$ for all i, j, k (with $\mathcal{L}_1 = \mathcal{L}_{\mathbb{G}}$ in the single-group settings);*

(C2) *$f \circ \tau = \lambda_\tau f + h_\tau$ for a scalar $\lambda_\tau \in \mathbb{Z}_p^\times$ and a polynomial $h_\tau \in \mathcal{L}_\star$, both computable from τ and the parameters;*

and, moreover,

(C3) *$\tau(\vec{x})$ is uniform on $\mathbb{Z}_p^n$ for every fixed $\vec{x}$ when $\tau \leftarrow \mathcal{D}$.*

Then the computational and decisional $(R, S, T, f; \mathrm{ch})$-Uber problems are random self-reducible: there is a probabilistic polynomial-time reduction $\mathcal{B}^{\mathcal{A}}$, using one draw from $\mathcal{D}$ and one oracle call, with $\mathsf{Succ}(\mathcal{B}; \vec{x}_0) = \mathsf{Succ}_{\mathrm{avg}}(\mathcal{A})$ and $\delta(\mathcal{B}; \vec{x}_0) = \delta_{\mathrm{avg}}(\mathcal{A})$ for every $\vec{x}_0 \in \mathbb{Z}_p^n$.

Proof. On input $\mathsf{Inst}(\vec{x}_0)$ (and a challenge Z in the decisional case) the reduction proceeds as follows.

Reparametrize. Draw $\tau \leftarrow \mathcal{D}$ and set $\vec{x}' := \tau(\vec{x}_0)$, which is unknown to $\mathcal{B}$ but uniform on $\mathbb{Z}_p^n$ by (C3).

Rebuild the instance. By (C1), each rebuilt exponent lies in the computable space of its group, so by the "$\supseteq$" part of Lemma 1 its encoding is computable from $\mathsf{Inst}(\vec{x}_0)$. Concretely, writing $r_i \circ \tau = \sum_a c_a\, g_a$ as a $\mathbb{Z}_p$-combination of the generators of $\mathcal{L}_1$, one has $[r_i(\vec{x}')]_1 = [(r_i \circ \tau)(\vec{x}_0)]_1 = \prod_a [g_a(\vec{x}_0)]_1^{c_a}$, and similarly for S (using ψ in Type 2 to realize any S-exponent in $\mathbb{G}_1$) and for T (using pairings $e([\cdot]_1, [\cdot]_2)$ to realize products). This yields $\mathsf{Inst}(\vec{x}')$, distributed exactly as a fresh average-case instance.

Transport the challenge. By (C2) compute λ_τ, h_τ and the element $[h_\tau(\vec{x}_0)]_\star$ (computable since $h_\tau \in \mathcal{L}_\star$).

Decisional case. Set $Z' := Z^{\lambda_\tau} \cdot [h_\tau(\vec{x}_0)]_\star$ and output $\mathcal{A}(\mathsf{Inst}(\vec{x}'), Z')$. If $Z = [f(\vec{x}_0)]_\star$ then

$$Z' = [\lambda_\tau f(\vec{x}_0) + h_\tau(\vec{x}_0)]_\star = [(f \circ \tau)(\vec{x}_0)]_\star = [f(\vec{x}')]_\star,$$

the real challenge for $\vec{x}'$. If Z is uniform on ch, then Z^{λ_τ} is uniform because $z \mapsto z^{\lambda_\tau}$ is a bijection of the order-p cyclic group ch (as $\gcd(\lambda_\tau, p) = 1$), so Z' is uniform and independent of $\mathsf{Inst}(\vec{x}')$. Writing p_1, p_0 for $\mathcal{A}$'s acceptance probabilities on real and random challenges,

$$\delta(\mathcal{B}; \vec{x}_0) = \mathbb{E}_\tau[p_1(\tau(\vec{x}_0))] - \mathbb{E}_\tau[p_0(\tau(\vec{x}_0))] \overset{(C3)}{=} \mathbb{E}_{\vec{x}'}[p_1] - \mathbb{E}_{\vec{x}'}[p_0] = \delta_{\mathrm{avg}}(\mathcal{A}).$$

Search case. Run $W \leftarrow \mathcal{A}(\mathsf{Inst}(\vec{x}'))$. On success $W = [f(\vec{x}')]_\star = [\lambda_\tau f(\vec{x}_0) + h_\tau(\vec{x}_0)]_\star$, so output $Y := (W \cdot [h_\tau(\vec{x}_0)]_\star^{-1})^{\lambda_\tau^{-1} \bmod p} = [f(\vec{x}_0)]_\star$. Since $\vec{x}'$ is uniform, the success probability equals $\mathsf{Succ}_{\mathrm{avg}}(\mathcal{A})$. □

Remark 2 (relaxing (C3)). When $\mathcal{D}$ makes $\tau(\vec{x})$ uniform only on a subset $\Omega \subseteq \mathbb{Z}_p^n$ (for instance $\mathbb{Z}_p^\times \times \mathbb{Z}_p$ under a multiplicative reparametrization of one coordinate), the transfer holds with the average taken over Ω. If the complement $\mathbb{Z}_p^n \setminus \Omega$ is a proper affine subvariety on which $f(\vec{x})$ takes a value the reduction can compute directly from $\mathsf{Inst}(\vec{x}_0)$ (a common case is $f(\vec{x}) = 0$ there), the reduction detects that case by an equality test on the given encodings and answers it without an oracle call, recovering full worst-case coverage.

6 Random self-reducibility: non-bilinear groups

Here $\mathcal{L}_\star = \mathcal{L}_{\mathbb{G}} = \langle R \rangle$ and no pairing exists, so (C2) demands the covariance defect to be a $\mathbb{Z}_p$-combination of the source exponents themselves. The motivating scalar case is discrete log: to solve g^x query the solver on g^{x+r} and subtract r [TW87]; its group-valued cousins fit Definition 3.

Proposition 1 (CDH and DDH). *Let $R = (1, X_1, X_2)$, so $\mathcal{L}_{\mathbb{G}} = \langle 1, X_1, X_2 \rangle$, and $f = X_1 X_2$ with challenge in $\mathbb{G}$. The translation group $\mathcal{T} = \{\vec{x} \mapsto \vec{x} + \vec{b}\}$ with $\vec{b} \leftarrow_\$ \mathbb{Z}_p^2$ satisfies* (C1)–(C3)*, so computational CDH and decisional DDH are random self-reducible.*

Proof. $X_i \circ \tau = X_i + b_i \in \mathcal{L}_{\mathbb{G}}$ gives (C1). For (C2), $f \circ \tau = (X_1 + b_1)(X_2 + b_2) = f + (b_2 X_1 + b_1 X_2 + b_1 b_2)$, so $\lambda_\tau = 1$ and $h_\tau = b_2 X_1 + b_1 X_2 + b_1 b_2 \in \langle 1, X_1, X_2 \rangle = \mathcal{L}_\star$. Translation gives exact uniformity, (C3). Explicitly the reduction returns $[f(\vec{x}_0)] = W \cdot [x_1]^{-b_2} [x_2]^{-b_1} [1]^{-b_1 b_2}$ in the search case. □

With $R = (1, X_1, \ldots, X_n)$ we have $\mathcal{L}_\star = V_{\leq 1}$ (affine polynomials), and translating a degree-d monomial leaves monomials of degree at most $d - 1$; hence (C2) holds under translation exactly when $\deg f \leq 2$. Without a pairing, degree 2 is the ceiling: a product target is the most that is randomizable, and it is.

7 Random self-reducibility: Type 1

Here $\mathcal{L}_\star = \mathcal{L}_{\mathbb{G}_T} = \langle R \cdot R \cup T \rangle$: the pairing supplies all products of source exponents, so the covariance defect may use any degree-2 combination of them.

Proposition 2 (BDDH). *Let $R = (1, X_1, X_2, X_3)$, $T = (1)$, so $\mathcal{L}_{\mathbb{G}_T} = V_{\leq 2}$, and $f = X_1 X_2 X_3$ with challenge in $\mathbb{G}_T$. Translation satisfies* (C1)–(C3)*, so decisional and computational BDDH are random self-reducible.*

Proof. $X_i \circ \tau = X_i + b_i \in \mathcal{L}_{\mathbb{G}} = \langle R \rangle$ gives (C1). Expanding,

$$f \circ \tau = X_1 X_2 X_3 + \big[b_3 X_1 X_2 + b_2 X_1 X_3 + b_1 X_2 X_3 + (\deg \leq 1)\big],$$

so $\lambda_\tau = 1$ and h_τ has total degree at most 2, hence lies in $V_{\leq 2} = \mathcal{L}_\star$; each of its monomials is realized by a single pairing, for instance $[x_1 x_2]_T = e([x_1]_1, [x_2]_2)$. Translation gives (C3). □

With coordinate source $R = (1, X_1, \ldots, X_n)$ we get $\mathcal{L}_\star = V_{\leq 2}$, so the same degree count gives translation-randomizability for every f with $\deg f \leq 3$. The pairing has raised the ceiling from 2 (no pairing) to 3.

