Integer Coordinate Families and Threshold RSA Signatures for Any Prime Exponent

Shivansh Jaiswal and Somitra Sanadhya
{shivanshj24, somitra}@iitk.ac.in

IIT Kanpur

Abstract. At CT-RSA 2026, Ng and Lau introduced (t, n)-integer coordinate families for threshold RSA signatures with $e = 3$, and found some families by heuristic search for $2 \leq t \leq n \leq 9$. They asked whether such families always exist and, if so, can they be constructed efficiently. We answer both of these questions positively. We prove that a (t, n)-integer coordinate family is also a $(t + 1, n)$-integer coordinate family. We construct such families for $t = 2$, and $2 \leq n < 19$, using a polynomial time algorithm in n and its output bit length. Answering another question raised in the Ng-Lau work, we prove super-polynomial lower bounds on coordinates of pairwise families. We also find the minimum families for $n \leq 9$.

Ng and Lau proposed a threshold signature scheme based on integer coordinate families. We show that the Ng–Lau scheme admits trivial forgeries, and has a flawed proof. We give a repaired scheme Π_{IC} which does not suffer from these issues. Finally, we propose an integer secret-sharing scheme Π_{Z} that works for every prime e and every (t, n), without relying on any special coordinate family. We prove that both the schemes Π_{IC} and Π_{Z} are TS-UF-CMA secure under the standard RSA assumption.

Keywords: Threshold RSA signatures · Secret sharing · Integer coordinate families.

1 Introduction

Threshold cryptographic schemes [6] move trust from a single central entity to multiple mutually suspecting entities. Moreover, such schemes allow for the establishment of trust even when some of the participating entities are not available. A (t, n) threshold scheme allows any t or more out of n parties to perform an operation securely. For example, Threshold RSA Signatures [15] can be used to split the signing key among n parties such that any t of them can sign a document, but any group containing $\leq (t - 1)$ parties cannot.

Recently, Ng and Lau [12] studied threshold RSA signatures with public exponent $e = 3$. Their motivation stems from the fact that Intel SGX [9] permits only $e = 3$ as the enclave signing key. They achieve this by choosing special integer evaluation points for Shamir secret sharing [14]. Their arithmetic object is a (t, n)-integer coordinate family in which every Lagrange coefficient needed

by any group of t parties out of n is an integer. This removes division modulo the hidden RSA group order. The work [12] found such algebraic families by heuristic search for $n \leq 9$, together with a few families with $n = 10$. The existence and construction of such families for general t and n was presented as a challenging open problem in [12].

1.1 Contributions

Section 7.1 in [12] poses four questions about such (t, n)-integer coordinate families. The work [12] asks whether such families exist for all t and n, whether they can be constructed efficiently, what lower bounds, if any, exist on their coordinate sizes, and whether they have a deeper algebraic structure. . In this work, we answer the first two questions completely, and answer the third for pairwise families. We also revisit the threshold signature scheme of [12]. Our contributions are as follows:

1. We correct the assertion in Section 4.1 of [12] states that lifting from $t = 2$ to larger thresholds does not hold universally under Definition 4. We show that any (t, n) integer coordinate family is also a $(t+1, n)$ integer coordinate family.
2. We provide an efficient algorithm to produce these coordinate families. Our algorithm allows us to produce such families for every threshold up to $n = 18$. Note that [12] mentioned that their heuristics "do not scale to $n \gtrsim 12$ without theoretical breakthroughs".
3. We provide a super-polynomial lower bound on the maximum coordinate of pairwise coordinate families. We also provide pairwise families with minimum coordinate sizes, and relatively small $(n-1, n)$ coordinate families. We find these families by exhaustive search and brute-force search respectively.
4. We show that the scheme of [12], as stated, admits trivial forgeries, and its proof makes an incorrect assumption. We give a modified scheme Π_{IC} which repairs these issues. We prove Π_{IC} to be TS-UF-CMA secure in the random oracle model under the RSA assumption, for a fixed coordinate family.
5. We propose an alternative scheme Π_{Z} which need not be based on a (t, n) integer coordinate family. It performs almost as well as Π_{IC} on its optimal coordinate families, and outperforms it on suboptimal coordinate families. Finding such integer coordinate families is hard in general, and hence the utility of Π_{Z}. We prove that Π_{Z} is TS-UF-CMA secure in the random oracle model under the RSA assumption.

1.2 Organization

Section 2 provides a brief overview of RSA signatures, threshold signature schemes, Shamir secret sharing and polynomial evaluations, and why ordinary Lagrange interpolation appears to require the secret value $\varphi(N)$ for the RSA modulus N. We then define integer coordinate family, which are the main object used in the rest of the paper. In Section 3, we prove that a $(2, n)$-integer family can be

"lifted" to any (t, n) family. This explains the experimentally observed lifting in [12].

In Section 4.1, we provide a general method for construction of pairwise integer coordinate families. Then we provide a more flexible construction in Section 4.2 and convert it into an efficient algorithm while also analyzing its complexity in Section 4.3.

In [12], a question was raised about the growth in the maximum element of the coordinate families, with the size n of the family. We answer it definitively for pairwise families by proving a super-polynomial lower bound on the maximum coordinate of such a family in Section 5.1. This lower bound does not hold for general families. Therefore, we next explore smallest pairwise families in Section 5.2 by exhaustive search. We also present relatively small $(n-1, n)$ families by brute-force search in Section 5.3.

Ng and Lau [12] had presented a threshold signature scheme. We show a few weaknesses in their scheme and repair it in Section 6.1. However, this scheme still needs to rely on an integer coordinate family. Its performance depends on whether the coordinate family is optimal or not, for which no general construction is known. Therefore we move to a new construction, provided in Section 7. We compare the two proposed schemes in Section 8. Finally, we conclude the paper with a few open problems in Section 9.

1.3 Notations

- $|x|$ represents the absolute value of x.
- $a \mid b$ stands for a is a divisor of b.
- $[n]$ stands for the set $\{1, 2, \ldots, n\}$. We use it to denote the set of parties participating in the protocol.
- The RSA modulus $N = p \cdot q$.
- $\mathbb{Z}$ is the set of integers, and $\mathbb{Z}_N^*$ is the multiplicative group of integers modulo N.
- (t, n): in Threshold sharing schemes, t is the threshold among n parties.
- $a \leftarrow B$: generate an element a from the set B (or algorithm B, which could be randomized)
- $a \xleftarrow{\$} B$: generate a by choosing it uniformly at random from B.
- $\mathrm{v_p}(m)$ (p-adic valuation of a nonzero integer m) is the largest exponent of p in m. That is, the largest integer k such that $p^k \mid m$.
- H: A hash function which is modelled as a random oracle.
- $SD(X, Y)$: Represents the statistical distance between the distributions X and Y.
- κ is the security parameter.
- $\varepsilon(\kappa)$ is a negligible function if, for every constant $c > 0, \varepsilon(\kappa) < \kappa^{-c}$ for sufficiently large κ.

2 Background

Before defining the integer coordinate family, we briefly cover the background which is needed for the rest of the paper.

2.1 The RSA assumption and signatures

Fix a prime public exponent e. Let $\mathsf{GenRSA}_e(1^\kappa)$ be a probabilistic poly-time (PPT) algorithm that outputs (N, p, q, d), where $N = pq$ for distinct odd primes p, q of size $\kappa/2$ bits each, $\gcd(e, \varphi(N)) = 1$, and $ed \equiv 1 \pmod{\varphi(N)}$, with $\varphi(N) = (p-1)(q-1)$. The secret key for the construction is d while the public key is (N, e). Note that an adversary who learns $\varphi(N)$ can compute $p + q = N + 1 - \varphi(N)$, and hence the factors of N. Therefore, $\varphi(N)$ should remain secret along with the factors p and q.

Definition 1 (RSA Assumption). *The advantage of any PPT adversary $\mathcal{A}$ against GenRSA_e is*

$$\mathsf{Adv}^{\mathsf{rsa}}_{\mathsf{GenRSA}_e,\mathcal{A}}(\kappa) = \Pr\left[x^e \equiv y \pmod{N} \;\middle|\; \begin{array}{l} (N,p,q,d) \leftarrow \mathsf{GenRSA}_e(1^\kappa), \\ y \xleftarrow{\$} \mathbb{Z}_N^*,\ x \leftarrow \mathcal{A}(N,e,y) \end{array} \right].$$

The RSA assumption for GenRSA_e asserts that this advantage is negligible in κ for every PPT adversary $\mathcal{A}$.

RSA signatures. Let $H : \{0,1\}^* \to \mathbb{Z}_N^*$ be a hash function, modeled as a random oracle. The signature on a message represented by an integer m, is then $\sigma = (H(m))^d \bmod N$. Verification of the signature σ on message m is successful if $\sigma^e \equiv H(m) \pmod{N}$. This has been adopted from [3], which also discusses its security.

2.2 Threshold signatures

A conventional signing server holds the secret exponent d. Compromising the server results in leaking the key and hence the adversary can create forged signature on any document. A (t, n) threshold signature divides control of signing a document among n parties. Any t of them can sign together, while fewer than t cannot. This is useful when authorization should require several machines, officers, or hardware devices; and when no single device should contain the full signing key. The work [12] focuses on the fixed exponent $e = 3$ due to its use in Intel SGX.

A (t, n)-threshold signature scheme $\Pi = (\mathsf{KGen}, \mathsf{PSign}, \mathsf{Combine}, \mathsf{Verify})$ consists of four algorithms, each of which may query a random oracle H.

- $\mathsf{KGen}(1^\kappa)$ is run by a trusted dealer and outputs a public key pk and shares $y_1, \ldots, y_n$.
- $\mathsf{PSign}(pk, i, y_i, M)$ outputs a partial signature σ_i. A partial signature σ_i is party i's contribution to a signature on M, computed from its own share of the key. It does not take the signing set as input.[1]
- $\mathsf{Combine}(pk, M, S, (\sigma_i)_{i \in S})$, for $S \subseteq [n]$ with $|S| \geq t$, outputs a signature σ.

[1] Note that the scheme in [12] requires the signing set to be known in advance for computing the partial signature.

$\mathsf{Exp}^{\mathsf{ts\text{-}uf\text{-}cma}}_{\Pi,\mathcal{A}}(\kappa)$

$(C, \mathsf{st}) \leftarrow \mathcal{A}(1^\kappa)$
if $|C| \geq t$ **then return** 0
$(pk, y_1, \ldots, y_n) \leftarrow \mathsf{KGen}^H(1^\kappa)$
$Q \leftarrow \emptyset$
$(M^*, \sigma^*) \leftarrow \mathcal{A}^{H,\mathcal{O}}(\mathsf{st}, pk, (y_c)_{c \in C})$
return $\mathsf{Verify}^H(pk, M^*, \sigma^*) \wedge M^* \notin Q$

Oracle $\mathcal{O}(i, M)$

if $i \in C$ **then return** $\perp$
$Q \leftarrow Q \cup \{M\}$
return $\mathsf{PSign}^H(pk, i, y_i, M)$

Oracle $H(M)$

if $\mathsf{T}[M] = \perp$ **then** $\mathsf{T}[M] \xleftarrow{\$} \mathbb{Z}_N^*$
return $\mathsf{T}[M]$

Fig. 1. Threshold unforgeability under static corruptions. C represents the set of corrupted parties, which is chosen by the adversary at the initiation of the protocol. The table T is initially empty. H is the hash oracle, and $\mathcal{O}$ is the partial-signing oracle.

