Rings and Modules for Identity-Based Post-Quantum Public-Key Cryptography

BASED ON THE PAPER **EPRINT.IACR.ORG/2014/794** BY DUCAS, LYUBASHEVSKY, AND PREST

Public Key Cryptography (PKC)

Also called Asymmetric Cryptography, because the public key and private key have different structures and complementary functions

	Public Key	**Private Key**
Encapsualtion *(Confidentiality during key management)*	**Encapsulate** a session key for its safe transfer	**Decapsulate** an encapsulated session key
Digital Signature *(Authentication and Identity Management)*	**Validate a digital signature** with respect to (the digest of) a message	**Create a valid digital signature** with respect to (the digest of) a message

PKC : Making Asymmetric Keys

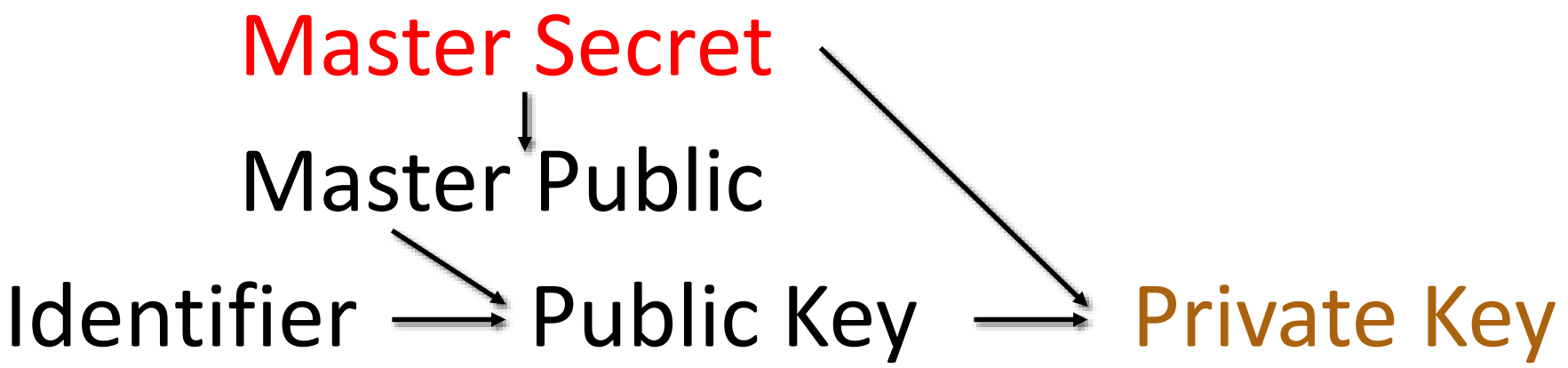
Seed ⟶ Private Key ⟶ Public Key ⟶ Digest

The arrows are deterministic and not invertible!

Seed and Digest are typically each 256 bits long.

The Public and Private Keys may need to be much longer.

Identity-Based PKC (IDPKC)



	Public Key	Private Key	Master Secret
Encapsualtion *(Confidentiality during key management)*	**Encapsulate** a session key for its safe transfer	**Decapsulate** an encapsulated session key	**Create a Private Key** from a Public Key

Where is IDPKC used?

Pairings-based cryptography became a hot topic in academic circles shortly after its promotion by Dan Boneh around 2001, and underwent a good deal of analysis and modification.

Around 2011, CESG published MIKEY-SAKKE as a set of IETF RFCs. The acronym expands as "Multimedia Internet Keying : Sakai-Kasahara Key Encryption", so called because it uses the MIKEY framework around the Sakai-Kasahara pairings-based primitive.

Secure Chorus is a set of standards for end-to-end secure enterprise comms, built on MIKEY-SAKKE.

The *Secure Chorus Group* of ten partners was established in February 2016 to promote these standards, so really IDPKC is only just getting off the ground (some fifteen years after its public disclosure).

Post-Quantum PKC (PQPKC)

Almost all PKC used today is vulnerable to quantum cryptanalysis. This is a concern for systems where confidentiality is supposed to be maintained for decades, or for systems where authentication 'trust anchors' are expected to remain reliable for decades.