8 Random self-reducibility: Type 2

Here $\mathcal{L}_1 = \langle R \cup S \rangle$ and $\mathcal{L}_T = \langle (R \cup S) \cdot S \cup T \rangle$: relative to Type 3 the homomorphism ψ adds the right-hand products $S \cdot S$ to the target space. This extra room is what makes the following target randomizable.

Proposition 3. *In a Type-2 group let $R = (1, X_1)$ in $\mathbb{G}_1$, $S = (1, X_2)$ in $\mathbb{G}_2$, $T = (1)$, and $f = X_1 X_2^2$ with challenge in $\mathbb{G}_T$. Then $f \notin \mathcal{L}_\star$ (so the assumption is generically hard), and translation satisfies* (C1)–(C3)*; the problem is random self-reducible.*

Proof. Here $\mathcal{L}_T = \langle (R \cup S) \cdot S \rangle = \langle 1, X_1, X_2, X_1X_2, X_2^2 \rangle$, and $X_1 X_2^2$ is not in this span, giving generic hardness. Under $\tau \colon \vec{x} \mapsto \vec{x} + \vec{b}$ one has $X_1 \circ \tau = X_1 + b_1 \in \mathcal{L}_1 = \langle 1, X_1 \rangle$ and $X_2 \circ \tau = X_2 + b_2 \in \mathcal{L}_2 = \langle 1, X_2 \rangle$, which is (C1). Expanding,

$$f \circ \tau = (X_1 + b_1)(X_2 + b_2)^2 = f + \underbrace{2b_2X_1X_2 + b_2^2X_1 + b_1X_2^2 + 2b_1b_2X_2 + b_1b_2^2}_{h_\tau},$$

so $\lambda_\tau = 1$ and every monomial of h_τ lies in $\mathcal{L}_\star$. The term X_2^2 is realized using ψ: $[x_2^2]_T = e\big(\psi([x_2]_2), [x_2]_2\big)$. Translation gives (C3). □

The point is precisely the X_2^2 term of h_τ: it is a product of two $\mathbb{G}_2$ exponents, which the pairing can form only after ψ has moved one of them into $\mathbb{G}_1$. In Type 1 this term is free; in Type 3 it is unavailable, as the next section shows.

9 Random self-reducibility: Type 3

Here $\mathcal{L}_1 = \langle R \rangle$, $\mathcal{L}_2 = \langle S \rangle$, and $\mathcal{L}_T = \langle R \cdot S \cup T \rangle$: the target space contains only cross products. This is the most restrictive setting, and reparametrizations must respect which coordinates are reachable in the challenge group.

Proposition 4 (co-CDH). *In a Type-3 group let $R = (1, X_1)$ in $\mathbb{G}_1$, $S = (1, X_2)$ in $\mathbb{G}_2$, and $f = X_1X_2$ with challenge in $\mathbb{G}_1$. Then $f \notin \mathcal{L}_\star$ (generically hard), and the reparametrization $\tau \colon (x_1, x_2) \mapsto (\alpha x_1,\ x_2 + b_2)$ with $\alpha \leftarrow_\$ \mathbb{Z}_p^\times$, $b_2 \leftarrow_\$ \mathbb{Z}_p$ satisfies* (C1),(C2) *and makes $\tau(\vec{x})$ uniform on $\mathbb{Z}_p^\times \times \mathbb{Z}_p$; the problem is random self-reducible, with the slice $x_1 = 0$ handled directly.*

Proof. Here $\mathcal{L}_\star = \mathcal{L}_1 = \langle 1, X_1 \rangle$, which contains no $\mathbb{G}_2$ variable; and $X_1X_2 \notin \langle 1, X_1 \rangle$, giving hardness. For (C1), $X_1 \circ \tau = \alpha X_1 \in \mathcal{L}_1$ and $X_2 \circ \tau = X_2 + b_2 \in \mathcal{L}_2 = \langle 1, X_2 \rangle$. For (C2), $f \circ \tau = \alpha X_1(X_2 + b_2) = \alpha f + \alpha b_2 X_1$, so $\lambda_\tau = \alpha \in \mathbb{Z}_p^\times$ and $h_\tau = \alpha b_2 X_1 \in \mathcal{L}_\star$. The image $\tau(\vec{x}) = (\alpha x_1, x_2 + b_2)$ is uniform on $\mathbb{Z}_p^\times \times \mathbb{Z}_p$ for every fixed $x_1 \neq 0$. If $x_1 = 0$ then $f(\vec{x}) = 0$ and the answer is the identity of $\mathbb{G}_1$; the reduction detects this by testing $[x_1]_1 = [0]_1$ and outputs the identity, per Remark 2. A multiplicative reparametrization of x_1 is forced because an additive shift $x_1 \mapsto x_1 + b_1$ would put b_1X_2 into h_τ, and $X_2 \notin \mathcal{L}_1$ in Type 3. □

9.1 Monomial targets never separate

It is tempting to look for a separation at the target $f = X_1X_2^2$ of Proposition 3, whose defect under translation contains the X_2^2 that Type 3 cannot form. That attempt fails, and it fails for a reason that rules out an entire class of candidates at once.

Proposition 5 (monomial targets are always self-reducible). *Let R, S, T consist of monomials, each containing 1, and let $f = X_1^i X_2^j$ with $i, j \geq 1$ be a monomial with $f \notin \mathcal{L}_\star$. Then in every one of the four settings the diagonal torus $\mathcal{T} = \{\tau_{a,d} \colon \vec{x} \mapsto (ax_1, dx_2) : a, d \in \mathbb{Z}_p^\times\}$, with a and d drawn uniformly from $\mathbb{Z}_p^\times$, satisfies* (C1) *and* (C2)*, and Theorem 1 applies in the relaxed form of Remark 2. No such f separates two settings.*

Proof. A monomial composed with a coordinate scaling is a scalar multiple of itself, so $m \circ \tau_{a,d} = a^k d^l m$ for $m = X_1^k X_2^l$. Hence every source monomial stays in the span of itself, giving (C1) in any setting, and $f \circ \tau_{a,d} = a^i d^j f$, giving (C2) with $\lambda_\tau = a^i d^j \in \mathbb{Z}_p^\times$ and $h_\tau = 0$ — the defect vanishes identically, so no uncomputable monomial can appear in it. The image $\tau_{a,d}(\vec{x})$ is uniform on $\mathbb{Z}_p^\times \times \mathbb{Z}_p^\times$ whenever $x_1, x_2 \neq 0$, and on the complement $\{x_1 = 0\} \cup \{x_2 = 0\}$ one has $f(\vec{x}) = 0$ because $i, j \geq 1$, so the answer is the identity and the reduction detects the case by testing $[x_1]_1$ and $[x_2]_2$ against the identity. Remark 2 then gives full worst-case coverage. □

In particular $f = X_1 X_2^2$ is random self-reducible in Type 3 as well as in Type 2, by $\tau \colon \vec{x} \mapsto (\alpha x_1,\ x_2 + b)$: here $f \circ \tau = \alpha f + 2\alpha b\, X_1 X_2 + \alpha b^2 X_1$, whose defect monomials $X_1 X_2$ and X_1 are both cross products $R \cdot S$ and hence in $\mathcal{L}_T$ even in Type 3. This is the same shape of reparametrization that Proposition 4 forces for co-CDH. A separating target must therefore be non-monomial, and must defeat the torus.