– $\mathsf{Verify}(pk, M, \sigma)$ outputs the bit 1 (resp. 0) when the signature σ is valid (resp. invalid) for the message M.

Π is *correct* if for every κ, every random oracle H, every output $(pk, y_1, \ldots, y_n)$ of $\mathsf{KGen}(1^\kappa)$, every M and every $S \subseteq [n]$ with $|S| \geq t$,

$$\mathsf{Verify}\big(pk, M, \mathsf{Combine}(pk, M, S, (\mathsf{PSign}(pk, i, y_i, M))_{i \in S})\big) = 1.$$

We formalize threshold unforgeability under static corruptions using a variant of Shoup's model [15, Section 2]. In our experiment, a forgery must concern a message for which no honest party has been queried for a partial signature.

Definition 2 (TS-UF-CMA). *The advantage of an adversary $\mathcal{A}$ against Π is $\mathsf{Adv}^{\mathsf{ts\text{-}uf\text{-}cma}}_{\Pi,\mathcal{A}}(\kappa) = \Pr[\mathsf{Exp}^{\mathsf{ts\text{-}uf\text{-}cma}}_{\Pi,\mathcal{A}}(\kappa) = 1]$, with the experiment of Fig. 1. Π is TS-UF-CMA secure in the random oracle model if this advantage is negligible for every PPT $\mathcal{A}$.*

The adversary chooses which honest party answers each query, and may combine the partial signatures itself. A message enters the queried set even if only one honest party has been queried on it. This is needed to prevent an adversary who is controlling $(t-1)$ corrupted parties to query a partial signature from an honest party, and thus create a valid signature. We consider static corruptions, where the adversary selects the set of corrupted parties C before the key generation algorithm is executed. The experiment measures unforgeability. It does not provide robustness, that is, the ability to detect malformed partial signatures and still produce a valid signature.

2.3 Shamir secret sharing

Let $\mathcal{F}$ be a finite field. Shamir's secret sharing technique [14] allows sharing a secret value $d \in \mathcal{F}$ among n participants such that any t or more of them can reconstruct the secret but fewer than t cannot. Let the set of all parties be

$\{1, 2, \ldots, n\}$. The dealer chooses a polynomial of degree at most $(t-1)$ over $\mathcal{F}$ with constant being d, say

$$f(z) = d + a_1 z + \cdots + a_{t-1} z^{t-1}. \tag{1}$$

The dealer then gives party i the share $y_i = f(x_i)$ at a public, distinct evaluation point x_i. Any t evaluations determine the polynomial's constant term $f(0) = d$. For a coalition indexed by the set of parties $S \subseteq [n]$ where $|S| \geq t$, Lagrange interpolation in $\mathcal{F}$ gives the following.

$$d = \sum_{i \in S} \lambda_{i,S} \cdot y_i \qquad \text{where} \quad \lambda_{i,S} = \prod_{j \in S \setminus \{i\}} \frac{x_j}{x_j - x_i}. \tag{2}$$

For a set of parties $S \subseteq [n]$ and $i \in S$, the Lagrange basis polynomial is

$$L_{i,S}(z) = \prod_{j \in S \setminus \{i\}} \frac{z - x_j}{x_i - x_j}. \tag{3}$$

The reconstruction coefficient is $\lambda_{i,S} = L_{i,S}(0)$. Note that signature computation requires evaluating m^d, and Equation 2 above reconstructs d as a sum of many terms. The exponentiation turns this linear reconstruction into a product of partial RSA signatures. This is the basic ingredient of a threshold signature scheme from secret sharing, e.g. Shoup's threshold-signature scheme [15].

Although Shamir secret sharing is usually implemented over a finite field $\mathcal{F}$, it is possible to use it over a ring under certain conditions. Essentially, we need the pairwise differences of the evaluation points to be invertible in the ring. This would permit Lagrange interpolation to compute the secret d as shown above. Using the RSA modulus N and considering evaluation points x_i, we need $(x_i - x_j) \in \mathbb{Z}^*_{\varphi(N)}$ to be able to compute $\lambda_{i,S}$. Note that Shamir secret sharing can be made to work for commutative rings $\mathbb{Z}/p^k\mathbb{Z}$ for any prime p [1], and non-commutative rings $\mathcal{M}_{m \times m}(\mathbb{Z}_{2^k})$ of $m \times m$ matrices over $\mathbb{Z}_{2^k}$ [7].

2.4 The (t, n) integer coordinate family

Equation 2 requires division by terms of the type $(x_j - x_i)$. If the coefficients are computed in the ring modulo $\varphi(N)$, the parties will need to know $\varphi(N)$. However, revealing $\varphi(N)$ would trivially allow factoring N. To avoid this difficulty, [12] suggests choosing the public points x_i so that every coefficient in Equation 2 is an integer. Then the parties can use ordinary integer arithmetic modulo the public value N. This approach of [12] motivates the following definition of an "Integer Coordinate Family".

Definition 3 (Integer Coordinate Family (adapted from [12])). *Let $2 \leq t \leq n$. A set $X = \{x_1, \ldots, x_n\} \subset \mathbb{Z}$ of distinct nonzero integers is a (t, n)-integer coordinate family if*

$$\lambda_{i,S} = \prod_{j \in S \setminus \{i\}} \frac{x_j}{x_j - x_i} \in \mathbb{Z}. \tag{4}$$

for every coalition $S \subseteq [n]$ with $|S| = t$, and every $i \in S$.

Note that the above definition is almost the same as the Definition 4 of [12]. We only add the requirement of x_i's to be nonzero. It seems to have been implicitly assumed in [12] as it is obvious that if any x_i is 0 then the corresponding share $f(x_i)$ will be the secret d itself.

Definition 4 (Pairwise and universal integer coordinate families). *An n-element integer coordinate family is pairwise if it is a $(2, n)$-integer coordinate family. It is universal if the same set is a (t, n)-integer family for every $2 \leq t \leq n$.*

Definition 5 (Minimum integer coordinate families). *Let $\mathcal{F}_{t,n}$ be the class of (t, n)-integer families, and let*

$$g(t, n) = \min_{X \in \mathcal{F}_{t,n}} \max_{x \in X} |x|, \qquad s(n) = g(2, n).$$

A family $X \in \mathcal{F}_{t,n}$ is minimum if $\max_{x \in X} |x| = g(t, n)$.

We use *(t, n)-integer family* as a short form of (t, n)-integer coordinate family, *pairwise family* as a short form of pairwise integer coordinate family, and *minimum family* as a short form of minimum integer coordinate family. Note that definition 4 requires a set X where $\forall x, y \in X, x \neq y$, $(x - y)|x$ and $(x - y)|y$, both implying each other.[2]

3 Lifting coordinate families to higher thresholds

We first show that pairwise families have a common sign. We then prove that if a coordinate set is a (t, n)-integer family, then the same coordinate set is also a $(t + 1, n)$ family.

Lemma 6 (Common Sign). *Let X be a pairwise integer coordinate family. Then, either every element of X is positive or every element is negative.*

Proof. Suppose that $x, y \in X$ satisfy $x < 0 < y$, then the condition (4) for $t = 2$ gives $y - x|x$. However, $|y - x| = |x| + |y| > |x|$, but a non-zero divisor of x cannot have an absolute value greater than $|x|$. Thus, all elements have the same sign. □

Negating every coordinate preserves the pairwise divisibility. Thus a negative pairwise coordinate family can be switched for a positive coordinate family, keeping the same maximum absolute coordinate.

Theorem 7 (Threshold monotonicity). *If $2 \leq t < n$, then every (t, n)-integer family is also a $(t + 1, n)$-integer family. This allows trivially lifting the family to higher thresholds.*

[2] If $y = k(x - y)$ for some integer k then $x = (x - y) + y = (k + 1)(x - y)$.

Proof. Let X be a (t, n)-integer family, let $S \subseteq [n]$ with $|S| = t + 1$, and let $i \in S$. For each $k \in S \setminus \{i\}$, the coalition $S \setminus \{k\}$ has size t and contains i, so $\lambda_{i,S\setminus\{k\}} = \lambda_{i,S} \cdot \frac{x_k - x_i}{x_k} \in \mathbb{Z}$. Taking the product over all t terms gives

$$\prod_{k\in S\setminus\{i\}} \lambda_{i,S\setminus\{k\}} = \lambda_{i,S}^{t} \prod_{k\in S\setminus\{i\}} \frac{x_k - x_i}{x_k} = \lambda_{i,S}^{t} \cdot \lambda_{i,S}^{-1} = \lambda_{i,S}^{t-1}$$

Hence $\lambda_{i,S}^{t-1} \in \mathbb{Z}$, which implies $\lambda_{i,S} \in \mathbb{Z}$. Since S and i were arbitrary, X is a $(t + 1, n)$-integer family, that is, $\mathcal{F}_{t,n} \subseteq \mathcal{F}_{t+1,n}$.

□

Corollary 8. *If X is a $(2, n)$-integer coordinate family, then X is a (t, n)-integer coordinate family for every t with $2 \leq t \leq n$.*

Section 4.1 in [12] states: "This avoids dependence on $(2, n)$ families, as lifting (see below) from $t = 2$ to $t \geq 3$ does not hold universally." Under Definition 4 in [12], Corollary 8 shows that the implication *does* hold universally. The remark may describe the behavior or organization of their search procedure rather than the mathematical property.

4 Existence and constructions of pairwise coordinate families

We now describe a method to construct universal (t, n)-integer coordinate families. First, we only consider the sets with positive integers as these are sufficient by Lemma 6. Further, by Corollary 8, a pairwise integer coordinate family is universal. Therefore, we construct only pairwise families containing positive integers. We give two constructions. First to prove existence, and second to allow for a more efficient search. Finally, we describe an algorithm based on the second construction. Both of our constructions extend a $(2, n)$-integer coordinate family to a $(2, n + 1)$-integer coordinate family.

4.1 Coordinate extension by LCM translation

The first construction translates the coordinates by the LCM of the coordinates.

Proposition 9 (LCM Translation). *Define a sequence of sets $(A_n)_{n\geq 2}$ recursively by $A_2 = \{1, 2\}$, and for each $n \geq 2$*

$$l_n = \operatorname{lcm}\{a : a \in A_n\}, \qquad A_{n+1} = \{l_n\} \cup \{l_n + a : a \in A_n\}. \tag{5}$$

Then for every $n \geq 2$, the set A_n is a pairwise integer family of cardinality n.

Proof. It is easy to see that the construction A_n satisfies the condition of Definition 3. The detailed proof using induction is provided in Appendix A.1 for completeness.

Proposition 9 itself can be used as an algorithm. It computes one lcm and one coordinate translation per step but the coordinates of the families it outputs may grow quickly. The next construction makes this more flexible.

4.2 CRT-based construction

In Proposition 9, the coordinates are translated by l_n, the LCM of the coordinates. We now develop a more flexible approach by separating the "common base" (e.g. l_n in Proposition 9) from the "offset" (e.g. a in Proposition 9).

Let X be a pairwise integer coordinate family of cardinality n, where the elements are in ascending order. Write each coordinate $x_i \in X$ as a *common base* x_1 plus an *offset* $d_i = (x_i - x_1)$:

$$X = \{x_1 < \cdots < x_n\} = \{x_1 + d : \ d \in D\}, \text{ where } D = \{0 = d_1 < \cdots < d_n\}.$$

Consider any two arbitrary coordinates $x_i, x_j \in X$. Then $(x_i - x_j) = (d_i - d_j)$ with base x_1, i.e. differences of coordinates are differences of their offsets.

