ACE Working Group
Internet-Draft
Intended status: Proposed Standard
Expires: June 2026

P.J. Nsiangani Ed.

draft-nsiangani-phi-ns-00

01 December 2025

Phi-Ns Cryptosystem

Status of this Memo

Copyright Notice

Abstract

This document describes Phi-Ns, a new asymmetric cryptographic primitive based on the structured decomposition of the quadratic gap between two primes. Phi-Ns defines a public key q and a private key p such that the difference q^2 - p^2 is expressed as a composite value T. The value T is then factored into small controlled components noted a, b, and R, and the final secret structure abR is encoded through a randomized serialization process.

The security of Phi-Ns does not rely on integer factorization, the discrete logarithm problem, or elliptic curves. Instead, it depends on the difficulty of recovering p from q when the attacker has no oracle, no structural anchor, and no method to determine whether a candidate p is correct. Because T can be recursively decomposed and reassembled into multiple unpredictable forms, the resulting search space grows combinatorially.

Phi-Ns further supports recursive expansion of the internal abR structure, allowing the construction of high-entropy secret states even

when p and q have relatively small bit lengths. This enables compact

public keys, efficient computation, and strong post-quantum resistance based on a non-factorization hardness assumption.

This document specifies the algorithmic model of Phi-Ns, its parameter choices, and the serialization rules used to encode the abR structure. It also outlines two public-key variants, PK-p and PK-q, which expose either p or q as the public value depending on deployment constraints.

Status of This Memo

This document is an Internet-Draft and is submitted in full conformance with the provisions of BCP 78 and BCP 79. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). They represent ongoing work and are not standards. This document is intended for publication on the Standards Track.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or made obsolete by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as work in progress.

This Internet-Draft will expire six months from the date of publication unless replaced by a more recent version.

The IETF invites comments and discussion about this document. Please review the rights, licenses, and restrictions relating to this document as described in the IETF Trust Legal Provisions: https://trustee.ietf.org/license-info

Copyright Notice

1. Introduction

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] and RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

Phi-Ns is an asymmetric cryptographic primitive based on the structured decomposition of the quadratic gap between two prime numbers. Instead of

deriving its security from integer factorization, discrete logarithms, or

elliptic curves, Phi-Ns defines a new hardness assumption: recovering the secret prime p, or the internal structure abR, from limited public information when no oracle exists and no method can confirm whether a candidate p is correct.

Given two primes p and q, Phi-Ns defines the fundamental relation:

$$q^2 - p^2 = T$$

where T is a composite value. The value T is then decomposed into the structured form abR, where a and b correspond to controlled exponents of 2 and 3 extracted from T, and R is the remaining composite factor. After this extraction step, the tuple (a, b, R) is serialized through a randomized encoding process that removes any predictable structure and achieves a high degree of entropy. Because R can contain multiple independent factors and the serialization order is randomized, the number of possible equivalent representations is extremely large.

A critical security property of Phi-Ns is that an attacker cannot confirm whether a guessed value p is correct or not. No oracle exists, and no structural anchor is leaked. A failure or a success is indistinguishable for all incorrect values. As a result, recovering the correct p requires exploring a search space with no direction, no gradient, and no method to prune incorrect candidates. The attacker must guess p, then guess the full abR structure, and finally reconstruct the exact serialized form expected by the system. This produces a combinatorial explosion of possible candidates, even when p and q have small bit sizes.

Phi-Ns supports two deployment modes:

* PK-q: the public key is q and the private key contains p and abR. In this mode the attacker must recover p from q without knowing the scale of the gap q^2 - p^2, and without knowing whether p lies near or far from q. Because the distribution of prime gaps is irregular and unbounded, no attacker can determine the size of the interval in which p may reside.

* PK-p: the public key is p and the private key contains q and abR. This mode hides q completely. Since the attacker cannot determine the size or direction of the quadratic difference q^2 - p^2, the search becomes harder because the attacker does not even know whether q is slightly larger than p or extremely distant from p. The entropy of the problem grows rapidly when q is chosen far from p, and no leakage occurs because the attacker does not know the magnitude of T.