Scheme	Type	Vulnerability
RSA	PKC	Shor's Algorithm (Factoring)
Diffie-Hellman (DH)	PKC	Shor's Algorithm (Discrete Log)
Elliptic Curve DH (ECDH)	PKC	Shor's Algorithm (Discrete Log)
Quadratic Residuosity	IDPKC	Shor's Algorithm (Discrete Log)
EC Pairings	IDPKC	Shor's Algorithm (Discrete Log)

PKC without any such (known) vulnerability is termed *Post-Quantum*.

Which PQPKC Primitives?

There are many areas of mathematics within which PQPKC primitives have been designed.

Species
Hash Functions
Structured Codes
Multivariate Quadratic Equations
Lattices and Rings
Elliptic Curve Isogenies
Braid Groups
…

Rings, Ideals, Modules, and Lattices

The paper [DLP] gives a fairly specific design for a ring-based IDPKC scheme, with many parameters fixed and a good deal of quantified analysis. I'll refer back to that periodically.

However, I want to take a fairly general approach in describing rings, ideals, modules, and lattices, to cover more possible design options, so will use notations slightly differently from ones appearing in the most influential papers.

Establish an isomorphism that preserves additive structure between a countable ring R and a lattice. Then we are free to consider ideals of R, R-modules, and their submodules, all with inherited metric structure.

The metric structures of interest will be the Euclidean norm (p=2) and other p-norms obtained from the coordinates (in the 'lattice' picture).

Lattice

In general, we want to work with a countable ring R, whose additive structure is isomorphic to a torsion-free Z-module of rank n.

Where many authors have preferred the structures arising from number fields, especially prime cyclotomic fields, it has been appropriate to consider the A_n^* lattice of rank n. However, it is sometimes just simpler to consider the regular cubic lattice Z^n.

Note that *all* n-dimensional lattices are isomorphic up to additive structure.

Considering Z^n, a lattice point is given by a string of n integer coordinates, and norms are straightforwardly determined from these coordinates.

Fix a basis for R, and that fixes a specific isomorphism $R \to Z^n$, for use throughout.

Ring

In general, we want to work with a countable ring R, whose additive structure is isomorphic to a torsion-free Z-module of rank n.

Not just any ring will do: there needs to be some statement to the effect that
"The product of a 'short' element with another 'short' element is 'short'*."

Normally our ring R will

- Be commutative
- Contain 1
- Be an integral domain (no zero-divisors)
- Be identified as $Z[X]/(f(X))$ for some monic integer polynomial $f()$ of degree n
- Be identified with Z^n via the geometric basis $(1, X, X^2, X^3, \ldots, X^{n-1})$

But none of these constraints is a logical requirement.

*The three notions of 'shortness' may be quite distinct.

Cyclotomic or Prime Ring?

The 'traditional' choice is a cyclotomic ring $(\text{integral domain}), Z[X]/(\Phi_m(X))$, with m a power of 2 or a prime. This has rank $n = \varphi(m)$, which is not prime.

In [Bernstein *et al*, 2015], a good case is made for using an integral domain of the form $Z[X]/(X^n - X - 1)$ with n prime. Prime degree integral domains have less intermediate field structure—and much less Galois structure—than cyclotomic rings.

With the usual basis $(1, X, X^2, X^3, \dots, X^{n-1})$, one can easily check how the infinity-norm of some product $a.b$ is bounded in terms of the 1-norm of a and the infinity-norm of b, so all these families of ring satisfy the maxim

"The product of a 'short' element with another 'short' element is 'short'."

Non-standard bases may also be considered.

NTRU Module

The "NTRU Module" Λ is a rank-2 R-module, a submodule of $R \times R$. It is generated by the rows of the matrix

$$\begin{pmatrix} s_1 & s_2 \\ q & 0 \\ 0 & q \end{pmatrix}$$

The Key Management Server (KMS) will choose the private data s_1, s_2.

Use some appropriate reduction algorithm to find a 'short' basis for Λ, and store that as the Master Secret.