For these coordinates, their absolute difference $|d_i - d_j|$ divides both the coordinates x_i, x_j exactly when

$$x_1 \equiv -d_i \equiv -d_j \pmod{|d_i - d_j|}. \tag{6}$$

Denote the system of all the congruences of the form (6) for all pairs in D by $\Gamma(D)$. Then every solution $x_1 > 0$ gives a positive pairwise family $\{x_1 + d : d \in D\}$. The base x_1 however, is just one solution to these congruences. Every positive solution z of $\Gamma(D)$ yields a pairwise integer coordinate family $\{z + d : d \in D\}$ with the same offsets D. We use the following lemma to find such a solution z.

Lemma 10 (Generalized CRT [11, Theorem 3.12]). *Let $n_1, \ldots, n_k$ be positive integers and let $a_1, \ldots, a_k$ be any integers. Then the simultaneous congruences*

$$x \equiv a_1 \pmod{n_1}, \ \ldots, \ x \equiv a_k \pmod{n_k}$$

have a solution x iff $\gcd(n_i, n_j) \mid (a_i - a_j)$ whenever $i \neq j$. When this condition is satisfied, the general solution forms a single congruence class $\pmod{P}$, where P is the least common multiple of $n_1, \ldots, n_k$.

We wish to obtain a pairwise integer coordinate family of cardinality $n + 1$ from a family of cardinality n. We achieve this by introducing a new offset $q \notin D$ to D, which keeps the system of congruences $\Gamma(\{q\} \cup D)$ solvable.

Let $M = \operatorname{lcm}\{|d - d'| : d, d' \in D, \ d \neq d'\}$, with $M = 1$ if $D = \{0\}$. That is, M is the lcm of the differences of offsets. By Lemma 10, every solvable system $\Gamma(D)$ has one solution class $x_1 \equiv \ r \pmod{M}$, with the representative $r \in [0, M)$. A new offset $q \notin D$ is *compatible* with D when $\Gamma(D \cup \{q\})$ is solvable. The next theorem characterizes compatible offsets and proves that an extension always exists. Unlike the LCM translation, which forces the next base to be l_n, we can first choose an offset compatible with D, and then find the base z as the solution of the updated congruence system.

Theorem 11 (CRT offset extension). *Let D be an offset set which keeps $\Gamma(D)$ solvable. Let $x_1 \equiv r \pmod{M}$, $0 \leq r < M$, be the solution class of $\Gamma(D)$.*

(i) *A positive integer $q \notin D$ is compatible with D if and only if*

$$\gcd(M, |q-d|) \mid (r+q) \qquad \textit{for every } d \in D. \tag{7}$$

(ii) *Every positive integer $q \notin D$ of the form $q = kM - r$ is compatible with D. In particular, the following offset is always compatible:*

$$q = \begin{cases} M - r, & M - r \notin D, \\ 2M - r, & M - r \in D. \end{cases} \tag{8}$$

Proof. (i) Adding q to the set D adds to the system $\Gamma(D)$ the congruences

$$z \equiv -q \pmod{|q-d|} \quad \text{where } d \in D. \tag{9}$$

We want the system $\Gamma(\{q\} \cup D)$ to be solvable, which happens when $\Gamma(D)$ and (9) are simultaneously solvable. By Lemma 10, solving the congruences given by $\Gamma(D)$ and (9) above is equivalent to solving $z \equiv r \pmod{M}$ and (9) together.

Any two congruences in (9) have the same residue $-q$, so they are always compatible. The pair $z \equiv r \pmod{M}$ and $z \equiv -q \pmod{|q-d|}$ is compatible if and only if $\gcd(M, |q-d|) \mid (r+q)$, which is the condition (7).

(ii) If the new offset is $q = kM - r$, then $M \mid r+q$. For some $d \in D$, $\gcd(M, |q-d|)$ divides $r+q$. Now since this works for every d, $\gcd(M, |q-d|)$ divides $r+q$ for all $d \in D$. Therefore offset q is compatible with D by (i).

For $d \in D \setminus \{0\}$, $d = d - 0$ is a difference of offsets, so $d \mid M$ and $d \leq M$. Since $0 \leq r < M$, both $M - r$ and $2M - r$ are positive, and $2M - r > M \geq \max(D)$. Therefore, $2M - r \notin D$. Hence (8) is a positive offset $\notin D$ of the form $kM - r$. □

Starting from $D = \{0\}$, where $M = 1$ and $r = 0$, and applying Theorem 11(ii) repeatedly gives compatible offset sets of every size. Thus, Theorem 11 is straightforward to implement as an algorithm.

4.3 Algorithm to construct pairwise families

Algorithm 1 stores the offset set $D \ni 0$, and the congruence class $z \equiv r \bmod (M)$ of $\Gamma(D)$ as a pair (r, M), where $0 \leq r < M$. An *unseeded* run starts from $D = \{0\}$, and $(r, M) = (0, 1)$. Since Theorem 11 can extend any pairwise coordinate family, once decomposed into its offsets, we can instead start from a "seeded" family. A *seeded* run starts from any verified pairwise coordinate family of size s, written as offsets D_s with its congruence class $\Gamma(D_s)$ represented as (r_s, M_s). At each iteration, we search for, and add one compatible offset. When an offset q is added, it *merges* each new set of congruences (9) into (r, M) by Lemma 10. Adding an offset q to a set of size m merges m congruences, and costs one extended gcd for every merge operation. If we start from a seed of size s, and aim to construct a family of size n then exactly $\binom{n}{2} - \binom{s}{2}$ merges will be needed.

Algorithm 1 `construct`(n, D_s, r_s, M_s): seeded CRT extension

Input: $n \geq 2$, $1 \leq s \leq n$, seed $D_s \ni 0$, representative (r_s, M_s) of $\Gamma(D_s)$, bound $T \geq 0$ on the number of candidates tested

$D \leftarrow D_s$, $(r, M) \leftarrow (r_s, M_s)$
for $m = s, \ldots, n-1$ **do**
  $q \leftarrow \bot$
  for each of at most T candidates c **do**
    if $c \notin D$ and $\gcd(M, |c-e|) \mid r + c$ for every $e \in D$ **then**
      $q \leftarrow c$; **break**
  if $q = \bot$ **then**
    $q \leftarrow M - r$; **if** $q \in D$ **then** $q \leftarrow 2M - r$ ▷ Theorem 11(ii)
  for each $e \in D$ **do**
    merge $z \equiv -q \pmod{|q-e|}$ into (r, M)
  $D \leftarrow D \cup \{q\}$
$z \leftarrow r$ if $r > 0$, else $z \leftarrow M$
return $\{z + d : d \in D\}$

The fallback $q = M - r$ or $q = 2M - r$ of Theorem 11(ii) always works, but using it at every step makes the coordinates grow quickly. Therefore, before falling back, the algorithm tests up to T smaller candidates against the condition (7). Appendix B describes how these candidates are chosen. This search is heuristic, but it only affects the size of the output, and the fallback is always compatible.

For seeded runs, we use the offsets of the (2,9)-integer family found by exhaustive search, provided in Table 1,

$$D_9 = \{0, 1680, 1800, 1872, 1890, 2000, 2025, 2100, 2160\}.$$

In our experiments we initially find a measurable difference in the sizes of the coordinates between the seeded and unseeded runs. But the difference becomes negligible as n increases, as can be seen in Fig. 2. Our runs with fixed parameters, first use the fallback offset of Theorem 11(ii) at $n = 17$ in the unseeded run and $n = 19$ in the seeded run, which explains the sudden jump observable in Fig. 2.

Existence of an algorithm which generates integer families with complexity polynomial in both n and b was posed as an open problem in [12]. Algorithm 1 meets this, for fixed parameters.

Proposition 12 (Running time). *Fix the starting seed and the search parameters. On input n, let X be the family output by Algorithm 1, and let b be its largest coordinate's bit length. The algorithm uses $O(n^7 b^2)$ bit operations and $O(n^2 b)$ bits of memory.*

The proof of Proposition 12 is in Appendix B. The bound is polynomial in n and in the output bit-size b combined. We do not know of a polynomial bound on b in terms of n. However, by Corollary 14, every construction needs $b = \Omega(n \log n)$.

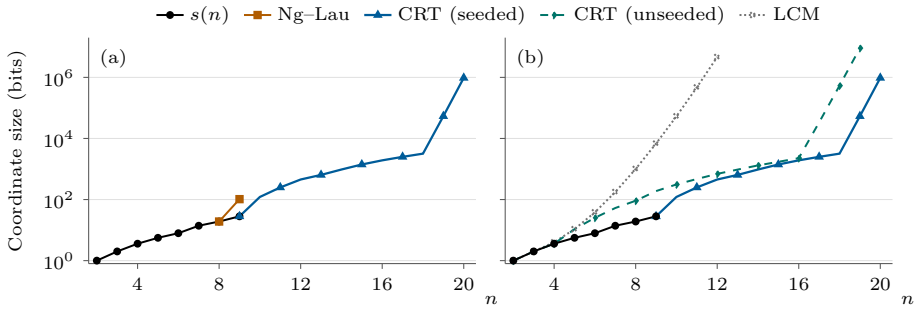

Fig. 2. Pairwise coordinate sizes $\log_2 \max_i |x_i|$ ($t = 2$). (a) Ng–Lau, seeded CRT, and $s(n)$. (b) LCM and CRT constructions. Seeded CRT uses the minimum $(2, 9)$-integer family.

5 Lower bound and exhaustive search on $s(n)$

Ng and Lau asked if their observed coordinate growth rate is fundamental or a consequence of heuristic search [12]. We show that for pairwise integer coordinate families, the super-polynomial growth of the largest coordinate is unavoidable.

5.1 Asymptotic Growth

Theorem 13 (Occupancy bound). *Let X be a pairwise integer coordinate family, p be a prime and integer $k \geq 0$. Then at most $p - 1$ elements $x \in X$ satisfy $\mathrm{v_p}(x) = k$.*

Proof. Fix p and k. Let $x, y \in X, x \neq y$ such that $\mathrm{v_p}(x) = \mathrm{v_p}(y) = k$. Let $x = p^k u$, $y = p^k v$, then $p \nmid uv$. Since $x - y \mid x$, we have $\mathrm{v_p}(x - y) \leq k$. But $x - y = p^k(u - v)$, so $p \nmid u - v$, which implies $u \not\equiv v \pmod p$. Thus for $x \in X$ with $\mathrm{v_p}(x) = k$, the residue $x/p^k \pmod p$ is distinct and nonzero, and there are only $p - 1$ such residues. □

For example, at $p = 2$, the above theorem states that at most one element has each 2-adic valuation $k = 0, 1, 2, \ldots$. At $n = 4$, the optimal pairwise family $\{6, 8, 9, 12\}$ has 2-adic valuations $\{1, 3, 0, 2\}$, one at each level from 0 to 3. In general, for any prime p, at most $p - 1$ elements can have $\mathrm{v_p}(x) = 0$. The rest $n-(p-1)$ elements must be multiples of p. Similarly, $n-2(p-1)$ elements must be multiples of p^2, and so on. Thus for a pairwise family, if its size $n > m(p-1)$, it must have elements divisible by p^m.