The structure of T is not fixed. Phi-Ns allows recursive expansion: the secret prime p itself may be expressed as:

p^2 = p2^2 + a2 b2 R2

and this process may be repeated multiple times. Each recursion creates a

new decomposition, which is also serialized in a randomized way. This creates a layered secret state with exponentially increasing entropy. In practice this means that even a 32-bit public parameter can lead to a secure system if the recursion depth is sufficient. Each decomposition generates new abR structures, each with its own randomized ordering, and each requiring full reconstruction by an attacker.

Because no element of abR, and no ordering of abR, is ever exposed, and because the attacker cannot distinguish invalid candidates from the correct one, the Phi-Ns search problem has no known reduction to RSA, DLP, ECC, multivariate systems, or lattice-based constructions. The security is derived from the impossibility of validating partial guesses and from the combinatorial explosion of decomposed structures.

This document defines the Phi-Ns primitive, the public key modes PK-p and
PK-q, the conditions for decomposition of T into abR, the requirements for randomized serialization, and the general architecture needed for interoperable implementations. The goal of this specification is to provide the necessary foundation to support future PKI frameworks, session protocols, and perpetual key systems that rely on the unique properties of Phi-Ns.

2. Terminology

This section defines terminology used throughout this document.
All terms are normative for the Phi-Ns cryptographic primitive.

Prime:
: A positive integer greater than 1 whose only divisors are 1 and itself.

p:
: The private prime in the PK-q mode. In the PK-p mode p is the public value. The value p participates in the equation q^2 - p^2 = T.

q:
: The public prime in the PK-q mode. In the PK-p mode q is private. The value q participates in the equation q^2 - p^2 = T.

T:
: The quadratic gap defined by T = q^2 - p^2. The value T must be strictly positive and composite.

a:
: The maximal exponent of 2 extracted from T. This value is private.

b:
: The maximal exponent of 3 extracted from T. This value is private.

R:

The remaining composite cofactor of T after removing 2^a and 3^b. The value R is private and may contain one or several distinct composite or prime factors.

abR:
The structured decomposition of T into the tuple (a, b, R). This tuple represents the complete internal secret state derived from T.

Serialized abR:
A randomized encoding of the tuple (a, b, R) obtained through a permutation and ordered concatenation process. The serialization is private and must be reproduced exactly during verification.

Recursive decomposition:
A process where p itself is replaced by a new p2 such that p^2 = p2^2 + a2 b2 R2. The structure abR may therefore contain nested decompositions.

PK-q:
A Phi-Ns mode where q is the public key and p, a, b, R are private. The attacker must recover p from q without knowing the magnitude of T or the internal decomposition.

PK-p:
A Phi-Ns mode where p is the public key and q, a, b, R are private. The attacker must recover q without knowing the magnitude or even the direction of the quadratic gap.

No-oracle condition:
The attacker receives no feedback when testing a candidate p or q. No test can confirm whether a guessed value is correct. All invalid candidates appear identical to the correct one.

Serialization space:
The total number of distinct encodings of the internal structure. It includes all permutations of factors inside R, all permutations of abR, and all choices of recursive decomposition when enabled.

Entropy amplification:
The increase in difficulty obtained by applying recursive decomposition or by enlarging the number of permutations possible during serialization.

Phi-Ns hardness assumption:
The assumption that recovering p or q from a public parameter, together with reconstructing the exact serialized internal structure abR, is computationally infeasible both classically and quantumly.

Implementer:
Any party building, integrating, or deploying Phi-Ns in software, hardware, or protocol environments.

Verifier:
Any party that validates information derived from a Phi-Ns key, including correctness of signatures or encrypted structures.

3. Algorithm Overview

Phi-Ns is an asymmetric cryptographic construction based on the structured decomposition of the quadratic gap between two primes. Given two primes p and q with q > p, the core identity is:

$$q^2 - p^2 = T$$

The value T is strictly positive and composite. Phi-Ns expands T into a structured internal secret called abR, defined as:

$$T = 2^a * 3^b * R$$

where:
- a is the maximal power of 2 dividing T.
- b is the maximal power of 3 dividing T.
- R is the remaining composite cofactor, which may include any number of prime or composite factors.

None of the values a, b, or R are public. Only one of the two primes (p or q) is exposed depending on the selected variant. The internal structure abR is then transformed through a randomized serialization process that produces a high-entropy private state.

Phi-Ns defines two public-key variants:

PK-q Mode:
The public key is q. The private key contains p and the serialized form of abR. The attacker must recover p from q without oracle feedback and without knowing the size or factorization of T. Because every incorrect p yields a valid-looking T, there is no way for the attacker to detect success.