Publish Λ itself as the Master Public data. This can be done by giving a basis in echelon form.

$$\begin{pmatrix} 1 & h_1 \\ 0 & q \end{pmatrix}$$

Public Key, Private Key

Use a cryptographic hash to convert a public identifier string (“Alice@gmail.com”) into a Public Key element of $R/q \equiv R^2/\Lambda$.

$$A \rightarrow \begin{pmatrix} 0 & A \end{pmatrix}$$

The Private Key corresponding to A is recovered by the KMS, using the Master Secret, by sampling a short vector $\begin{pmatrix} a_1 & a_2 \end{pmatrix}$ from the coset $\begin{pmatrix} 0 & A \end{pmatrix} + \Lambda$.

This sampling process must not leak (too much) information about the Master Secret, since the resulting sample is released as a user’s Private Key.

Ephemeral Encapsulation Primitive

Encapsulate:

- Choose some ‘short’ ring elements b_0, b_1, b_2.
- Combine them with Public Key A to produce ciphertext:

$$C_1 \coloneqq 2(b_2.h_1 + b_1)\ (mod\ q)$$

$$C_2 \coloneqq 2b_2.A + b_0\ \ (mod\ q)$$

Decapsulate:

- Combine the ciphertext with the user Private Key and *lift* back to R:

$$C_2 + C_1.a_1 = 2(b_1.a_1 + b_2.a_2) + b_0\ \ (mod\ q)$$

- This ‘shares’ the low bits of b_0.

Measuring Success

Decapsulation as described is successful if and only if every coefficient of $2(b_1.a_1 + b_2.a_2) + b_0$ lies in the range $\left(\frac{-q}{2}, \frac{q}{2}\right)$. So it depends critically on the inner product of $\left(\frac{1}{2}, a_1, a_2\right)$ with (b_0, b_1, b_2).

The scheme or protocol can fairly easily force the ephemeral (b_0, b_1, b_2) to satisfy particular length constraints, but what about the user Private Key (a_1, a_2)? Recall that this is to be chosen by sampling from the coset $(0, A) + \Lambda$.

Two proposals for potentially improving the success rate that I'd like to mention are

- Increase the rank of the module
- Require the low bits of b_0 to constitute a codeword of an error correction code

(Of course, with sufficiently large parameters it is always possible to obtain good success rates, but we'd prefer to keep parameters small if possible.)

Increasing to rank 3

For example, a rank-3 system would choose the module to be

$$\begin{pmatrix} s_1 & s_2 & s_3 \\ s_4 & s_5 & s_6 \\ q & 0 & 0 \\ 0 & q & 0 \\ 0 & 0 & q \end{pmatrix}$$

with echelon form

$$\begin{pmatrix} 1 & 0 & h_1 \\ 0 & 1 & h_2 \\ 0 & 0 & q \end{pmatrix}$$

Increasing to rank 3

Then the Public Key would be $(0 \quad 0 \quad A)$ while the Private Key would be $(a_1 \quad a_2 \quad a_3)$. There is perhaps then more flexibility to ensure that this vector be 'short'.

The ciphertext equations would be

$$C_1 \coloneqq 2(b_3.h_1 + b_1)\ (mod\ q)$$

$$C_2 \coloneqq 2(b_3.h_2 + b_2)\ (mod\ q)$$

$$C_3 \coloneqq 2b_3.A + b_0\ \ (mod\ q)$$

And the decapsulation equation would be

$$C_3 + C_1.a_1 + C_2.a_2 = 2\ (b_1.a_1 + b_2.a_2 + b_3.a_3) + b_0\ \ (mod\ q)$$

Codeword in b_0

Fix some $[n, k, d]$ binary code with good error correction properties (and k at least 256, say), and require that the low bits of b_0 form a codeword.

It is no longer necessary that *every* coordinate of $(b.a)$ lie inside the range $\left(\frac{-q}{2}, \frac{q}{2}\right)$. Now if only $n - \frac{d-1}{2}$ of the coordinates lie in range, then the usual lift *followed by error correction* will still recover the low bits of b_0.