Corollary 14. *There is an absolute constant $C > 0$ such that, every pairwise integer coordinate family with n elements contains an element x with $\log |x| \geq \frac{1}{2} n \log n - Cn$. In particular, $s(n) \geq n^{n/2} e^{-Cn}$, so coordinates of polynomial size in n are impossible.*

Proof. Let $T = \max_{x \in X} |x|$. Fix a prime $p \leq n$, and consider the sum $\sum_{x \in X} \mathrm{v_p}(x)$ Order these $\mathrm{v_p}(x)$ elements in increasing order of p-adic valuations. By Theorem

13, at most $k(p-1)$ of them are below k. Then, the i-th smallest ($i = 0, \ldots, n-1$) p-adic valuation is at least $\lfloor i/(p-1) \rfloor$. Using this,

$$v_p(\prod_x x) = \sum_{x \in X} \mathrm{v_p}(x) \geq \sum_{i=0}^{n-1} \lfloor \frac{i}{p-1} \rfloor \geq \frac{n(n-1)}{2(p-1)} - n \geq \frac{n(n-1)}{2p} - n \geq \frac{n^2}{2p} - \frac{3n}{2}.$$

Since $\sum_x \log|x| = \sum_p \big(\sum_x \mathrm{v_p}(x)\big) \log p$, and the inner sum is nonnegative, dropping the primes $p > n$ and inserting the bound above yields

$$n \log T \geq \sum_{x \in X} \log|x| \geq \sum_{p \leq n} \Big(\frac{n^2}{2p} - \frac{3n}{2}\Big) \log p = \frac{n^2}{2} \sum_{p \leq n} \frac{\log p}{p} - \frac{3n}{2} \sum_{p \leq n} \log p.$$

By Apostol [2, Theorem 4.10], for $n \geq 2$, $\sum_{p \leq n} (\log p)/p = \log n + O(1)$ and $\sum_{p \leq n} \log p = O(n)$. This gives us

$$\log T \geq \frac{1}{2} n \log n - O(n). \tag{10}$$

This provides us with a lower bound on the maximum element of every $(2, n)$-integer coordinate family.

Since $s(n)$ is the minimum of the maximum element over all pairwise integer coordinate families, and (10) gives a lower bound on maximum element of any pairwise family, we obtain a lower bound on $s(n)$. Hence, $\log s(n) \geq \frac{1}{2} n \log n - Cn$, which implies $s(n) \geq n^{n/2} e^{-Cn}$. □

Corollary 14 answers the question asked by Ng and Lau [12] regarding the growth in the size of coordinates for only pairwise (and thus universal) families. Our result is that $s(n) = g(2, n)$ is indeed super-polynomial in n. The argument does not apply to all (t, n)-integer coordinate families for $t > 2$. At higher thresholds, as reported in heuristic search in [12], and Section 5.3, substantially smaller coordinates can be obtained.

5.2 $(2, n)$-families with smallest maximum coordinate at given n

The values $s(n)$ in Table 1 are the families with the smallest maximum coordinate for $2 \leq n \leq 9$. These values are exact. By Lemma 6, it suffices to search for families of positive integers. An exhaustive search over positive families shows that no pairwise family of size n has a smaller maximum coordinate element. The search procedure transforms the problem to that of a clique search to find these families with the smallest maximum at a given n. The details of the algorithm are given in Appendix C.

5.3 $(n-1, n)$-families found by brute-force

We also attempt to find $(n-1, n)$ families, which we obtain directly through a brute-force search. We consider sets with maximum element T. For every

Table 1. Minimum pairwise families found by exhaustive search for $2 \leq n \leq 9$. The maximum element in X is $s(n)$.

n	$s(n)$	Minimum pairwise coordinate family X
2	2	$\{1, 2\}$
3	4	$\{2, 3, 4\}$
4	12	$\{6, 8, 9, 12\}$
5	48	$\{36, 40, 42, 45, 48\}$
6	240	$\{210, 216, 220, 224, 225, 240\}$
7	15120	$\{14976, 14980, 14994, 15000, 15008, 15015, 15120\}$
8	554400	$\{552720, 552825, 552960, 553000, 553014, 553140, 553280, 554400\}$
9	309584160	{309582000,309583680,309583800,309583872,309583890, 309584000,309584025,309584100,309584160}

Table 2. Verified $(n-1, n)$-integer coordinate families.

n	max X	X at $t = n - 1$
3	4	$\{2, 3, 4\}$
4	6	$\{3, 4, 5, 6\}$
5	12	$\{6, 8, 9, 10, 12\}$
6	10	$\{4, 5, 6, 8, 9, 10\}$
7	15	$\{7, 8, 9, 10, 12, 14, 15\}$
8	21	$\{14, 15, 16, 17, 18, 19, 20, 21\}$
9	40	$\{28, 30, 32, 33, 34, 35, 36, 38, 40\}$
10	54	$\{40, 42, 44, 45, 46, 48, 49, 50, 52, 54\}$
11	55	$\{39, 40, 42, 44, 45, 48, 49, 50, 52, 54, 55\}$
12	55	$\{39, 40, 42, 43, 44, 45, 48, 49, 50, 52, 54, 55\}$
13	55	$\{39, 40, 42, 43, 44, 45, 48, 49, 50, 51, 52, 54, 55\}$
14	55	$\{39, 40, 42, 43, 44, 45, 46, 48, 49, 50, 51, 52, 54, 55\}$
15	78	$\{63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 74, 75, 76, 77, 78\}$
16	210	$\{195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210\}$
17	209	$\{187, 188, 189, 190, 191, 192, 194, 195, 196, 198, 200, 203, 204, 205, 207, 208, 209\}$
18	209	$\{187, 188, 189, 190, 191, 192, 195, 196, 198, 199, 200, 203, 204, 205, 206, 207, 208, 209\}$
19	209	$\{187, 188, 189, 190, 191, 192, 195, 196, 197, 198, 200, 201, 203, 204, 205, 206, 207, 208, 209\}$
20	209	$\{184, 185, 186, 187, 188, 189, 190, 192, 195, 196, 197, 198, 200, 201, 203, 204, 205, 207, 208, 209\}$

$T = n, n+1, \ldots,$ we test every n-element subset of $\{1, \ldots, T\}$ that contains T. Thus all candidate sets with the maximum element T are checked. This procedure works well for $n \leq 9$. However, for larger n, the number of candidate sets becomes extremely large. For these, we attempted some heuristics and let the search run for a few hours on a standard laptop.

To check a candidate set, we calculate the reconstruction Lagrange coefficients for all $(n-1)$ sized subsets. The candidate set is accepted only if every Lagrange coefficient in every $(n-1)$ sized subset is an integer. The result of this search is provided in Table 2. Unlike Section 5.2, we do not claim these families to be minimum.

6 The security of the Ng–Lau scheme

We now return to the cryptographic application which motivated this work. Integer coordinate families make the reconstruction coefficients integral, but this alone does not establish threshold unforgeability of RSA signatures. We first examine the security argument of [12], and then present a modified scheme Π_{IC}. We also prove its security under certain conditions.

6.1 The scheme

The scheme of [12] uses a (t, n) integer family X.

Dealing. A trusted dealer samples $f(Z) = d + \sum_{j=1}^{t-1} a_j Z^j$ with $a_j \xleftarrow{\$} \mathbb{Z}_{\varphi(N)}$ and gives party i the share $y_i = f(x_i) \bmod \varphi(N)$.

Signing. To sign a message representative $m \in \mathbb{Z}_N^*$, a coalition $S \subseteq [n]$ of size t computes $\sigma_i = m^{\lambda_{i,S} y_i}$.

Combining. The signature is $\prod_{i \in S} \sigma_i = m^d$.

Ng and Lau note that the scheme "is compatible with standard RSA message encoding (PSS, PKCS#1, etc)" and requires only $\gcd(m, N) = 1$. Although the scheme's correctness holds, we identify certain flaws.

The work [12] claims EUF-CMA security for their construction under the standard RSA assumption [12, Theorem 1]. Theorem 1, as stated is false and the reduction proof is invalid. The scheme operates on raw $m \in \mathbb{Z}_N^*$ without any encoding. Consequently, the scheme is vulnerable to trivial forgeries. For example, the adversary wins the EUF-CMA game with $e = 3$ by outputting $(M^*, \sigma^*) = (8, 2)$, since $(\sigma^*)^e = 2^3 = 8 = M^* \pmod N$.

Even with a hash-based encoding, the proof relies on an incorrect uniformity assumption. The reduction assumes that the corrupted shares are "uniformly distributed in $\mathbb{Z}_{\varphi(N)}^{|C|}$ independent of d", which is incorrect. We show that the corrupted shares need not be uniform on the full residue space, although they may be independent of d. The exact distribution is given in Lemma 16 with the proof of the lemma available in Appendix D.1.

To resolve these gaps in their scheme, we propose a modified scheme and provide reduction proof for its security in the random oracle model under the standard RSA assumption.

Public constants and the modified scheme Let the set of corrupted parties be denoted by indices $C \subseteq [n]$ with $|C| = t - 1$. For a coordinate family X, party i has the share (x_i, y_i) where $x_i \in X$ and $y_i = f(x_i)$ is the share of the secret. Define $C_0 = C \cup \{0\}$ as the extended index set, and set $x_0 := 0$. This means that party 0 gets the secret d as its share $f(0)$. $L_{i,C_0}(z)$ represents the Lagrange basis polynomial from (3), where $i \in C_0$.

Let $L_{i,C_0}(x_k)$ be the Lagrange coefficient in its reduced fraction form. Let U_C be the set containing the denominators of the coefficients $L_{i,C_0}(x_k)$ for all $i \in C_0$,

$\mathsf{KGen}(1^\kappa)$

$(N, p, q, d) \leftarrow \mathsf{GenRSA}_e(1^\kappa);\ \varphi = (p-1)(q-1)$

$a_1, \ldots, a_{t-1} \xleftarrow{\$} \mathbb{Z}_\varphi$

$f(Z) := d + \sum_{j=1}^{t-1} a_j Z^j$

$pk \leftarrow (N, e, X, n, t)$

for $i \in [n]$ **do** $y_i \leftarrow f(x_i) \bmod \varphi, \quad x_i \in X$

return $(pk, y_1, \ldots, y_n)$

$\mathsf{PSign}(pk, i, y_i, M)$

return $\sigma_i \leftarrow H(M)^{R_X y_i} \bmod N$

$\mathsf{Combine}(pk, M, S, (\sigma_i)_{i \in S})$

$w \leftarrow \prod_{i \in S} \sigma_i^{\lambda_{i,S}} \bmod N$

return $\sigma \leftarrow w^\alpha\, H(M)^\beta \bmod N$

$\mathsf{Verify}(pk, M, \sigma)$

return $\sigma^e \equiv H(M) \pmod N$

Fig. 3. The scheme for a prime e and a (t, n)-integer family X, with $H : \{0,1\}^* \to \mathbb{Z}_N^*$. The constants $\lambda_{i,S}$, R_X, α, β depend only on (e, X). Negative exponents are computed by inversion in $\mathbb{Z}_N^*$.

and all honest parties $k \in [n] \setminus C$. Let D_X be the least common multiple of all the elements of sets U_C for all C. The advantage of defining D_X this way is that when we multiply D_X by $L_{i,C_0}(x_k)$ the denominator is cancelled irrespective of the set of corrupt parties.

For the RSA public parameter e, we write $D_X = e^v R_X$, where v is the maximum exponent of e in D_X (i.e. $e^v \mid D_X$ but $e^{v+1} \nmid D_X$). Finally, choose public integers α, β with $\alpha R_X + e\beta = 1$ by Bezout's lemma. These constants depend only on (e, X, t) which are public.