PK-p Mode:
The public key is p. The private key contains q and the serialized form of abR. The attacker must recover q from p without knowing the direction or the magnitude of the gap q^2 - p^2. This removes the anchoring property associated with PK-q and increases search ambiguity.

In both modes the attacker must also reconstruct the correct abR decomposition. Even if p were guessed, reconstruction of T would require:
- determining the correct a and b,
- factoring the remaining R (fully or partially),
- reproducing the exact randomized serialization order applied during key generation.

Phi-Ns supports an optional recursive mechanism. A private key holder may decompose p itself into:

$$p^2 = p2^2 + 2^(a2) * 3^(b2) * R2$$

and repeat the process to obtain arbitrarily deep nested structures. Each recursion multiplies the number of possible secret states,

producing exponential entropy amplification even when p and q have modest bit lengths.

Serialization of abR incorporates random permutation of factors, random placement of a and b, and randomized block grouping. This yields a very large space of encodings. The verifier cannot derive this encoding from public data and must rely on the private key.

In all cases Phi-Ns assumes a no-oracle environment: the attacker obtains no indication whether a candidate p or q is correct, and no

test exists to validate a reconstruction. As a result the search space contains no gradients, no distinguishing features, and no structural anchors.

Phi-Ns is not based on integer factorization, discrete logarithms, elliptic curves, lattices, or multivariate equations. The approach introduces a new hardness class based on ambiguity of quadratic inversion combined with non-unique structured decomposition.

4. Detailed Specification

This section defines the Phi-Ns asymmetric primitive using only the mathematical elements that belong to the specification. No additional profiles or assumptions are introduced. All size choices for p, q, and R are controlled by the implementer.

4.1. Core Equation

Phi-Ns is defined by two primes p and q such that:

$$T = q^2 - p^2$$

T MUST be strictly positive and MUST be composite. The implementer selects the bit lengths of p and q based on the security target.

4.2. Extraction of the abR Structure

The quadratic gap T is decomposed into the triple (a, b, R) through the following deterministic steps:

Step 1: Compute $T = q^2 - p^2$
Step 2: Extract a such that 2^a divides T but $2^{(a+1)}$ does not
Step 3: Extract b such that 3^b divides $(T / 2^a)$ but $3^{(b+1)}$ does not
Step 4: Set $R = T / (2^a * 3^b)$

R MUST NOT be divisible by 2 or 3. R SHOULD be large enough to preserve entropy but the exact size is left to the implementer.

4.3. Randomized Serialization of abR

To conceal structure, the triple (a, b, R) is transformed into a serialized encoding called the abR-encoding. The process is:

- Perform partial factorization of R using small trial division.

- The resulting values are called atoms. The remaining cofactor, if any, is also considered an atom.
- A seed and a salt are created for this key generation instance.
- A pseudorandom permutation derived from the seed and the salt permutes all atoms.
- Atoms are grouped into blocks. The number of blocks is derived from a pseudorandom function.
- The final abR-encoding consists of:
    a, b, salt, number_of_blocks, and the permuted list of blocks.

For a given seed the encoding MUST be deterministic. The encoding MUST hide the internal order and structure of the factors of R.

4.4. Optional Recursive Decomposition

Phi-Ns MAY decompose p itself using the same relation. The recursion is:

    p^2 = p_prime^2 + T_prime

where T_prime is decomposed into a_prime, b_prime, and R_prime. The process MAY repeat any number of times.

Recursive decomposition increases entropy because each abR triple in the chain adds uncertainty. The public key does not reveal whether recursion was used.

4.5. Public Key Exposure Modes (PK-p and PK-q)

Phi-Ns defines two exposure modes:

PK-p mode:
    Public key: p
    Private components: q and the abR-encoding
    The attacker sees p but has no anchor toward q. Since no oracle is available, the attacker cannot know whether a guessed p,q pair is correct. The search space has no stopping rule.

PK-q mode:
    Public key: q
    Private components: p and the abR-encoding
    The attacker sees q and must guess p. For each candidate p, the attacker must test whether q^2 - p^2 yields a T that could match an abR encoding. Because the abR structure is randomized and because no oracle confirms correctness, there is no way to know when the correct p has been found.