9.2 A separating target

Proposition 6 (a Type-2 versus Type-3 separation). *Let $p \geq 5$, $R = (1, X_1)$ in $\mathbb{G}_1$, $S = (1, X_2)$ in $\mathbb{G}_2$, $T = (1)$, and*

$$f = X_1^2 + X_1 X_2^2 + X_2^3$$

with challenge in $\mathbb{G}_T$. Then $f \notin \mathcal{L}_T$ in Type 2 and in Type 3, so the assumption is generically hard in both. In Type 2 the admissible reparametrizations act transitively and uniformly on $\mathbb{Z}_p^2$, so Theorem 1 applies with exact advantage transfer. In Type 3 the admissible group is

$$\mathcal{T}_3 = \{\tau_c \colon \vec{x} \mapsto (x_1 - 3c,\ x_2 + c) \ :\ c \in \mathbb{Z}_p\} \cong (\mathbb{Z}_p, +),$$

of order p; its orbits are the p lines $x_1 + 3x_2 = k$, each of size p; and on no such line is f computable from the instance. Hence (C3) *fails at* every $\vec{x}_0 \in \mathbb{Z}_p^2$*, even in the relaxed form of Remark 2.*

Proof. Hardness. In Type 3, $\mathcal{L}_T = \langle R \cdot S \rangle = \langle 1, X_1, X_2, X_1 X_2 \rangle$; in Type 2, $\mathcal{L}_T = \langle (R \cup S) \cdot S \rangle = \langle 1, X_1, X_2, X_1 X_2, X_2^2 \rangle$. Neither contains X_1^2, so $f \notin \mathcal{L}_T$ in either.

Type 3. By (C1), $\tau_1 = aX_1 + c_1$ and $\tau_2 = dX_2 + c_2$ with $a, d \in \mathbb{Z}_p^\times$. The monomials of $f \circ \tau$ lying outside $\mathcal{L}_T$ are X_1^2, $X_1 X_2^2$, X_2^3 and X_2^2; the first three occur in f and the fourth does not. Reading off coefficients,

$$[X_1^2] :\ a^2 = \lambda_\tau, \qquad [X_1 X_2^2] :\ ad^2 = \lambda_\tau, \qquad [X_2^3] :\ d^3 = \lambda_\tau.$$

From the first two, $a = d^2$; from the last two, $a = d$; hence $d^2 = d$ and, as $d \in \mathbb{Z}_p^\times$, $d = 1$, so $a = 1$ and $\lambda_\tau = 1$. The pure X_2^2 coefficient is $c_1 + 3c_2$ and must vanish, giving

$c_1 = -3c_2$. Writing $c := c_2$ yields exactly $\mathcal{T}_3$, and one checks $h_{\tau_c} = 2c\,X_1X_2 + (c^2 - 6c)X_1 - 3c^2X_2 + (9c^2 - 2c^3) \in \mathcal{L}_T$, so (C2) does hold on $\mathcal{T}_3$. Since $x_1 + 3x_2$ is invariant under τ_c, the orbit of $\vec{x}_0$ is the line $\ell_k = \{x_1 + 3x_2 = k\}$ through it, of size p out of p^2.

No escape on an orbit. Parametrize ℓ_k by $s := x_2$, so $x_1 = k - 3s$. There

$$f|_{\ell_k} = (k - 3s)^2 + (k - 3s)s^2 + s^3 = -2s^3 + (k + 9)s^2 - 6ks + k^2,$$

while $\mathcal{L}_T$ restricts to $\langle 1,\ k - 3s,\ s,\ (k - 3s)s\rangle = \langle 1, s, s^2\rangle$. As $p \geq 5$ the coefficient -2 is nonzero and s^3 is not a $\mathbb{Z}_p$-combination of $1, s, s^2$ as a function on $\mathbb{Z}_p$, so $f|_{\ell_k} \notin \mathcal{L}_T|_{\ell_k}$ for every k. The escape clause of Remark 2 is therefore unavailable on any orbit.

Type 2. Now $\mathcal{L}_1 = \langle 1, X_1, X_2\rangle$, so (C1) permits $\tau_1 = a_{11}X_1 + a_{12}X_2 + c_1$, and $X_2^2 \in \mathcal{L}_T$ removes the X_2^2 constraint. The surviving conditions are $a_{11}^2 = \lambda_\tau$, $a_{11}d^2 = \lambda_\tau$ and $a_{12}d^2 + d^3 = \lambda_\tau$, giving $a_{11} = d^2$ and $a_{12} = d^2 - d$, with c_1, c_2 free. Then $\tau(\vec{x}) = (d^2x_1 + (d^2 - d)x_2 + c_1,\ dx_2 + c_2)$, and as c_1, c_2 range over $\mathbb{Z}_p$ the two coordinates range independently over all of $\mathbb{Z}_p$: $\tau(\vec{x})$ is exactly uniform on $\mathbb{Z}_p^2$ for every fixed $\vec{x}$, which is (C3) unrelaxed. □

Two features of Type 2 drive the separation, and the proof uses both: ψ places X_2^2 in $\mathcal{L}_T$, which removes the constraint that ties c_1 to c_2 in Type 3, and it places X_2 in $\mathcal{L}_1$, which lets τ_1 carry the shear term $a_{12}X_2$. Neither is available in Type 3, and either alone would leave a smaller admissible group.

On the other side, all three terms of f are needed, and the coefficient conditions $a^2 = ad^2 = d^3 = \lambda_\tau$ are what make them work together: any two of the three force only a single relation between a and d, leaving a one-parameter torus, whereas all three force $d = 1$ and kill the torus that Proposition 5 would otherwise use to rescue the Type-3 reduction. Dropping a term accordingly costs the separation, but in different ways. Without X_1^2 one has $a = d$; the torus survives, and the failure locus shrinks to the single line $x_1 + 3x_2 = 0$, an $O(1/p)$ fraction. Without $X_1X_2^2$ one has $a^2 = d^3$, and without X_2^3 one has $a = d^2$; in both cases the surviving one-parameter torus leaves Type-3 orbits of size $\Theta(p^2)$, so the loss is a constant factor rather than a break — Type 2 retains full coverage in each case, so what fails is the Type-3 half of the separation, not the Type-2 half.

The hypothesis $p \geq 5$ is not cosmetic. The proof needs s^3 to be independent of $1, s, s^2$ *as a function* on $\mathbb{Z}_p$, which holds exactly when $p > 3$. At $p = 3$ Fermat gives $s^3 = s$, so $f|_{\ell_k} = ks^2 + s + k^2$ lies in $\mathcal{L}_T|_{\ell_k}$ and the escape clause of Remark 2 becomes available on every orbit; at $p = 2$ the assumption is vacuous, since X_1^2, X_2^2 and X_2^3 collapse to X_1, X_2, X_2 and f agrees pointwise with $X_1 + X_1X_2 + X_2 \in \mathcal{L}_T$. Both failures are of hardness or of the escape clause, not of the group computation, which is characteristic-free.

Remark 3 (the failure is as large as it can be). The set of τ satisfying (C1) and (C2) is closed under composition and inversion, so it is a group and the orbits of $\mathbb{Z}_p^n$ under it partition the space. Orbit size is constant on orbits, so the set of $\vec{x}_0$ at which (C3) fails is a union of orbits. If it is not all of $\mathbb{Z}_p^n$, it omits some orbit of size at least $p^n - O(p)$ and therefore has size $O(p)$. The failure fraction is thus either $O(1/p)$ or exactly 1, with nothing in between, and Proposition 6 realizes the second alternative.

Proposition 6 rules out the single-reparametrization technique of Theorem 1, which is the tool behind every positive result above. It does not rule out every reduction: Section 10 shows that a multi-query interpolation reduction still solves the *computational* problem, at high solver advantage, for this target as for any other with these source tuples. The Type-2 versus Type-3 gap established here is accordingly a gap for decisional reductions and for reductions that must work at negligible advantage. Ruling out every decisional reduction would additionally require a lower-bound argument such as a meta-reduction.

10 Relation to classical random self-reducibility

Random self-reducibility long predates its use on group-based assumptions, and the classical technique differs from Theorem 1 in one respect that decides exactly what Proposition 6 does and does not rule out. This section isolates the difference and delimits the reach of each technique.