With the public constants given above, Figure 3 specifies the modified scheme Π_{IC}. Let x be the message representative. Then our modifications are

1. *Lagrange coefficients in the combiner.* In the scheme of [12], party k signs with $x^{\lambda_{k,S} \cdot y_k}$, which requires each party to know the signing set S in advance. We move $\lambda_{k,S}$ to the combiner, as in Shoup's scheme [15], so that party k outputs x^{y_k}. This removes the need for coordination among the signers before the signing.
2. *Clearing factor.* Next, we modify the partial signature from x^{y_i} to $x^{R_X y_i}$. The combiner then computes $w = \prod_{i \in S} \sigma_i^{\lambda_{i,S}} = x^{R_X \sum \lambda_{i,S} y_i} \equiv x^{R_X d} \bmod N$. The product w has $R_X d$ in the exponent, instead of just d. The combiner removes R_X using the public relation $\alpha R_X + e\beta = 1$. The constant D_X is then used to simulate the honest partial signatures.
3. *Full-domain hash.* We encode the messages with a full-domain hash $H : \{0,1\}^* \to \mathbb{Z}_N^*$, modeled as a random oracle, and verify signatures by checking $\sigma^e = H(M)$. The message representative x is thus, $x = H(M)$.

Correctness of the modified scheme Let $s = H(M) \in \mathbb{Z}_N^*$. Since X is a (t, n)-integer family, the coefficients $\lambda_{i,S}$ are integers, and $\sum_{i \in S} \lambda_{i,S} y_i \equiv d \bmod \varphi(N)$.

The combiner computes

$$w \equiv \prod_{i \in S} \sigma_i^{\lambda_{i,S}} \equiv x^{R_X \sum_i \lambda_{i,S} y_i} \equiv x^{R_X d} \bmod N.$$

Using $ed \equiv 1 \bmod \varphi(N)$ and the relation $\alpha R_X + e\beta = 1$, given by public integers α, β, we get

$$\sigma^e \equiv x^{e\alpha R_X d + e\beta} \equiv x^{\alpha R_X + e\beta} \equiv x \bmod N.$$

This proves correctness.

6.2 Security

We may assume $|C| = t - 1$ since an adversary with fewer than $t - 1$ corrupted parties can be simulated by one that corrupts $t - 1$ parties.

The proof has two scheme-specific steps: (i) simulating honest partial signatures, and (ii) sampling corrupted shares. The reduction then follows the RSA-FDH argument of Coron [4].

Simulating honest partial signatures. For $C \subseteq [n]$ with $|C| = t - 1$ and $k \in [n] \setminus C$, define

$$u_{i,C}(k) = D_X L_{i,C_0}(x_k) \quad (i \in C_0)$$

These integers depend only on the public coordinates C and k. For fixed C, abbreviate $u_{i,C}(k)$ by $u_i(k)$ and write $\mathbf{y}_C = (y_c)_{c \in C}$. For an arbitrary integer vector $\mathbf{y}_C$, define the simulator for partial signatures as

$$\mathsf{Sim}_C^{\mathsf{IC}}(k; r, \mathbf{y}_C) = r^{u_0(k)} \prod_{c \in C} r^{e u_c(k) y_c} \bmod N$$

The simulator takes $r \in \mathbb{Z}_N^*$ as an input. The reduction samples r, programs $H(M) \equiv r^{e^{v+1}} \bmod N$, and stores r with the entry for M. To answer a partial signature query on M, the corresponding entry r for M is given as input to the simulator. Lemma 15 shows that for valid corrupted shares, its output equals the honest partial signature on M whenever $H(M) \equiv r^{e^{v+1}} \bmod N$.

Lemma 15. *For every output $(pk, y_1, \ldots, y_n)$ of KGen, every C of size $t - 1$, $k \in [n] \setminus C$, and $r \in \mathbb{Z}_N^*$,*

$$(r^{e^{v+1}})^{R_X y_k} = \mathsf{Sim}_C^{\mathsf{IC}}(k; r, \mathbf{y}_C).$$

Proof. Fix C and $k \notin C$. Interpolation gives $D_X f(x_k) = u_0(k)d + \sum_{c \in C} u_c(k) f(x_c)$ in $\mathbb{Z}$. Reducing modulo $\varphi(N)$ and using $e^{v+1} R_X = e D_X$ gives

$$e^{v+1} R_X y_k \equiv e d u_0(k) + \sum_{c \in C} e y_c u_c(k) \equiv u_0(k) + \sum_{c \in C} e y_c u_c(k) \pmod{\varphi(N)}.$$

Raising r to the power $e^{v+1} R_X y_k$ proves the identity. □

Sampling corrupted shares. In the proof of security, the reduction must give the adversary $\mathcal{A}$ the corrupted shares $(y_c)_{c\in C}$ without knowledge of $\varphi(N)$ or d. We first determine the distribution, and then construct a sampler.

Fix $C \subseteq [n]$ with $|C| = t-1$ and let $V_C = (x_c^j)_{c\in C,\, 1\le j\le t-1}$, the Vandermonde matrix without the constant column. This matrix has nonzero determinant $\det V_C = \pm\prod_{c\in C} x_c \prod_{c<c'}(x_{c'} - x_c)$. The corrupted shares can be represented as $(y_c)_{c\in C} = (d\mathbf{1} + V_C\mathbf{a}) \bmod \varphi(N)$, with $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_{\varphi(N)}^{t-1}$. If V_C was invertible modulo $\varphi(N)$, then $V_C\mathbf{a}$ would be uniform on all of $\mathbb{Z}_{\varphi(N)}^{t-1}$, and the shares would reveal nothing about d, as in Shamir sharing over a field. Invertibility of V_C fails only modulo primes dividing both $\varphi(N)$ and $|\det V_C|$. Let $G_C = \gcd(\varphi(N), |\det V_C|)$ and $e' = e^{-1} \bmod G_C$. This inverse exists because $G_C \mid \varphi(N)$. Let

$$W_C = (e'\mathbf{1} + V_C\mathbb{Z}_{G_C}^{t-1}) \bmod G_C, \quad Y_C = \big\{\mathbf{z} \in [0,\varphi(N))^{t-1} :\ \mathbf{z} \bmod G_C \in W_C\big\}.$$

Lemma 16 (Share distribution). *The corrupted shares $(y_c)_{c\in C}$ are uniform on Y_C.*

The proof is provided in Appendix D.1. Through this lemma we find that the congruence condition defining Y_C depends only on G_C. The sampler needs G_C, which depends on $\varphi(N)$. Unlike $\varphi(N)$, however, G_C divides the public integer $\det V_C$, so the reduction can enumerate the candidates for G_C when the set of candidates and the cost for enumerating are polynomially bounded. The sampler, after guessing G_C still needs to know $\varphi(N)$ to construct Y_C exactly. We fix this by accepting a small statistical distance. Our shares instead lie in the set $[0,B)^{t-1}$ where $B = g\lfloor N/g\rfloor \ge \varphi(N)$. The sampler $\mathrm{Samp}_{C,g}(N)$ draws $\mathbf{b} \leftarrow \mathbb{Z}_g^{t-1}$ and $\mathbf{u} \leftarrow [0, \lfloor N/g\rfloor)^{t-1}$ uniformly and outputs $\big((e^{-1} \bmod g)\mathbf{1} + V_C\mathbf{b}\big) \bmod g + g\mathbf{u}$. It runs in polynomial time and does not use $\varphi(N)$.

Lemma 17. *For every (N,p,q,d) output by $\mathrm{GenRSA}_e(1^\kappa)$,*

$$\mathrm{SD}((y_c)_{c\in C},\ \mathrm{Samp}_{C,G_C}(N)) \le (t-1)\,\frac{N-\varphi(N)}{N} \le (t-1)2^{2-\kappa/2}.$$

The proof is provided in Appendix D.2. The sampler needs G_C, which depends on $\varphi(N)$, but G_C divides the public integer $\det V_C$. Let

$$\mathcal{G}_C = \{g \ge 1\ :\ g \mid \det V_C,\ e \nmid g\},$$

and let $K_X = \max_{|C|=t-1} |\mathcal{G}_C|$. Since $e \nmid \varphi(N)$, $G_C \in \mathcal{G}_C$. The set $\mathcal{G}_C$ depends only on X. [3]

Theorem 18. *Let e be a prime and X be a fixed (t,n)-integer family. If the RSA assumption holds for GenRSA_e, then Π_{IC} is TS-UF-CMA secure in the random oracle model.*

[3] They are computed from the prime factorizations of the integers x_i and $(x_i - x_j)$.

Proof. Let $\mathcal{A}$ be a PPT adversary with advantage $\varepsilon(\kappa)$ that requests partial signatures on at-most $q_S = \text{poly}(\kappa)$ distinct messages. We may assume $|C| = t-1$ and that $\mathcal{A}$ queries the hash oracle for the output forgery message M^* before it outputs (M^*, σ^*). Let Win_i be the event that game i outputs 1.

Game 0 is the standard $\mathsf{Exp}^{\mathsf{ts\text{-}uf\text{-}cma}}_{\Pi_{\mathsf{IC}},\mathcal{A}}(\kappa)$ as outlined in Figure. 1. Thus, $\Pr[\mathsf{Win}_0] = \varepsilon(\kappa)$ which is $\mathcal{A}$'s advantage.

Game 1 the challenger answers query $\mathcal{O}(k, M)$ with $\mathsf{Sim}^{\mathsf{IC}}_C(k; H(M)^{d^{v+1}}, \mathbf{y}_C)$. The challenger knows d and supplies $r = H(M)^{d^{v+1}} \bmod N$ to the simulator. This satisfies the required relation $r^{e^{v+1}} \equiv H(M)^{(ed)^{v+1}} \equiv H(M) \bmod N$. Thus by Lemma 15, this is exactly the real partial signature, so $\Pr[\mathsf{Win}_0] = \Pr[\mathsf{Win}_1]$. The honest shares are no longer used.

Game 2 the challenger draws $(y_c)_{c\in C} \leftarrow \mathsf{Samp}_{C,G_C}(N)$ instead of the real corrupted shares. By Lemma 17 it directly follows that $|\Pr[\mathsf{Win}_2] - \Pr[\mathsf{Win}_1]| \leq \text{SD}((y_c)_{c\in C},\ \text{Samp}_{C,G_C}(N)) \leq (t-1)2^{2-\kappa/2}$. The key is required to obtain G_C and calculate $w_M = H(M)^{d^{v+1}}$. The reduction removes both the uses.

Reduction when G_C is known. We now apply Coron's argument [4]. For given $g \in \mathcal{G}_C$, on input (N, z), $\mathcal{B}_g(N, z, C, \mathsf{st})$ sets $pk = (N, e, X)$. It draws $\mathbf{y}_c \xleftarrow{\$} \mathsf{Samp}_{C,g}(N)$, and runs the second stage $\mathcal{A}^{H,\mathcal{O}}(\mathsf{st}, pk, \mathbf{y}_c)$. On each first query to $H(M)$ by $\mathcal{A}$, it samples $w_M \xleftarrow{\$} \mathbb{Z}^*_N$ and a bit b_M with $\Pr[b_M = 1] = 1 - 1/(q_S+1)$. It returns the value

$$H(M) := \begin{cases} w_M^{e^{v+1}} \bmod N & \text{if } b_M = 1, \\ z w_M^e \bmod N & \text{if } b_M = 0. \end{cases}$$

It answers the partial signing oracle queries $\mathcal{O}(k, M)$ with $\mathsf{Sim}^{\mathsf{IC}}_C(k; w_M, \mathbf{y}_C)$ if $b_M = 1$ and aborts otherwise, since the query cannot be answered due to invalid programming of the hash. On a valid forgery (M^*, σ^*) with $b_{M^*} = 0$, it outputs $\sigma^* w_{M^*}^{-1} \equiv z^{1/e} \bmod N$ which solves the RSA problem. By the property of RSA permutation, both maps $w \mapsto w^{e^{v+1}}$ and $w \mapsto zw^e$ permute $\mathbb{Z}^*_N$, so H is a random function independent of the bit b_M, and when $g = G_C$, $\mathcal{B}_g$ simulates the Game 2 perfectly unless it aborts. As in [4], this happens with probability at least $\Pr[\mathsf{Win}_2]/(\exp(1)\,(q_S + 1))$.