4.6. Perpetual PKI Property

Phi-Ns supports perpetual key continuity. The public key (either p or q) MAY remain constant while the private key is refreshed by re-randomizing the abR decomposition using new seed and salt values. No reissuance of certificates is required unless local policy demands it. This property is derived directly from the fact that the abR encoding carries the entropy of the secret state, not the size of p or q.

4.7. Minimum and Maximum Sizes

This specification does not enforce a minimum size for p, q, or R. Implementers are expected to select sizes in accordance with the target security level. Because entropy is carried primarily by the abR encoding, even small p and q can be made secure through recursion and through expansion of the abR structure.

The specification does not restrict the maximum size of any value.

4.8. Parameter Profiles

This specification defines three non normative parameter profiles. They exist to provide guidance to implementers while preserving full control over p, q, and R. These profiles are examples only and do not restrict

the algorithm.

Profile S:
- Intended use: embedded systems, IoT, constrained devices
- Typical p size: 64 to 96 bits
- Typical q size: 64 to 96 bits
- Expected R size after decomposition: 150 to 300 bits
- Recursion: optional, typically 1 or 2 levels
- Notes: public key remains small while entropy is carried by the abR encoding and optional recursion.

Profile M:
- Intended use: general purpose cryptography
- Typical p size: 128 to 192 bits
- Typical q size: 128 to 192 bits
- Expected R size after decomposition: 300 to 600 bits
- Recursion: optional, typically 1 to 3 levels
- Notes: recommended for most deployments that need long term security with compact public keys.

Profile L:
- Intended use: high security, post quantum hardening
- Typical p size: 256 bits or greater
- Typical q size: 256 bits or greater
- Expected R size after decomposition: 512 bits or greater depending on the factorization depth and the randomized serialization structure.
- Recursion: optional without upper limit
- Notes: for environments requiring maximum entropy. The abR serialization and recursive decomposition provide exponential growth of the secret search space.

Profile selection does not change the algorithm. All profiles rely on the same decomposition mechanism, the same abR serialization method, and the same security model based on the absence of oracle feedback. Implementers MAY define additional profiles as long as the core mechanism remains unchanged.

6. Security Considerations

This section describes the security foundations of Phi-Ns, the assumed attack model, and the cryptanalytic implications of exposing either p or

q as the public key. Security relies on the difficulty of inverting the structured relation:

```
q*q - p*p = 2^a * 3^b * R = T
```

where T is never published, and the internal structure (a, b, R, permutation, partition) is committed through SHA-256 without leakage. The adversary receives only the public key and the commitment.

6.1. Hardness Assumption

The adversary must recover the full private tuple:

```
(p or q), a, b, R, factorization of R,
permutation of atomic factors,
partition of blocks, and serialized encoding.
```

The inversion problem is:

```
Given PK and commit,
find all private values such that ENC(abR) matches commit.
```

This problem is non-linear, combinatorial and recursive. It does not reduce to integer factorization, discrete logarithm or lattice problems. Shor’s algorithm is ineffective because no usable composite is exposed. Grover’s algorithm yields only quadratic speedup over an already super-exponential domain.

6.2. No Oracle

Phi-Ns gives no feedback when a candidate reconstruction is incorrect. The mapping:

```
commit = SHA256(ENC(abR))
```

behaves as a random oracle. Every incorrect guess produces an unrelated digest. The attacker cannot detect proximity to the correct structure and
cannot prune the search space. There is no partial verification.

6.3. Growth of the Search Space

To invert Phi-Ns, the adversary must guess simultaneously:

- the correct prime (p in PKQ, q in PKP),
- the correct gap T = q*q - p*p,
- the correct exponents a and b,
- the correct residual R,
- the full atomic factorization of R,
- the permutation of atomic factors,
- the number of blocks K,
- the assignment of atoms to blocks,
- the final serialized bytes of ENC(abR).

Each dimension multiplies the search space. Even modest parameters produce super-exponential complexity. For example, a 128-bit R with 20

atomic factors yields more than:

    20! * Bell(20) > 2^200

possible blockings and permutations, excluding the search over p or q.
For realistic security levels, the total inversion space exceeds 2^300.