10.1 The multi-query condition

Condition (C2) asks a single reparametrized challenge to be a scalar multiple of the original up to a computable defect. Feigenbaum and Fortnow [FF93] call a problem non-adaptively random self-reducible when $f(\vec{x}_0)$ is recoverable by an arbitrary polynomial-time function of the answers on several identically distributed queries; see [BT06] for the place of that notion in average-case complexity. The specialization in which the recovery is $\mathbb{Z}_p$-linear is the one this setting supports:

(C2′) there are $\tau_1, \ldots, \tau_m$, jointly sampled so that each τ_i is marginally distributed as in (C3), together with coefficients $c_1, \ldots, c_m \in \mathbb{Z}_p$ and a polynomial $h \in \mathcal{L}_\star$, all computable from the sampled data, such that

$$f = \sum_{i=1}^{m} c_i \, (f \circ \tau_i) + h.$$

Condition (C2) is the case $m = 1$ with $c_1 \neq 0$: from $f \circ \tau = \lambda_\tau f + h_\tau$ one gets $f = \lambda_\tau^{-1}(f \circ \tau) - \lambda_\tau^{-1} h_\tau$, which is (C2′) with $c_1 = \lambda_\tau^{-1}$ and $h = -\lambda_\tau^{-1} h_\tau$.

Given (C1) for each τ_i, condition (C2′) suffices for the *computational* problem: the reduction rebuilds $\mathsf{Inst}(\tau_i(\vec{x}_0))$ for each i as in the proof of Theorem 1, queries the solver on all m of them to get $W_1, \ldots, W_m$, and returns

$$\prod_{i=1}^{m} W_i^{\,c_i} \cdot [h(\vec{x}_0)]_\star = \Big[\sum_{i=1}^{m} c_i \, f(\tau_i(\vec{x}_0)) + h(\vec{x}_0)\Big]_\star = [f(\vec{x}_0)]_\star.$$

The step that makes this go through is that a $\mathbb{Z}_p$-linear combination of *exponents* is realized by group operations on their encodings, so linear post-processing of the solver's answers is available to the reduction for free. Encodings are transparent to it: the map $\vec{x} \mapsto [f(\vec{x})]_\star$ is a Reed–Muller codeword read through the encoding, and a linear local-decoding procedure for that code never notices the encoding is there. This

is why the linear part of the classical toolkit — Lipton's line restriction [Lipton91], and the degree-k curves with which Beaver and Feigenbaum [BF90] make the queries k-wise independent — carries over to computational Uber problems unchanged.

10.2 Interpolation, and what it costs

The classical instantiation restricts f to a random line through $\vec{x}_0$ and interpolates. It needs nothing of f beyond its degree, and nothing of the setting beyond (C1) for translations.

Proposition 7 (line interpolation). *Suppose R, S and T are stable under translation, in the sense that $r \circ \tau_{\vec{b}} \in \mathcal{L}_1$, $s \circ \tau_{\vec{b}} \in \mathcal{L}_2$ and $t \circ \tau_{\vec{b}} \in \mathcal{L}_T$ for every $\vec{b} \in \mathbb{Z}_p^n$ and every $r \in R$, $s \in S$, $t \in T$, where $\tau_{\vec{b}} \colon \vec{x} \mapsto \vec{x} + \vec{b}$. Let $d \coloneqq \deg f$ and suppose $d \le p - 2$. Then the* computational $(R, S, T, f; \mathrm{ch})$*-Uber problem is random self-reducible by a non-adaptive $(d+1)$-query reduction $\mathcal{B}^{\mathcal{A}}$ with*

$$\mathsf{Succ}(\mathcal{B}; \vec{x}_0) \;\ge\; 1 - (d+1)\big(1 - \mathsf{Succ}_{\mathrm{avg}}(\mathcal{A})\big) \qquad \textit{for every } \vec{x}_0 \in \mathbb{Z}_p^n.$$

Proof. Draw $\vec{b} \leftarrow_{\$} \mathbb{Z}_p^n$ and fix $d+1$ distinct nonzero $t_1, \dots, t_{d+1} \in \mathbb{Z}_p$, which exist as $d + 1 \le p - 1$. Put $\tau_i \coloneqq \tau_{t_i \vec{b}}$. Each τ_i is a translation, so (C1) holds by hypothesis and $\mathsf{Inst}(\vec{x}_0 + t_i \vec{b})$ is computable from $\mathsf{Inst}(\vec{x}_0)$; and since $t_i \neq 0$ and $\vec{b}$ is uniform, $\vec{x}_0 + t_i \vec{b}$ is uniform on $\mathbb{Z}_p^n$, so each query is marginally a fresh average-case instance. The $d+1$ queries are not jointly independent: they lie on one line.

Put $g(t) \coloneqq f(\vec{x}_0 + t\vec{b})$, a polynomial in t of degree at most d, and let $c_i \coloneqq \prod_{j \neq i} (-t_j)(t_i - t_j)^{-1}$ be the Lagrange coefficients at 0, so that $q(0) = \sum_i c_i q(t_i)$ for every q of degree at most d. This is (C2′) with $m = d+1$ and $h = 0$. Query $W_i \leftarrow \mathcal{A}(\mathsf{Inst}(\vec{x}_0 + t_i \vec{b}))$ and output $Y \coloneqq \prod_i W_i^{c_i}$; if every $W_i = [g(t_i)]_\star$ then $Y = [\sum_i c_i g(t_i)]_\star = [g(0)]_\star = [f(\vec{x}_0)]_\star$. A union bound over the $d+1$ queries gives the stated bound. □

The source tuples of Propositions 4, 5 and 6 are all translation-stable, so Proposition 7 applies to each: the computational problems there are self-reducible in Type 3 whatever the covariance obstruction, including the separating target of Proposition 6, whose degree 3 needs four queries. The reduction never rebuilds the covariance defect — the solver returns it implicitly inside its own answers — so no uncomputable monomial ever has to be formed.

What it costs is the regime. The queries must be correlated for the interpolation identity to hold, so the transfer runs through a union bound and the bound is vacuous unless $\mathsf{Succ}_{\mathrm{avg}}(\mathcal{A}) > d/(d+1)$. In the classical setting that threshold is soft: Berlekamp–Welch decoding brings the tolerable disagreement to any constant below $1/2$ [GS92], and list decoding brings it past $1/2$ [Sudan97, STV01]. Neither is available here. Berlekamp–Welch solves $N(t_i) = y_i E(t_i)$ for the coefficients of N and E, a linear system whose *coefficient matrix contains the received values* y_i; the reduction holds only $[y_i]_\star$, and forming that system would require multiplying and inverting encoded values. The same objection applies to the list-decoding refinements. So the very criterion that lets interpolation survive the encoding — linearity in the received word — excludes

the machinery that would soften its threshold, and $d/(d+1)$ is a genuine wall rather than an artifact.

Remark 4 (challenge transport). Interpolation does not reach the decisional problem. Recovering $f(\vec{x}_0)$ from the answers is linear and hence free in the exponent, but a decisional reduction must also run in the other direction, transporting its own challenge Z forward to a challenge for each rebuilt instance, and decision bits do not interpolate. Concretely, for the target $f = X_1^2 + X_1X_2^2 + X_2^3$ of Proposition 6 the reduction would need the coefficients of $g(t) = f(\vec{x}_0 + t\vec{b})$; the coefficient of t is the directional derivative

$$b_1\big(2X_1 + X_2^2\big) + b_2\big(2X_1X_2 + 3X_2^2\big) = 2b_1X_1 + 2b_2X_1X_2 + (b_1 + 3b_2)X_2^2,$$

whose X_2^2 term lies outside $\mathcal{L}_T$ in Type 3 unless $b_1 + 3b_2 = 0$ — that is, unless $\vec{b}$ points along the one direction whose orbits are the degenerate lines of Proposition 6.

10.3 What each technique buys

The two techniques are not ordered by reach. On *computational* problems interpolation is the more permissive of the two: it asks nothing of f but its degree, so wherever the sources are translation-stable it succeeds regardless of the covariance obstruction, and Proposition 7 applies to every target in this note. What separates them is the regime. Theorem 1 transfers advantage exactly, and reaches the decisional problem; Proposition 7 says nothing below solver advantage $d/(d+1)$ and nothing about decisional problems at all (Remark 4). The gap is not one of tightness — a reduction requiring $\mathsf{Succ}_{\mathrm{avg}} > 3/4$ is not a loose reduction, it does not exist below that threshold — and, by the preceding subsection, it cannot be closed by the classical decoding machinery, because that machinery is not linear in the received word. Since Uber assumptions are stated at negligible advantage, and very often in decisional form, the covariance route is in practice the only one available; that is why Theorem 1, and not interpolation, is the tool used throughout this note, and why the separation of Proposition 6 is meaningful despite Proposition 7.