Guessing G_C. We construct $\mathcal{B}(N, z)$ as a reduction separate from $\mathcal{B}_g(N, z, C, \mathsf{st})$. $\mathcal{B}(N, z)$ runs the first stage of $\mathcal{A}$ only once, to obtain (C, st), which is independent of the key. Then, for every $g \in \mathcal{G}_C$, it runs $\mathcal{B}_g(N, z, C, \mathsf{st})$ on the same N and z, and it outputs any y with $y^e \equiv z \bmod N$, that one of these runs is guaranteed to produce, since $G_C \in \mathcal{G}_C$, so

$$\mathsf{Adv}^{\mathsf{rsa}}_{\mathsf{GenRSA}_e,\mathcal{B}}(\kappa) \geq \frac{\Pr[\mathsf{Win}_2]}{\exp(1)\,(q_S+1)} \geq \frac{\varepsilon(\kappa) - (t-1)2^{2-\kappa/2}}{\exp(1)\,(q_S+1)}.$$

Reduction $\mathcal{B}$ makes at most K_X (constant with fixed X) runs, each with $\text{poly}(\kappa, \text{len}(X))$ overhead per oracle query, so it is PPT. If $\varepsilon(\kappa)$ were non-negligible, the advantage of $\mathcal{B}$ would also be non-negligible, contradicting the RSA assumption. □

Theorem 18 is stated for a fixed family X. A simulator without $\varphi(N)$ must simulate a distribution that depends on G_C. Statistically hiding G_C is not possible, since share distributions depending on G_C are statistically far. No single distribution independent of G_C can be statistically close to all of them.

7 Threshold sharing in $\mathbb{Z}$ for arbitrary e and N

The scheme Π_{IC} although proven secure, is limited by the available coordinate families. Pairwise integer coordinate families exist for every (t, n), but the only small ones we have are the exhaustive minima for $t = 2$, $n \leq 9$ and the brute-force results for $t = n - 1$. There is no general technique known to construct such small families. Further, although not a major issue, the scheme still leaks G_C. This motivates the scheme Π_{Z}, which we propose in this section.

Sharing the secret exponent with an integer polynomial at the points $1, \ldots, n$ by clearing the Lagrange denominators with $\Delta = n!$, and drawing the coefficients from a large interval goes back to Frankel, Gemmell, MacKenzie, and Yung [8]. This is a standard technique used in threshold RSA for general moduli [10]. The non-interactive protocol for a general modulus was given by Damgård and Koprowski [10]. These schemes directly share d, and require $e > n$. Shoup's scheme needs both, $e > n$ and safe primes [15]. Constructions that work with every prime e and every n use interaction [13] or several shares per party [5].

The scheme Π_{Z} builds upon [10], and clears all powers of e. The scheme has a trusted dealer, one integer share, and one exponentiation per party. It provides non-interactive signing, and can be used with arbitrary modulus. We prove it TS-UF-CMA secure in the random-oracle model under RSA.

7.1 The scheme Π_{Z}

In this section, the evaluation points are $x_0 = 0$ and $x_i = i$ for $i \in [n]$. Let $\Delta = n!$. For the RSA public parameter e, we write $\Delta = e^{\nu}\Delta'$, where ν is the maximum exponent of e in Δ (i.e. $e^{\nu} \mid \Delta$ but $e^{\nu+1} \nmid \Delta$). Let η be a statistical security parameter, with $2^{-\eta}$ negligible in κ and $A = \mathrm{bitlen}(N) + \mathrm{bitlen}(\Delta) + t + \eta$. Since $e \nmid \Delta'$, there are public integers α, β with $\alpha\Delta'^3 + e\beta = 1$. Given these constants, the scheme Π_{Z} is outlined in Figure 4.

Correctness. Let $x = H(M) \in \mathbb{Z}_N^*$. Lagrange interpolation gives $\sum_{i \in S} \lambda_{i,S} f(i) = f(0) = \Delta s$ in $\mathbb{Q}$. Each $\Delta\lambda_{i,S}$ is in $\mathbb{Z}$, due to clearing factor Δ. so

$$w \equiv \prod_{i \in S} \sigma_i^{\Delta\lambda_{i,S}} \equiv x^{\Delta^2 \sum_i \lambda_{i,S} y_i} \equiv x^{\Delta^3 s} \pmod{N}. \tag{11}$$

From $\Delta = e^{\nu}\Delta'$ and $s \equiv e^{-3\nu} d \bmod \varphi(N)$. Substituting into (11), we get $w \equiv x^{\Delta'^3 d} \bmod N$. Using $ed \equiv 1 \pmod{\varphi(N)}$ and the relation $\alpha\Delta'^3 + e\beta = 1$, we finally obtain

$$\sigma^e \equiv x^{e\alpha\Delta'^3 d + e\beta} \equiv x^{\alpha\Delta'^3 + e\beta} \equiv x \pmod{N}.$$

$\mathsf{KGen}(1^\kappa)$

$(N, p, q, d) \leftarrow \mathsf{GenRSA}_e(1^\kappa);\ \varphi = (p-1)(q-1)$
$s \leftarrow e^{-3\nu} d \bmod \varphi$
$A \leftarrow \text{bitlen}(N) + \text{bitlen}(\Delta) + t + \eta$
$a_1, \ldots, a_{t-1} \xleftarrow{\$} [0, 2^A)$
$f(Z) := \Delta s + \sum_{j=1}^{t-1} a_j Z^j \in \mathbb{Z}[Z]$
for $i \in [n]$ **do** $y_i \leftarrow f(i)$ // over $\mathbb{Z}$
$pk \leftarrow (N, e, n, t)$
return $(pk, y_1, \ldots, y_n)$

$\mathsf{PSign}(pk, i, y_i, M)$

return $\sigma_i \leftarrow H(M)^{\Delta y_i} \bmod N$

$\mathsf{Combine}(pk, M, S, (\sigma_i)_{i \in S})$

$w \leftarrow \prod_{i \in S} \sigma_i^{\Delta \lambda_{i,S}} \bmod N$
return $\sigma \leftarrow w^\alpha H(M)^\beta \bmod N$

$\mathsf{Verify}(pk, M, \sigma)$

return $\sigma^e \equiv H(M) \pmod N$

Fig. 4. The scheme for a prime e and $1 \le t \le n$, with $H : \{0,1\}^* \to \mathbb{Z}_N^*$. The constants $\Delta, \nu, \alpha, \beta$ depend only on (e, n); $\Delta\lambda_{i,S}$ is publicly computable from n, i, S. Negative exponents are computed by inversion in $\mathbb{Z}_N^*$.

7.2 Security

In contrast to schemes dealing shares modulo $\varphi(N)$, integer secret sharing eliminates share leakage. The corrupted shares statistically hide the secret [8,5], enabling a simulation that doesn't require knowledge of $\varphi(N)$. We first state the simulation identity and then show the bound on the statistical distance.

Using the notation and conventions from the public constants in Section 6.1, define $u_{i,C}(k) = \Delta L_{i,C_0}(x_k)$ $(i \in C_0,\ k \in [n] \setminus C)$. For $r \in \mathbb{Z}_N^*$, define

$$\mathsf{Sim}_C^{\mathsf{Z}}(k; r, \mathbf{y}_C) = r^{\Delta' u_0(k)} \prod_{c \in C} r^{e^{2\nu+1} u_c(k) y_c} \bmod N.$$

Lemma 19. *For every output of* KGen*,* C *of size* $t-1$*, and* $w \in \mathbb{Z}_N^*$*,*

$$(w^{e^{2\nu+1}})^{\Delta y_k} = \mathsf{Sim}_C^{\mathsf{Z}}(k; w, \mathbf{y}_C)$$

Proof. Fix C, k. Interpolating f at the nodes indexed by C_0 gives $\Delta y_k = u_0(k)\Delta s + \sum_{c \in C} u_c(k) y_c$ in $\mathbb{Z}$. Since $\Delta s = \Delta' e^\nu s \equiv \Delta' e^{-2\nu} d \pmod{\varphi(N)}$ and $w^{ed} = w$, raising $w^{e^{2\nu+1}}$ to this power gives us the equality.

For an integer s, let $\mathcal{D}_{C,s}$ be the distribution of $(f(c))_{c \in C}$ where $f(Z) = \Delta s + \sum_{j=1}^{t-1} a_j Z^j$ with $a_j \xleftarrow{\$} [0, 2^A)$. The real corrupted shares are distributed with the integer $s_0 = e^{-3\nu} d \bmod \varphi$. To bound the distance we absorb the secret into the random coefficients. We provide a proof of statistical secrecy for our parameters.

Lemma 20. *For every* $C \subseteq [n]$ *with* $|C| = t-1$ *and all* $s, s' \in [0, \varphi(N))$*, we have* $\mathrm{SD}(\mathcal{D}_{C,s}, \mathcal{D}_{C,s'}) < 2^{-\eta-1}$.

Proof. See Appendix E.1.

Theorem 21. *Let e be a prime and $1 \leq t \leq n \leq poly(\kappa)$. If the RSA assumption holds for GenRSA_e, then scheme Π_{Z} is TS-UF-CMA secure in the random oracle model.*

Proof. Let $\mathcal{A}$ be a PPT adversary with advantage $\varepsilon(\kappa)$ that requests partial signatures on at-most $q_S = \text{poly}(\kappa)$ distinct messages. We may assume $|C| = t-1$ and that $\mathcal{A}$ queries the hash oracle for the output forgery message M^* before it outputs (M^*, σ^*). Let Win_i be the event that game i outputs 1.

Game 0 is the standard $\mathsf{Exp}^{\mathsf{ts\text{-}uf\text{-}cma}}_{\Pi_{\mathsf{Z}},\mathcal{A}}(\kappa)$ as outlined in Figure. 1. Thus, $\Pr[\mathsf{Win}_0] = \varepsilon(\kappa)$ which is $\mathcal{A}$'s advantage.

Game 1 answers $\mathcal{O}(k, M)$ with $\mathsf{Sim}^{\mathsf{Z}}_C(k; H(M)^{d^{2\nu+1}}, \mathbf{y}_C)$. By Lemma 19, this is exactly the real partial signature, $\Pr[\mathsf{Win}_0] = \Pr[\mathsf{Win}_1]$. The honest shares are no longer used.

Game 2 the challenger doesn't draw the corrupted shares from the real share distribution polynomial. Instead, the challenger draws $(y_c)_{c\in C} \leftarrow \mathcal{D}_{C,0}$. By Lemma 20 it follows that $|\Pr[\mathsf{Win}_2] - \Pr[\mathsf{Win}_1]| \leq \text{SD}(\mathcal{D}_{C,s}, \mathcal{D}_{C,0}) < 2^{-\eta-1}$. Only the secret d is required for simulation of partial queries.