6.4. Recursive Decomposition

Phi-Ns optionally allows recursive decomposition:

    p*p = p2*p2 + 2^a2 * 3^b2 * R2

and p2 may itself be decomposed again. Each layer introduces new
factors,
new permutations and new partitions. The complexity multiplier per layer
is approximately:

    layer_cost = (factor_count)! * Bell(factor_count)

so two layers with 20 factors each yield an inversion space above
2^400. This allows very small public keys (for example, 32 or 64 bits)
to
reach effective security comparable to 128-bit or 256-bit schemes when
recursion is used.

6.5. PKQ vs PKP Exposure

PKQ exposes q. The attacker must find p near q because primes of similar
size have predictable spacing. The search range for p is therefore
bounded.

PKP exposes p. The attacker must guess q without knowing its magnitude.
Since q is derived from a 256-bit PRF, the attacker cannot estimate the
gap:

    delta = q - p

The domain for candidate q is effectively unbounded above p. This makes
PKP strictly harder to invert. PKP enables extremely small public keys
for IoT and embedded deployment.

Numeric illustration (PKP):

    p = 15485863
    q is unknown and could be any prime > p
    For q in [p+1, p+2^128], the adversary must test all primes
    before checking decomposition. This search space is > 2^126.

6.6. Quantum Resistance

Shor’s algorithm does not apply because there is no composite number to
factor. Grover’s algorithm improves brute-force by a square root, but
the
domain is super-exponential in R and the block structure. For any
deployment profile of at least 128-bit classical security, the quantum-

reduced cost remains far outside practical reach.

6.7. Resistance to Meet-in-the-Middle

Meet-in-the-middle attacks require partial leakage on two halves of a structure. Phi-Ns reveals none. All private values interact through ENC(abR), which destroys correlation via permutation and partitioning. Commitments do not permit partial verification, removing a key pre-requisite for MITM optimization.

6.8. Side-Channel Considerations

Implementations MUST run constant-time big-integer operations. All private structures (seed, p, q, a, b, R, blocks) MUST be zeroized after use. Implementations SHOULD use masked big-integer arithmetic and avoid secret-dependent branching. The specification itself does not leak structure.

6.9. PKI Perpetuity and Key Rotation

Phi-Ns supports perpetual public keys. Since only one prime is public, the private key may be rotated infinitely without modifying the public key. Rotation replaces the decomposition of T with a new (a, b, R) structure, a new permutation, and a new block partition. The public identity remains stable. This property enables long-lived certificates and lightweight renewal for IoT, automotive, industrial and satellite environments.

6.10. Summary

Phi-Ns security relies on a new inversion problem combining quadratic relations and combinatorial decomposition. No known classical or quantum attack reduces the search below super-exponential cost. The recursive model allows arbitrarily high entropy from small public keys. The absence
of oracle feedback and the unpredictability of ENC(abR) define the core hardness of the primitive.

In designing Phi-Ns, conventional hash standards such as [FIPS180] and deterministic random bit generators as specified in [NIST90A] are assumed for commitments and key-derivation functions. The scheme is positioned in the context of post-quantum cryptography work [NISTPQC] and existing national guidance such as [ANSSI] and [NCSC]. Earlier work on one-time signatures [Lamport], hash constructions [SHA256], and prior descriptions of Phi-Ns [PhiNs] informed aspects of this design.

7. IANA Considerations

This document does not require any IANA actions.

No new registries are created, no parameters are allocated, and no existing registries are modified by this specification.

If future versions of Phi-Ns define wire formats, CBOR tags, or protocol

identifiers, those will be registered in separate companion documents.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, May 2017.

8.2. Informative References

[NISTPQC] National Institute of Standards and Technology, "Post-Quantum Cryptography Standardization Project", 2016-2025.

[ANSSI] Agence Nationale de la Securite des Systemes d’Information, "RGS v2.0 Rfrentiel Gnral de Scurit", 2020.

[NCSC] National Cyber Security Centre, "Cryptographic Recommendations for TLS, VPNs, and Messaging", 2022.

[Lamport] Lamport, L., "Constructing Digital Signatures from a One Way Function", October 1979.

[SHA256] Dobbertin, H., Bosselaers, A., and Preneel, B., "The Hash Functions MD5, SHA-1 and RIPEMD", 1996.

[PhiNs] Nsangani, J., "Phi-Ns: Quadratic Structured Decomposition for Asymmetric Cryptography", Patent Pending, 2025.

Author’s Address

Parfait Junior Nsiangani (Editor)
Email: jnsiangani@gmail.com