Two further points of contact are worth recording, with their differences. Shift-based self-correction in the style of Blum, Luby and Rubinfeld [BLR93] uses the same $\vec{x} \mapsto \vec{x} + \vec{b}$ as the CDH and DDH reduction of Section 6, but their corrector queries the oracle twice per trial and takes a majority over repetitions, tolerating a constant error rate, where the reduction here makes one query and computes the correction itself from the instance, transferring advantage exactly. And the worst-case-to-average-case reductions of lattice cryptography, from Ajtai [Ajtai96] to Regev [Regev09], share the motivation of this note — basing an average-case assumption on a worst-case one — but are reductions *between distinct problems* rather than self-reductions, and rest on lattice geometry rather than on polynomial structure; Regev's is moreover a quantum reduction.

11 Deciding whether the affine technique applies

Conditions (C1) and (C2) are polynomial conditions on the entries of τ, and (C3) is a statement about the size of an orbit. Finding an admissible $\mathcal{T}$ is therefore a

computation rather than a matter of ingenuity, and the hand arguments of Sections 6–9 are all instances of one procedure. This section writes it out and, at each step, names the standard operation of computational algebra that performs it. Nothing bespoke is needed: the whole of it is subspace arithmetic, one stabilizer computation, one Jacobian kernel, one ideal of maximal minors, and one primary decomposition, in that order. What the procedure decides is the applicability of Theorem 1, not random self-reducibility as such; Section 11.6 is precise about the gap.

11.1 Set-up

Fix a setting and an instance $(R, S, T, f; \mathrm{ch})$ in n variables over $\mathbb{Z}_p$, and let D be the largest degree occurring among f and the generators listed for $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_T$ in Lemma 1 — so $D \geq \deg r + \deg s$ for the pairing products, not merely the largest degree in $R \cup S \cup T$. Affine substitution does not raise degree, so every polynomial the procedure meets lies in $P_D := \mathbb{Z}_p[\vec{X}]_{\leq D}$, a $\mathbb{Z}_p$-space of dimension $N = \binom{n+D}{n}$; represent its elements by coefficient vectors. Treat the $n^2 + n$ entries of $(A, \vec{b})$ in $\tau_{A,\vec{b}} \colon \vec{x} \mapsto A\vec{x} + \vec{b}$ as indeterminates; for $q \in P_D$ the coefficients of $q \circ \tau_{A,\vec{b}}$ are then polynomials of degree at most $\deg q$ in those indeterminates, obtained by expansion. Composition with an affine map is $\mathbb{Z}_p$-linear on P_D and respects composition, so it is a representation

$$\rho \colon \mathrm{AGL}_n(\mathbb{Z}_p) \longrightarrow \mathrm{GL}(P_D), \qquad \rho(\tau) q = q \circ \tau,$$

whose matrix entries are polynomials of degree at most D in $(A, \vec{b})$. Everything below is an operation on ρ and on the subspaces of P_D that Lemma 1 supplies.

Two structural facts fix the shape of the procedure. First, (C1) and (C2) cut out an algebraic subset $\mathcal{T} \subseteq \mathrm{AGL}_n(\mathbb{Z}_p)$ which is a subgroup, and Lemma 2 below identifies it as a stabilizer, which is where the group structure asserted in Remark 3 comes from. Second, with $\mathcal{D}$ uniform on $\mathcal{T}$ the image $\tau(\vec{x}_0)$ is uniform on the orbit $\mathcal{T}\vec{x}_0$, because $\tau \mapsto \tau(\vec{x}_0)$ is $|\mathrm{Stab}_{\mathcal{T}}(\vec{x}_0)|$-to-one onto that orbit. Deciding (C3) is therefore deciding how large the orbits are, and by the Lang–Weil estimate an orbit of dimension e carries $\Theta(p^e)$ points, with implied constants depending only on n and D. For p large relative to those, comparing $\dim \mathcal{T}\vec{x}_0$ with n decides (C3) up to exactly the $O(1/p)$ slack that Remark 2 already tolerates.

11.2 The admissible set is a stabilizer

Conditions (C1) and (C2) look like conditions of different kinds — one a membership requirement on the rebuilt sources, the other a covariance requirement on the challenge. Under ρ they are two instances of one condition: each says that τ fixes something.

Lemma 2 (stabilizer form). *Write $\bar{f} := f + \mathcal{L}_\star \in P_D/\mathcal{L}_\star$. For $\tau \in \mathrm{AGL}_n(\mathbb{Z}_p)$:*

(i) (C1) *holds if and only if $\rho(\tau)$ stabilizes each of $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_T$;*

(ii) *assuming* (C1)*, so that $\rho(\tau)$ descends to a map $\bar{\rho}(\tau)$ of $P_D/\mathcal{L}_\star$, condition* (C2) *holds if and only if $\bar{\rho}(\tau)$ fixes the line $\langle \bar{f} \rangle$, and λ_τ is then the scalar by which it acts there.*

Hence the admissible set is a stabilizer,

$$\mathcal{T} = \{\tau : \rho(\tau)\mathcal{L}_\gamma = \mathcal{L}_\gamma \ (\gamma = 1, 2, T) \ \text{ and } \ \bar{\rho}(\tau)\langle \bar{f}\rangle = \langle \bar{f}\rangle\},$$

the ρ-preimage of an intersection of three subspace stabilizers — maximal parabolic subgroups of $\mathrm{GL}(P_D)$, or all of $\mathrm{GL}(P_D)$ in the degenerate case $\mathcal{L}_\gamma = P_D$ — with the stabilizer of a point of $\mathbb{P}(P_D/\mathcal{L}_\star)$. In particular $\mathcal{T}$ is a subgroup of $\mathrm{AGL}_n(\mathbb{Z}_p)$, and $\lambda : \mathcal{T} \to \mathbb{Z}_p^\times$, $\tau \mapsto \lambda_\tau$, is a character of it.

Proof. (i) In each of the four settings the computable spaces of Lemma 1 are generated by R, S, T under multiplication: $\mathcal{L}_2 = \langle S\rangle$, $\mathcal{L}_1 = \langle R\rangle$ or $\langle R \cup S\rangle$, and $\mathcal{L}_T = \mathcal{L}_1 \cdot \mathcal{L}_2 + \langle T\rangle$, with $\mathcal{L}_1 = \mathcal{L}_2 = \langle R\rangle$ in the single-group settings and $\mathcal{L}_T$ absent when there is no pairing. If $\rho(\tau)$ stabilizes the three spaces then (C1) follows, since $R \subseteq \mathcal{L}_1$, $S \subseteq \mathcal{L}_2$ and $T \subseteq \mathcal{L}_T$. Conversely assume (C1). Then $\rho(\tau)\langle S\rangle \subseteq \mathcal{L}_2 = \langle S\rangle$; and both $\langle R\rangle$ and $\langle S\rangle$ land in $\mathcal{L}_1$ — in Type 2 because $\mathcal{L}_2 \subseteq \mathcal{L}_1$, in the other settings because $\mathcal{L}_1 = \langle R\rangle$ and only R is involved — so $\rho(\tau)\mathcal{L}_1 \subseteq \mathcal{L}_1$. Composition with τ is a ring endomorphism of $\mathbb{Z}_p[\vec{X}]$, so $(uv) \circ \tau = (u \circ \tau)(v \circ \tau)$ and therefore $\rho(\tau)(\mathcal{L}_1 \cdot \mathcal{L}_2) \subseteq \mathcal{L}_1 \cdot \mathcal{L}_2$; with $\rho(\tau)\langle T\rangle \subseteq \mathcal{L}_T$ from (C1) this gives $\rho(\tau)\mathcal{L}_T \subseteq \mathcal{L}_T$. Each inclusion is between spaces of equal finite dimension under an invertible map, hence an equality.

(ii) The identity $f \circ \tau = \lambda_\tau f + h_\tau$ with $h_\tau \in \mathcal{L}_\star$ says exactly that the image of $f \circ \tau$ in $P_D/\mathcal{L}_\star$ is $\lambda_\tau \bar{f}$, and the requirement $\lambda_\tau \in \mathbb{Z}_p^\times$ says the image is a *nonzero* multiple, that is, that the line is fixed rather than merely mapped into itself. The final claims follow because the preimage of an intersection of subgroups under a homomorphism is a subgroup, and because the scalar by which a group acts on a fixed line is multiplicative in the group. □

Lemma 2 supersedes the direct check in Remark 3 that $\mathcal{T}$ is closed under composition and inversion: it is a preimage of a subgroup. It also places λ correctly as a character rather than an incidental scalar, which is what makes $\lambda_\tau \in \mathbb{Z}_p^\times$ compose in the multi-query condition (C2′).