We now apply Coron's argument [4]. On input (N, z), $\mathcal{B}$ runs Game 2. On each fresh query to $H(M)$ by $\mathcal{A}$, it samples $w_M \xleftarrow{\$} \mathbb{Z}^*_N$ and a bit b_M with $\Pr[b_M = 1] = 1 - 1/(q_S + 1)$. It returns the value

$$H(M) := \begin{cases} w_M^{e^{2\nu+1}} \bmod N & \text{if } b_M = 1, \\ z \cdot w_M^e \bmod N & \text{if } b_M = 0. \end{cases}$$

It answers the partial signing oracle queries $\mathcal{O}(k, M)$ with $\mathsf{Sim}^{\mathsf{Z}}_C(k; w_M, \mathbf{y}_C)$ if $b_M = 1$ and aborts otherwise, since the query cannot be answered due to inadequate programming of the hash. On a valid forgery (M^*, σ^*) with $b_{M^*} = 0$, it outputs $\sigma^* w_{M^*}^{-1} \equiv z^{1/e} \bmod N$ which solves the RSA problem. By the property of RSA permutation, both maps $w \mapsto w^{e^{2\nu+1}}$ and $w \mapsto zw^e$ permute $\mathbb{Z}^*_N$, so H is a random function independent of the bit b_M, and $\mathcal{B}$ simulates the Game 2 perfectly unless it aborts. As in [4], it follows that

$$\mathsf{Adv}^{\mathsf{rsa}}_{\mathsf{GenRSA}_e,\mathcal{B}}(\kappa) \geq \frac{\Pr[\mathsf{Win}_2]}{\exp(1)\,(q_S + 1)} \geq \frac{\varepsilon(\kappa) - 2^{-\eta-1}}{\exp(1)\,(q_S + 1)}.$$

Because $\mathcal{B}$ invokes $\mathcal{A}$ once and incurs only $\text{poly}(\kappa, n)$ computational overhead per oracle query, $\mathcal{B}$ runs in probabilistic polynomial time. Consequently, if $\varepsilon(\kappa)$ were non-negligible, the advantage of $\mathcal{B}$ would also be non-negligible, contradicting the RSA assumption. □

For the integer secret-sharing scheme, the reduction runs the adversary once and incurs only the RSA-FDH loss and the statistical term $2^{-\eta-1}$. As in Section 6, the theorem assumes a trusted dealer, static corruptions, a random oracle, and keys drawn from GenRSA_e.

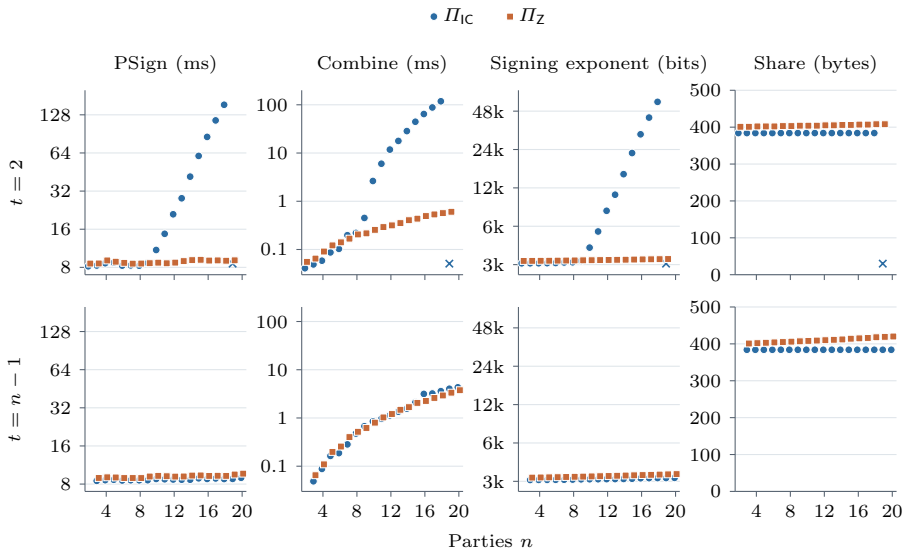

Fig. 5. 3072-bit RSA with $e = 3$; Π_{Z} uses $\eta = 128$. Measured on one core of an Intel Core i7-12650H. Π_{IC} uses pairwise families at $t = 2$ and Table 2's families at $t = n - 1$. Π_{Z} uses natural coordinates. Crosses represent the unavailable Π_{IC} measurements due to timeout. The first three columns use logarithmic scales.

8 Performance evaluation of Π_{IC} and Π_{Z}

Figure 5 compares the two schemes for 3072-bit moduli. Each party computes a single exponentiation in both the schemes, hence the signing time grows with the length of the signing exponent. With the small families we found by exhaustive search and brute-force in Sections 5.2 and 5.3, the performance of the two schemes is comparable. However, for $t = 2$ and $n \geq 10$, the families available for Π_{IC} to use are large. This significantly escalates the length of the clearing factors. For thresholds between 2 and $n - 1$, Π_{IC} only has the universal integer coordinate families, which follow the same drawbacks as in the $t = 2$ case. Thus, the efficiency of scheme Π_{Z} either almost matches or exceeds that of Π_{IC}.

9 Conclusions

We revisited the integer-coordinate approach of Ng and Lau [12] to threshold RSA with a small public exponent. We showed that a pairwise integer family works for every threshold, which settles the universal existence question posed in [12]. We provided an explicit construction algorithm with costs polynomial in n and size of its output. We also showed that universal coordinate families must grow super-polynomially in magnitude, and at least $n \log n$ in bit size.

On the application side, we found that the shares reduced modulo $\varphi(N)$ reveal $\gcd(\varphi(N), \det V_C)$, and our proof for Π_{IC} pays for this with an enumeration

whose cost grows with family and threshold. To avoid this problem, we proposed an integer shares scheme Π_{Z}, a variant of the known integer secret sharing [5,8]. This supports every prime public exponent e for every (t, n) with a single exponentiation per party, and a reduction that runs the adversary only once. It also doesn't depend on finding good coordinate families.

9.1 Further open problems

This work answers many of the questions asked by Ng and Lau [12]. However, our work also poses newer questions concerning the coordinate structure, and the security of the signature schemes.

Coordinate growth and construction. We provide the lower bound $\log s(n) \geq \frac{1}{2} n \log n - O(n)$ for the maximum coordinate size of pairwise integer coordinate families. A natural target is an upper bound, either matching the lower bound, or a determination of the correct asymptotic order. Other natural avenues are to provide asymptotic analysis of threshold-specific families. The objective in that setting is $g(t, n)$, whose growth may differ substantially from pairwise case, as is noticed in Section 5.3 for $(n-1, n)$ families.

Reduction for Π_{IC} without candidate enumeration. The modular-share proof must reproduce the congruence constraints imposed by $G_C = \gcd(\varphi(N), |\det V_C|)$. Enumerating its public candidates costs up to K_X adversary invocations. Whether a reduction that simulates the shares modulo $\varphi(N)$ without guessing G_C exists, or if the leaked G_C is harmless is unknown.

Message encodings and end-to-end evaluation. The proofs use a random oracle into $\mathbb{Z}_N^*$. A standardized signature interface requires a separate specification and security analysis.

Acknowledgements

The second author would like to thank Prabhat Kumar Yadav (220774), and Kuldeep Thakare (220557) for introducing the open problems in [12] to him. They studied the paper [12] as a course project in CS641: Modern Cryptography, taught by the second author at IIT Kanpur during Jan-April 2026 semester.

References

1. Abspoel, M., Cramer, R., Damgård, I., Escudero, D., Yuan, C.: Efficient information-theoretic secure multiparty computation over $\mathbb{Z}/p^k\mathbb{Z}$ via galois rings. In: Theory of Cryptography - 17th International Conference, TCC 2019, Proceedings, Part I. LNCS, vol. 11891, pp. 471–501. Springer (2019)
2. Apostol, T.M.: Introduction to Analytic Number Theory. Undergraduate Texts in Mathematics, Springer, New York, NY (1976). https://doi.org/10.1007/978-1-4757-5579-4

3. Bellare, M., Rogaway, P.: The Exact Security of Digital Signatures-How to Sign with RSA and Rabin. In: Maurer, U. (ed.) Advances in Cryptology - EUROCRYPT '96. pp. 399–416. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)
4. Coron, J.S.: On the Exact Security of Full Domain Hash. In: Advances in Cryptology – CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer (2000)
5. Damgård, I., Thorbek, R.: Linear Integer Secret Sharing and Distributed Exponentiation. In: Public Key Cryptography - PKC 2006. LNCS, vol. 3958, pp. 75–90. Springer (2006), full version: Cryptology ePrint Archive, Report 2006/044
6. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology. p. 307p–315. CRYPTO '89, Springer-Verlag, Berlin, Heidelberg (1989)
7. Escudero, D., Soria-Vazquez, E.: Efficient information-theoretic multi-party computation over non-commutative rings. In: Annual International Cryptology Conference on Advances in Cryptology CRYPTO'21. pp. 335–364. Springer (2021)
8. Frankel, Y., Gemmell, P., MacKenzie, P.D., Yung, M.: Optimal-resilience proactive public-key cryptosystems. In: 38th Annual Symposium on Foundations of Computer Science (FOCS '97). pp. 384–393. IEEE Computer Society (1997). https://doi.org/10.1109/SFCS.1997.646127
9. Intel Corporation: Intel software guard extensions programming reference (2016)
10. Ivan Damgård, I., Koprowski, M.: Practical Threshold RSA Signatures without a Trusted Dealer. In: Advances in Cryptology – EUROCRYPT 2001. EUROCRYPT '01, LNCS, vol. 2045, pp. 152—-165. Springer-Verlag, Berlin, Heidelberg (2001)
11. Jones, G.A., Jones, J.M.: Elementary Number Theory. Springer Undergraduate Mathematics Series, Springer (1998)
12. Ng, S., Lau, J.: Efficient $e = 3$ Threshold RSA via Integer Coordinates for Intel SGX. In: Topics in Cryptology – CT-RSAC 2026: Cryptographers' Track at the RSAC 2026 Conference, San Francisco, CA, USA, March 23–26, 2026, Proceedings. pp. 262—287. Springer-Verlag, Berlin, Heidelberg (2026)
13. Rabin, T.: A simplified approach to threshold and proactive rsa. In: Krawczyk, H. (ed.) Advances in Cryptology — CRYPTO '98. pp. 89–104. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)
14. Shamir, A.: How to share a secret. Communications of the ACM **22**(11), 612—-613 (Nov 1979). https://doi.org/10.1145/359168.359176
15. Shoup, V.: Practical Threshold Signatures. In: Preneel, B. (ed.) Advances in Cryptology — EUROCRYPT 2000. pp. 207–220. Springer Berlin Heidelberg, Berlin, Heidelberg (2000)

A Auxiliary Proofs

A.1 LCM Translation

Proof. The set A_2 is trivially pairwise. Assume that for some $m \geq 2$, A_m is pairwise. Since all the elements of A_m are strictly positive, $l_m < l_m + a$ for every $a \in A_m$. Consider the translated elements $\{l_m + a : a \in A_m\}$. Note that l_m is smaller than all the elements in this translated set, and thus is not present in it. Hence, $|A_{m+1}| = m + 1$ and all the elements remain positive.

Now, there are 2 types of pairs in the set A_{m+1}: (l_m, l_m+a) and (l_m+a, l_m+b), where $a \neq b$. The difference of the pair $(l_m, l_m + a)$ for $a \in A_m$ is a that divides l_m and $a \mid (l_m + a)$. The difference of the pair $(l_m + a, l_m + b)$ is $a - b$. Since A_m is a pairwise family, $(a - b) \mid a$ and $(a - b) \mid b$. In addition, $b \mid l_m$, it follows that $a - b \mid l_m$, and hence $a - b$ divides both translated coordinates. Hence A_{m+1} is a pairwise integer family of size $m + 1$. By induction, A_m is a pairwise integer family for all $m \geq 2$.