11.3 Orbits: invariants and a rank locus

By Lemma 2, $\mathcal{T}$ is an algebraic subgroup of AGL_n acting on affine n-space, and (C3) asks that its orbits be everything. Two standard descriptions of orbits apply, one generic and one pointwise, and between them they replace the orbit bookkeeping of Sections 6–9.

Generically, orbits are the fibres of the invariant map. By Rosenlicht's theorem on generic quotients there is a dense open subset of $\overline{\mathbb{Z}_p}^n$ on which the orbits of $\mathcal{T}$ are exactly the fibres of the map assembled from a finite generating set of the invariant field, and $\mathrm{trdeg}\,\overline{\mathbb{Z}_p}(\vec{X})^{\mathcal{T}} = n - \max_{\vec{x}} \dim \mathcal{T}\vec{x}$. So the generic orbit is n-dimensional precisely when $\mathcal{T}$ admits no non-constant rational invariant. This is the exact shape of the Type-3 obstruction in Proposition 6: there the invariant is $x_1 + 3x_2$, the orbits are its level sets, and the "no escape on an orbit" step of that proof is a membership test on the generic fibre.

Pointwise, the failure locus is where the infinitesimal action drops rank. Write $\mathfrak{agl}_n = \mathfrak{gl}_n \ltimes \mathbb{Z}_p^n$ and let $(M, \vec{v})$ act on polynomials as the derivation

$$\delta_{M,\vec{v}} = \sum_{i=1}^{n} (M\vec{X} + \vec{v})_i \, \frac{\partial}{\partial X_i},$$

the derivative at $t = 0$ of $q \mapsto q \circ (\mathrm{id} + t\,(M, \vec{v}))$. Differentiating the conditions of Lemma 2 at the identity turns them into a *linear* system in $(M, \vec{v})$:

(L1) $\delta r \in \mathcal{L}_1$, $\delta s \in \mathcal{L}_2$ and $\delta t \in \mathcal{L}_T$ for all $r \in R$, $s \in S$, $t \in T$;

(L2) $\delta f \in \langle f \rangle + \mathcal{L}_\star$, the coefficient of f being $d\lambda(\delta)$.

Its solution space $\widehat{\mathfrak{t}}$ is the tangent space at the identity to the scheme cut out by (C1) and (C2), so $\mathrm{Lie}\,\mathcal{T} \subseteq \widehat{\mathfrak{t}}$ and $\dim \mathcal{T} \le \dim \widehat{\mathfrak{t}}$, with equality when that scheme is smooth at the identity.

Lemma 3 (rank locus). *Let* $\mathfrak{t} \subseteq \mathrm{Lie}\,\mathcal{T}$ *be spanned by* $(M_1, \vec{v}_1), \ldots, (M_d, \vec{v}_d)$ *and let* $\Theta(\vec{X})$ *be the* $n \times d$ *matrix of affine-linear forms whose* i*-th column is* $M_i\vec{X} + \vec{v}_i$*. Then* $\operatorname{rank} \Theta(\vec{x}) \le \dim \mathcal{T}\vec{x}$ *for every* $\vec{x}$*, with equality when* $\mathfrak{t} = \mathrm{Lie}\,\mathcal{T}$ *and the orbit map* $\tau \mapsto \tau(\vec{x})$ *is separable. Consequently the failure locus* $Z := \{\vec{x} : \dim \mathcal{T}\vec{x} < n\}$ *satisfies*

$$Z \;\subseteq\; V\big(I_n(\Theta)\big),$$

where $I_n(\Theta)$ *is the ideal of* $n \times n$ *minors of* Θ *— the* 0*-th Fitting ideal of* $\operatorname{coker} \Theta$ *— generated in degree at most* n*, and equal to* (0) *when* $d < n$*.*

Proof. Orbits of an algebraic group are homogeneous, hence smooth, so $\dim \mathcal{T}\vec{x} = \dim T_{\vec{x}}(\mathcal{T}\vec{x})$. The orbit map $\mathcal{T} \to \mathcal{T}\vec{x}$ has differential $\mu \mapsto \mu(\vec{x})$ at the identity, whose image lies in $T_{\vec{x}}(\mathcal{T}\vec{x})$ and contains the column span of $\Theta(\vec{x})$, with equality when $\mathfrak{t} = \mathrm{Lie}\,\mathcal{T}$; it is the whole tangent space exactly when the orbit map is separable. The displayed inclusion is the rank bound read contrapositively: $\operatorname{rank} \Theta(\vec{x}) = n$ forces $\dim \mathcal{T}\vec{x} = n$. □

Lemma 3 is what makes the failure locus computable rather than merely definable. The entries of Θ are affine-linear, so its maximal minors have degree at most n and are written down by expansion; the locus they cut out is a determinantal variety, and its components come from a primary decomposition. In every example of this note the inclusion is an equality and the minors reproduce the loci found by hand: (1) — empty locus — for CDH, DDH and BDDH, the ideal (x_1) for co-CDH, the ideal (x_1x_2) for the monomial targets of Proposition 5, and (0) — the whole space — for the Type-3 half of Proposition 6, where $\mathfrak{t}$ is one-dimensional and no 2×2 minor exists.

11.4 The procedure

Admissible$(\Gamma, R, S, T, f, \mathrm{ch}, p)$ — decides applicability of Theorem 1. The bracketed name on each step is the operation it calls.

Returns one of Not-Hard, Exact, Relaxed, Affine-Fails, together with $\mathcal{T}$ and, in the last case, a witness locus.

1. *Computable spaces* [image of a multiplication map]. From Lemma 1 form the generating sets for the setting Γ and compute bases of $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_T \subseteq P_D$ by Gaussian elimination; the target space is the image of the multiplication map $\mathcal{L}_1 \otimes \mathcal{L}_2 \to P_D$ plus $\langle T \rangle$. Put $\mathcal{L}_\star \coloneqq \mathcal{L}_{\mathrm{ch}}$.

2. *Hardness* [subspace membership]. Test $f \in \mathcal{L}_\star$. If so, return Not-Hard: the challenge is already computable from the instance and there is no assumption to reduce.

3. *Admissible group* [stabilizer; Gröbner basis of a saturated ideal]. Let $\pi \colon P_D \to P_D/\mathcal{L}_\star$ and $\bar{f} \coloneqq \pi(f)$, nonzero by step 2. With the $n^2 + n$ entries of $(A, \vec{b})$ and one further indeterminate λ, let I be generated by

   (a) the images of $\mathrm{coef}(r \circ \tau)$ in $P_D/\mathcal{L}_1$, of $\mathrm{coef}(s \circ \tau)$ in $P_D/\mathcal{L}_2$, and of $\mathrm{coef}(t \circ \tau)$ in $P_D/\mathcal{L}_T$, for each $r \in R$, $s \in S$, $t \in T$ — this is (C1) in the form of Lemma 2(i);

   (b) the coefficients of $\pi(f \circ \tau) - \lambda \bar{f}$ — this is (C2) in the form of Lemma 2(ii).

   Compute a Gröbner basis of the saturation $I : (\lambda \det A)^\infty$, by Rabinowitsch in one extra variable. Its variety is $\mathcal{T}$, the character λ is the λ-coordinate, and $\dim \mathcal{T}$ is the Hilbert dimension of the basis. If $\mathcal{T}$ is trivial, return Affine-Fails with witness $\mathbb{Z}_p^n$.

4. *Infinitesimal action* [Jacobian kernel: one linear system]. Solve (L1)–(L2) for $\widehat{\mathfrak{t}}$. If $\dim \widehat{\mathfrak{t}} < n$ then $\dim \mathcal{T} \le \dim \widehat{\mathfrak{t}} < n$, every orbit is deficient, and by Remark 3 (C3) fails everywhere: return Affine-Fails with witness $\mathbb{Z}_p^n$. Otherwise retain those basis elements whose one-parameter subgroups can be verified against (C1)–(C2) by substitution, obtaining $\mathfrak{t} \subseteq \mathrm{Lie}\,\mathcal{T}$; retaining fewer only makes the verdict more conservative.

5. *Transitivity* [the translation subgroup, a $\mathbb{Z}_p$-subspace]. Let $W \coloneqq \{\vec{b} : \tau_{\mathrm{id},\vec{b}} \in \mathcal{T}\}$, a subgroup of $(\mathbb{Z}_p^n, +)$ and hence a $\mathbb{Z}_p$-subspace; compute it as $\{\vec{v} : (0, \vec{v}) \in \widehat{\mathfrak{t}}\}$ and verify each basis vector against (C1)–(C2). If $W = \mathbb{Z}_p^n$ then $\mathcal{T}$ is transitive: return Exact, and Theorem 1 applies with unrelaxed (C3) and exact advantage transfer.

6. *Failure locus* [maximal minors: a Fitting ideal, then its minimal primes]. Otherwise form Θ from $\mathfrak{t}$ and compute $I_n(\Theta)$. By Lemma 3 the failure locus satisfies $Z \subseteq V(I_n(\Theta))$; take the minimal primes $Z_1, \dots, Z_m$ of $V(I_n(\Theta))$ by

primary decomposition. If $I_n(\Theta) = (1)$ the orbits all have dimension n and, by Remark 3, return RELAXED. If $I_n(\Theta) = (0)$, return AFFINE-FAILS with witness $\mathbb{Z}_p^n$.

7. *Escape clause* [ideal membership; normal form with the field equations adjoined]. For each component Z_j test both:
   (a) *detectability* — is the ideal $I(Z_j)$ generated by polynomials lying in $\mathcal{L}_\gamma$ for some group γ? If so the reduction decides membership by testing each $[q(\vec{x}_0)]_\gamma$ against the identity;
   (b) *computability* — is $f \in \mathcal{L}_\star$ *as a function* on $Z_j(\mathbb{Z}_p)$? Adjoin the field equations and reduce: compute normal forms modulo a Gröbner basis of $I(Z_j) + (X_1^p - X_1, \ldots, X_n^p - X_n)$, an ideal that is radical and whose variety is exactly $Z_j(\mathbb{Z}_p)$, and test membership of the normal form of f in the span of those of $\mathcal{L}_\star$.

   If both hold for every j, return RELAXED: Theorem 1 applies via Remark 2, with advantage transfer up to $O(1/p)$. Otherwise return AFFINE-FAILS with the first offending Z_j as witness.

Three of these steps are worth a word. Step 3 introduces λ as an indeterminate rather than imposing the proportionality $\pi(f \circ \tau) \parallel \bar{f}$ through the vanishing of the 2×2 minors of $[\, \pi(f \circ \tau) \mid \bar{f} \,]$: the two cut out the same set, but the minors number $\binom{\dim P_D/\mathcal{L}_\star}{2}$ and have degree $2D$, whereas the linear-in-λ form has $\dim P_D/\mathcal{L}_\star$ generators of degree D, and the saturation at λ is exactly the requirement $\lambda_\tau \in \mathbb{Z}_p^\times$. Step 5 tests a sufficient condition for transitivity rather than transitivity itself, because that is all the EXACT verdict needs and because the translation subgroup, being a subgroup of $(\mathbb{Z}_p^n, +)$, is a subspace and so costs one linear solve; a transitive $\mathcal{T}$ with $W \neq \mathbb{Z}_p^n$ is merely reported as RELAXED, at a cost of $O(1/p)$ in the advantage. Step 7(b) is where the distinction between polynomials and functions bites — f may lie outside $\mathcal{L}_\star$ as a polynomial and inside it as a function on $Z_j(\mathbb{Z}_p)$ — and the standard device for it is to adjoin the field equations $X_i^p - X_i$, which is what forces the hypothesis $p \geq 5$ in Proposition 6: at $p = 3$ the relation $s^3 = s$ is among them.

Proposition 8 (correctness). *ADMISSIBLE terminates, and its output is correct: it returns NOT-HARD exactly when $f \in \mathcal{L}_\star$; it returns EXACT only when some efficiently sampleable $\mathcal{D}$ satisfies* (C1)–(C3) *as stated in Theorem 1; it returns RELAXED only when Theorem 1 applies in the form of Remark 2; when it returns AFFINE-FAILS with witness $\mathbb{Z}_p^n$, no distribution on any subgroup of* $\mathrm{AGL}_n(\mathbb{Z}_p)$ *satisfies* (C1)–(C3) *at any point, even relaxed; and when it returns AFFINE-FAILS with witness a component Z_j on which $\dim \mathcal{T}\vec{x} < n$, the same holds at the points of Z_j.*

Proof. Termination is that every step is a finite linear-algebra or Gröbner computation over $\mathbb{Z}_p$. Step 2 is the generic-model hardness criterion of [BBG05, Boyen08]. Steps 3 and 4 are exact transcriptions of Lemma 2 and of its derivative at the identity: membership of $q \circ \tau$ in a subspace is the vanishing of its image in the quotient, and

$f \circ \tau = \lambda_\tau f + h_\tau$ with $h_\tau \in \mathcal{L}_\star$ says that $\pi(f \circ \tau)$ equals $\lambda \bar{f}$, the saturation enforcing $\lambda_\tau \neq 0$ and $\det A \neq 0$. Since $\mathcal{T}$ is a group and $\mathcal{D}$ is uniform on it, $\tau(\vec{x}_0)$ is uniform on $\mathcal{T}\vec{x}_0$; so (C3) holds exactly when the orbit is all of $\mathbb{Z}_p^n$, which step 5 certifies through a transitive subgroup of translations, and holds to within $O(1/p)$ when the orbit has full dimension, which step 6 certifies through Lemma 3. Maximality is what gives the negative half: by Lemma 2, $\mathcal{T}$ contains *every* τ meeting (C1) and (C2), so any admissible subgroup is contained in $\mathcal{T}$ and its orbits in $\mathcal{T}$-orbits; if the $\mathcal{T}$-orbit of a point of Z_j is already deficient and f is not computable on it, no smaller group can do better. For the witness $\mathbb{Z}_p^n$ at step 4 this is unconditional, since $\dim \mathcal{T} \leq \dim \widehat{\mathfrak{t}}$ bounds the orbit dimension at every point at once. □

Remark 5 (the negative verdict at step 7 is conservative). Lemma 3 gives $Z \subseteq V(I_n(\Theta))$ and not equality, so a component Z_j returned by step 6 is a *candidate* failure locus: it is a genuine one when $\mathfrak{t} = \mathrm{Lie}\,\mathcal{T}$ and the orbit maps along Z_j are separable, and otherwise the orbits along Z_j may be larger than Θ can see, in which case the procedure has failed to certify rather than certified a failure. The positive verdicts are unaffected, since they rest on the inclusion in the direction Lemma 3 does supply. When the distinction matters it is settled by the more expensive test the rank locus replaces: compute the orbit of a generic point of Z_j as the image of the morphism $\tau \mapsto \tau(\vec{x}_0)$, by eliminating $(A, \vec{b})$ from $\vec{y} = A\vec{x}_0 + \vec{b}$ over I and applying the closure theorem, and compare its dimension with n. In every example of this note the two agree.

Remark 6 (cost). Only step 3 is expensive: its Gröbner computation is doubly exponential in $n^2 + n + 1$ in the worst case, though the systems arising here are far from worst case — they are the coefficient-matching equations of Section 9, near-triangular in the entries of A, and for $n = 2$ they are seven variables and are solved by hand. It is also, for the verdict, optional: every one of the four outputs is reached from steps 1, 2 and 4–7, which need only $\widehat{\mathfrak{t}}$ and $\mathfrak{t}$. What step 3 adds is $\mathcal{T}$ in closed form and the character λ, and with them the maximality on which the negative half of Proposition 8 rests. Steps 1, 2, 5 and 7 are $O(N^3)$ linear algebra with $N = \binom{n+D}{n}$; step 4 is a linear system in $n^2 + n$ unknowns; step 6 is $\binom{d}{n}$ determinants of size n followed by a primary decomposition in n variables of an ideal generated in degree at most n. Every primitive named above is a standard call in a general-purpose system such as Singular, Macaulay2 or Magma.