B Implementation and Cost of `construct`

B.1 Generating candidates

Before the fallback of Theorem 11(ii), we test at most T smaller offsets. Offsets must satisfy (7). Consider prime power p^{a_p} in the factorization of M. Since $p^{a_p} \mid M$, $\gcd(p^{a_p}, |q - d|) \mid \gcd(M, |q - d|)$. Thus, offset q satisfying (7) also satisfies the weaker condition $\gcd(p^{a_p}, |q-d|) \mid r+q$. It suffices to check residues $\bmod(p^{a_p})$, since if q satisfies the given condition, $q + kp^{a_p}$ will also satisfy it for $k \in \mathbb{Z}$. We check all the residues $\bmod p^{a_p}$, and keep those residues which satisfy the condition for all $d \in D$.

We choose a set of primes $\mathcal{P}$, and an upper bound U, and consider powers of these small primes, which keep $p^{a_p} \leq U$. Thus, residues modulo p^{a_p} which satisfy our condition are listed, and these congruences are then combined using CRT. This is valid since the prime powers are coprime. This finally gives us the residues $w \bmod W$, where W is the product of these prime powers. The candidates are then $q = kW + w$, which are then checked against the condition in (7). At most T such candidates are checked, and if none of those satisfy the condition, the fallback is used. Our search uses $\mathcal{P} = \{2, 3, 5, 7, 11\}$, $U = 10^6$, and $T = 10^6$.

B.2 Proof of Proposition 12

Proof. For $m = s, \ldots, n$, let D_m denote the offset set after $m - s$ extensions, and let (r_m, M_m) be the representative of set of congruences $\Gamma(D_m)$. Each $d \in D_n$ satisfies $d = (z + d) - z < z + d \leq \max X < 2^b$. Consequently every difference $|d - e|$ of distinct $d, e \in D_n$ is a positive integer below 2^b. There are $\binom{n}{2}$ such differences.

Since $D_m \subseteq D_n$, the differences occurring in D_m form a subset of the differences of D_n, so M_m divides $\operatorname{lcm}\{|d-e| : d \neq e \in D_n\}$, hence

$$M_m \leq \prod_{\substack{d,e \in D_n \\ d<e}} |d-e| < \left(2^b\right)^{\binom{n}{2}} = 2^{b\binom{n}{2}}.$$

Also $0 \leq r_m < M_m$.

While extending D_m, $q_* = M_m - r_m$ (or $2M_m - r_m$) is compatible by Theorem 11(ii), and thus occurs in the candidate sequence $q = kW + w$. Candidates are examined in increasing order, so the stage either accepts a candidate $q \leq q_*$ or stops before q_*. Hence every seen candidate $q \leq q_* \leq 2M_m$, and every new difference $|q-e|$ with $e \in D_m$ is at most $\max(q, e) \leq 2M_m$, since $0 \in D_m$ gives $e \mid M_m$ and hence $e \leq M_m$. Thus, $q \leq 2M_m < 2^{1+b\binom{n}{2}} = 2^\beta$. All operands are therefore below 2^β.

At each stage, at most T candidates and one fallback, are tested against at most m old offsets. The accepted offset then triggers exactly m merges. Summed over $m = s, \ldots, n-1$, this is $\binom{n}{2} - \binom{s}{2}$ merges and at most $(T+1)(\binom{n}{2} - \binom{s}{2})$ individual tests of (7). One merge is one extended gcd and $O(1)$ multiplications and divisions; one test is one gcd and one remainder. On β-bit operands each costs $O(\beta^2)$; dropping constant factor of s since seed is fixed gives $O\big((T+1)n^2\beta^2\big) = O(n^6 b^2)$ for constant T.

At each of the $n - s$ stages, for $p \in \mathcal{P}$, value $\mathrm{v_p}(|d-e|)$ is computed for all $O(n^2)$ pairs of distinct offsets $d, e \in D_m$. One evaluation takes at most β divisions by p, each $O(\beta)$, so this costs $O(n^2\beta^2)$ per stage and $O(n^3\beta^2) = O(n^7 b^2)$ overall, dominating the total. The algorithm stores D_m (at most n integers below 2^b), the pair (r_m, M_m), the current candidate, and the current difference (each $O(\beta) = O(n^2 b)$ bits), and residue lists whose length and entry size are capped by the fixed parameters, independently of n. The total is $O(n^2 b)$ bits.

C Exhaustive search for minimum pairwise families

This appendix describes the exhaustive search used to obtain the minimum pairwise integer families in Table 1.

Lemma 22 (Divisor representation). *Let X be a positive pairwise integer coordinate family and $a = \min X$. Then there are $n-1$ distinct positive divisors $d_1, \ldots, d_{n-1}$ of a with $X = \{a, a+d_1, \ldots, a+d_{n-1}\}$. In particular* $\max X \leq 2a$.

Proof. For $x \in X \setminus \{a\}$, put $d = x - a > 0$. The pairwise condition gives $d \mid a$, and distinct x give distinct d. Every positive divisor of a is at most a, so $x = a + d \leq 2a$.

Let B be the bound on the maximum element of our pairwise family. We search for a family with a bound B. If the search is successful, we decrease the bound and search again. If a family doesn't exist for a lower bound, we have arrived at the minima. Let a be the smallest element in our family, $a < B$.

Lemma 22 says that a positive pairwise family is $\{a\} \cup \{a + d : d \in D\}$, where D consists of $n - 1$ positive divisors of a, each at most $B - a$. For $d < d'$ in D, the pair $a + d$, $a + d'$ is pairwise exactly when $(d' - d) \mid (a + d)$. We form a graph, with these divisors as the nodes. An edge exists between $d < d'$ when $(d' - d) \mid (a + d)$. Then a together with a clique of size $n - 1$ forms a pairwise family. We search for such a clique by a standard clique-finding procedure.

D Auxiliary Proofs for Π_{IC}

This appendix provides proofs for the algebraic properties used in Section 6.

D.1 Proof of Lemma 16

In this proof, we index the rows of the matrix by $c \in C \subseteq [n]$.

Proof. The shares are $y = (d\mathbf{1} + V_C\mathbf{a}) \mod \varphi(N)$ where $\mathbf{a} \xleftarrow{\$} \mathbb{Z}_{\varphi}^{t-1}$. Since $\mathbf{a} \mapsto V_C\mathbf{a} \bmod (\varphi(N))$ is a group homomorphism of $\mathbb{Z}_{\varphi(N)}^{t-1}$, so $V_C\mathbf{a} \bmod \varphi(N)$ is uniform on its image I, and y is uniform on $d\mathbf{1} + I$.

For any vector $\mathbf{z} \in \mathbb{Z}_{\varphi(N)}^{t-1}$, $\mathbf{z} \equiv V_C\mathbf{a} \bmod \varphi(N)$ for some $\mathbf{a} \in \mathbb{Z}_{\varphi(N)}^{t-1}$ iff $\mathbf{z} \equiv V_C\mathbf{b} \bmod G_C$ for some $\mathbf{b} \in \mathbb{Z}_{G_C}^{t-1}$. The forward direction holds since $G_C \mid \varphi(N)$, with $\mathbf{b} = \mathbf{a} \bmod G_C$. Conversely, if $\mathbf{z} \equiv V_C\mathbf{b} \pmod{G_C}$, then $\mathbf{z} = V_C\mathbf{b} + G_C\mathbf{w}$ for some $\mathbf{w} \in \mathbb{Z}^{t-1}$. Write $G_C = s \det V_C + s'\varphi(N)$ by Bézout's identity, with $s, s' \in \mathbb{Z}$, then, $G_C\mathbf{w} \equiv s \det(V_C)\mathbf{w} \bmod \varphi(N)$. Use $\det(V_C)\mathbf{w} = V_C \mathrm{adj}(V_C)\mathbf{w}$. Then, $\mathbf{z} \equiv V_C\mathbf{a} \pmod{\varphi(N)}$ with $\mathbf{a} = (\mathbf{b} + s\mathrm{adj}(V_C)\mathbf{w}) \bmod \varphi(N)$.

Finally, since $ed \equiv 1 \pmod{\varphi(N)}$ and $G_C \mid \varphi(N)$, we have $d \equiv e' \pmod{G_C}$. Applying the above to $\mathbf{z} - d\mathbf{1}$, $\mathbf{z} \in d\mathbf{1} + V_C\mathbf{a} \bmod \varphi(N) \iff \mathbf{z} - e'\mathbf{1} \equiv V_C\mathbf{b} \pmod{G_C}$ for some $\mathbf{b} \iff \mathbf{z} \bmod G_C \in W_C$, i.e. the set on which y is uniform is exactly Y_C. □

D.2 Proof of Lemma 17

Proof. Let $G = G_C$ and $B = G\lfloor N/G \rfloor$. Every $\mathbf{z} \in [0, B)^{t-1}$ can be written uniquely as $\mathbf{z} = \boldsymbol{\rho} + G\mathbf{u}$ with $\boldsymbol{\rho} = \mathbf{z} \bmod G$ and $\mathbf{u} \in [0, \lfloor N/G \rfloor)^{t-1}$. In the sampler, $\boldsymbol{\rho}$ is uniform on W_C (by the homomorphism argument modulo G) and $\mathbf{u}$ is uniform, so its output is uniform on $\{\mathbf{z} \in [0, B)^{t-1} : \mathbf{z} \bmod G \in W_C\}$. Since $G \mid \varphi(N) < N$, we have $\varphi(N) \leq B$, so Y_C is the part of this set inside $[0, \varphi(N))^{t-1}$. The distance between the uniform distributions on a set and on a subset is the probability that the sample falls outside the subset. Coordinate c falls outside $[0, \varphi(N))$ exactly when $u_c \geq \varphi(N)/G$. The probability of this event is $1 - \varphi(N)/B \leq (N - \varphi(N))/N$. Also, $N - \varphi(N) = p + q - 1$ and $p, q \geq 2^{\kappa/2-1}$, $\frac{N-\varphi(N)}{N} < \frac{1}{p} + \frac{1}{q} \leq 2^{2-\kappa/2}$. A union bound over the $t - 1$ coordinates then completes the proof. □

E Auxiliary Proofs for Π_{Z}

E.1 Proof of Lemma 20

Proof. Let $\Delta\big(1 - \prod_{c\in C}(1 - Z/c)\big) = \sum_{j=1}^{t-1} b_j Z^j$. This polynomial equals Δ at every $c \in C$, so the shares of C are $\big(\sum_j (a_j + s b_j) c^j\big)_{c\in C}$: they depend on $(s, \mathbf{a})$ only through $\mathbf{a} + s\mathbf{b}$. Since $\prod_{c\in C} c$ divides $n! = \Delta$, each b_j is an integer with $|b_j| \le \Delta\binom{t-1}{j}$. Hence the distance is at most the distance between the uniform distributions on $[0, 2^A)^{t-1}$ and on its translate by $(s' - s)\mathbf{b}$. By a union bound over the coordinates, this is at most

$$\frac{|s' - s| \sum_j |b_j|}{2^A} < \frac{2^{\mathrm{bitlen}(N)} \cdot 2^{\mathrm{bitlen}(\Delta)+t-1}}{2^A} = 2^{-\eta-1}.$$

□