11.5 The primitives, and the procedure on the examples of this note

Collecting the bracketed names gives the whole of Admissible as a pipeline:

step	task	operation
1	the computable spaces $\mathcal{L}_\gamma$	image of a multiplication map
2	hardness, $f \in \mathcal{L}_\star$	subspace membership
3	the admissible group $\mathcal{T}$	stabilizer of subspaces and of a point of $\mathbb{P}(P_D/\mathcal{L}_\star)$; Gröbner basis of a saturated ideal
3	the scalars λ_τ	the character of $\mathcal{T}$ on $\langle \bar{f} \rangle$
4	Lie $\mathcal{T}$	kernel of the Jacobian at the identity
5	transitivity	the translation subspace $W \le \mathbb{Z}_p^n$
6	the failure locus	0-th Fitting ideal $I_n(\Theta)$, then its minimal primes
7	detectability of Z_j	ideal membership
7	computability on Z_j	normal form modulo $I(Z_j) + (X_i^p - X_i)$

Run on the assumptions treated above, it reproduces every one of the hand arguments, with the whole of the orbit analysis coming from a single determinant:

target	$\mathfrak{t}$, as vector fields	$I_n(\Theta)$	verdict
CDH, DDH (Prop. 1)	$\langle \partial_1, \partial_2, X_1\partial_1, X_2\partial_2 \rangle$	(1)	Exact
BDDH (Prop. 2)	$\langle \partial_i,\ X_i\partial_i : i \le 3 \rangle$	(1)	Exact
co-CDH (Prop. 4)	$\langle X_1\partial_1, X_2\partial_2, \partial_2 \rangle$	(x_1)	Relaxed
$X_1^i X_2^j$ (Prop. 5)	$\supseteq \langle X_1\partial_1, X_2\partial_2 \rangle$	$\ni x_1x_2$	Relaxed
Prop. 6, Type 2	$\langle \partial_1, \partial_2, (2X_1{+}X_2)\partial_1 + X_2\partial_2 \rangle$	(1)	Exact
Prop. 6, Type 3	$\langle -3\partial_1 + \partial_2 \rangle$	(0)	Affine-Fails

The last two rows are the separation, and in this form it is a statement about one number: the Type-2 conditions leave a three-dimensional $\mathfrak{t}$ containing both translations, so Θ has an invertible 2×2 submatrix and the failure locus is empty, while the Type-3 conditions cut $\mathfrak{t}$ down to a single vector field, Θ becomes a 2×1 matrix, and no 2×2 minor exists to be nonzero. The intermediate rows show the two ways a non-empty failure locus can still be harmless, both settled by step 7: the locus $x_1 = 0$ for co-CDH and $x_1x_2 = 0$ for the monomial targets are detectable by an equality test on the encodings, and f vanishes on each, so the reduction answers there without an oracle call. Row three also shows what maximality buys: the group the procedure returns for co-CDH is three-dimensional, strictly larger than the $\{(\alpha x_1,\ x_2 + b_2)\}$ used in the proof of Proposition 4, which chose a sufficient subgroup rather than the largest one.

11.6 What the procedure does not decide

Affine-Fails is not a proof that the assumption fails to be random self-reducible. It is a proof about one technique, and three gaps remain open beneath it.

First, multi-query reductions. Proposition 7 solves the computational problem whenever the sources are translation-stable and $\deg f \le p - 2$, whatever the verdict of Admissible — as it does for the separating target of Proposition 6. A run returning Affine-Fails should therefore be followed by that test, which is a one-line check. The general linear multi-query condition (C2′) is decidable too, but by a different route: it asks whether f lies in $\langle f \circ \tau : \tau \text{ satisfies (C1)} \rangle + \mathcal{L}_\star$. By Lemma 2(i) the τ satisfying

(C1) form a group $\mathcal{G}_1$ — the same stabilizer as $\mathcal{T}$, without the condition on $\bar{f}$ — so that span is the $\rho(\mathcal{G}_1)$-submodule of P_D generated by f, and step 3 run without its part (b) returns $\mathcal{G}_1$. The test is then the standard closure computation: apply generators of $\mathcal{G}_1$ to a spanning set repeatedly until the dimension stabilises, which happens after at most N rounds, and test membership of f in the result plus $\mathcal{L}_\star$. It is necessary but not sufficient, since (C2′) also requires the τ_i realizing the combination to be marginally distributed as in (C3).

Second, non-linear recombination. Feigenbaum and Fortnow [FF93] allow the answers to be combined by an arbitrary polynomial-time function, and nothing here decides that condition; the procedure sees only the $\mathbb{Z}_p$-linear fragment that the group encoding makes free.

Third, reparametrizations outside $\mathrm{AGL}_n(\mathbb{Z}_p)$, and reductions that are not reparametrization based at all. A genuine impossibility result for the decisional problem would need a lower-bound argument such as a meta-reduction, as noted after Proposition 6.

Admissible is accordingly a decision procedure for a sufficient condition, and a semi-decision procedure for random self-reducibility: it can certify it, and it cannot refute it.

12 Discussion

Across the four settings the reduction of Theorem 1 is one and the same; only the admissibility conditions move, and they move only because the computable target space $\mathcal{L}_\star$ grows with the available products:

$$\text{none} \subsetneq R \cdot S \text{ (Type 3)} \subsetneq R \cdot S \cup S \cdot S \text{ (Type 2)} \subseteq R \cdot R \text{ (Type 1, single group).}$$

A larger $\mathcal{L}_\star$ makes the covariance condition (C2) easier to meet, so the affine-self-reducible subclass is largest in Type 1 and smallest in Type 3, with the non-bilinear case the degenerate product-free floor. Proposition 6 witnesses the strict gap between Type 2 and Type 3, at a target where the Type-3 reduction fails at every worst-case point — the most the failure locus can ever be, by Remark 3. The gap is one for decisional reductions and for reductions that must work at negligible advantage: Proposition 7 still solves the computational problem in Type 3 at high solver advantage, as it does for every target here. Proposition 5 marks the other boundary, that no monomial target can separate any two settings at all.

Self-reducibility is orthogonal to hardness. The master theorem of [BBG05, Boyen08] makes an assumption hard exactly when $f \notin \mathcal{L}_\star$, whereas Theorem 1 needs the defect $f \circ \tau - \lambda_\tau f$ to lie *in* $\mathcal{L}_\star$. These constrain different objects, so hard yet self-reducible assumptions abound: CDH and DDH (non-bilinear), BDDH (Type 1), and co-CDH and every monomial target (Type 3, Proposition 5) are all of this kind. Quantitatively, Bauer, Fuchsbauer, and Loss [BFL20] show that in the algebraic group model every member of the Uber family is implied by the q-discrete-logarithm assumption, with q governed by the degrees of the defining polynomials, and Rotem [Rotem22] refines q to the maximal degree in which a single variable occurs and gives improved bilinear reductions.

13 Bibliography

[TW87] M. Tompa and H. Woll. Random self-reducibility and zero knowledge interactive proofs of possession of information. In *28th Annual Symposium on Foundations of Computer Science (FOCS 1987)*, pages 472–482. IEEE, 1987.

[BF90] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *7th Annual Symposium on Theoretical Aspects of Computer Science (STACS 1990)*, volume 415 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 1990.

[Lipton91] R. J. Lipton. New directions in testing. In *Distributed Computing and Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–202. AMS, 1991.

[GS92] P. Gemmell and M. Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43(4):169–174, 1992.

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993. Preliminary version in *STOC 1990*, pages 73–83.

[FF93] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22(5):994–1005, 1993.

[Ajtai96] M. Ajtai. Generating hard instances of lattice problems. In *28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 99–108. ACM, 1996.

[Sudan97] M. Sudan. Decoding of Reed–Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[STV01] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

[BBG05] D. Boneh, X. Boyen, and E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 440–456. Springer, 2005.

[BT06] A. Bogdanov and L. Trevisan. Average-case complexity. *Foundations and Trends in Theoretical Computer Science*, 2(1):1–106, 2006.

[Boyen08] X. Boyen. The uber-assumption family: a unified complexity framework for bilinear groups. In *Pairing-Based Cryptography – Pairing 2008*, volume 5209 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2008.

[GPS08] S. D. Galbraith, K. G. Paterson, and N. P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[Regev09] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6):article 34, 2009. Preliminary version in *STOC 2005*, pages 84–93.

[EHKRV13] A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In *Advances in Cryptology – CRYPTO 2013*, volume 8043 of *Lecture Notes in Computer Science*, pages 129–147. Springer, 2013. Journal version: *Journal of Cryptology* 30(1):242–288, 2017.

[BFL20] B. Bauer, G. Fuchsbauer, and J. Loss. A classification of computational assumptions in the algebraic group model. In *Advances in Cryptology – CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pages 121–151. Springer, 2020.

[Rotem22] L. Rotem. Revisiting the uber assumption in the algebraic group model: fine-grained bounds in hidden-order groups and improved reductions in bilinear groups. In *3rd Conference on Information-Theoretic Cryptography (ITC 2022)*, volume 230 of *LIPIcs*, article 13. Schloss Dagstuhl, 2022.