The Lattice Problems

To begin cryptanalysis, before we try any formal reductions to any ‘standard’ assumptions (such as Ring-LWE for example), we must first identify clearly what are the *unreduced* lattice problems.

1) How do decapsulation failures leak information about the user Private Key (to someone who knows and controls the ephemeral data)?

2) Do the ciphertext equations leak private ephemeral data?

2b) Are fake user Private Keys hard to construct?

3) Do the user Private Keys leak data about the Master Secret?

We must also ensure that any *scheme* in which the *primitive* is deployed is itself secure, enforcing plaintext awareness, non-malleability of ciphertext, active reconstruction and validation of ephemeral data, and so on, at least to some pragmatic extent.

How do decapsulation failures leak?

Provided that we always enable all ephemeral data to be reconstructed from the low bits of b_0, and provided we ensure that any scheme does actually verify that the ephemeral data is correctly formed (and of fixed length), then *plaintext-awareness* is enforced and there is no scope for *malicious ephemerals*.

In that case, all that matters is the failure rate of decapsulation for the* Private Key.

If we can bound that failure rate at something like 2^{-32}, then it would cost an Attacker some four billion online queries to witness a single failure, deducing something about a from the knowledge that too many coordinates of $(b.a)$ lay out of range. There is probably no pragmatic attack to worry about in this case, especially if active network monitoring is deployed to detect any attempts to execute billions of queries.

*Consider what happens for an 'unusually poor' Private Key, not just the average behaviour, nor just the 'worst case'.

Do the ciphertext equations leak?

Even if we are using a rank-3 Λ, the analysis of the ciphertext equations can still reduce to lattice problems associated to rank-2 R-modules.

For example, isolating the single equation $C_1 \coloneqq 2(b_3.h_1 + b_1)\ (mod\ q)$, the associated *Closest Vector Problem* is to recover the shortest element in the coset of the lattice spanned by $(1, h_1)$ and $(0, q)$, offset by $\left(0, \frac{q+1}{2} C_1\right)$, which is almost certainly going to be $(-b_3, b_1)$.

So we must make sure that these Closest Vector Problems in $2n$ dimensions are sufficiently intractable. This is done by (appropriate choice of n and q and) appropriate distributions for each of b_0, b_1, b_2, b_3.

No ring (or isomorphic lattice) is inherently insecure: **what matters is the distributions of the key elements used within it**. The [DLP] paper identifies some distributions that enable formal reductions to standard hard problems (for the usual power-of-two cyclotomic ring).

Are fake Private Keys hard to construct?

A *fake* Private Key is one that comes from an Attacker's attempts to find *any* short element of the coset $(0,0,A) + \Lambda$. (This may or may not involve first finding a 'good' basis for Λ.)

This is a *Close Vector Problem*, where any sufficiently short answer will do: though the shorter the better. Note that because the coset is randomly selected, there will be *many* potentially useful fake Private Keys out there.

Contrast this to the problem of faking a signature. If a signature vector is too long, then it won't pass validation, and is not in fact a fake; it is of no use to an Attacker. But if an IDPKC Private Key a is a bit longer than normal, then there is still some chance that $(b.\, a)$ will lift correctly, enabling decapsulation of b_0 and hence recovery of the session key.

We must ensure that it is hard to solve Close Vector Problems in (say) $3n$ dimensions.

Of course, the *Short Basis Problem* for Λ must also be hard, so that the Master Secret (or some other basis that is nearly as 'good') can't be discovered either.

Do the user Private Keys leak?

A major contribution of the [DLP] paper is to refine the security argument for this question about Gaussian sampling, so that a significantly shorter Private Key a can be sampled, all the while still ensuring that any information that is leaked about the Master Secret will not be 'accessible'.

In an ideal world, the distribution of (h_1, a_1, a_2, A) ought not depend on whether a_1, a_2 were chosen first and then $A = a_2 - h_1.\, a_1 \ (mod\ q)$, or whether A and h_1 were chosen first and then a_1, a_2 sampled accordingly.

If the two distributions cannot be made identical, one common alternative is to bound their **statistical distance**.

The nice idea from [DLP] is to bound their **Kullback-Leibler divergence** instead. This leads to a sampling algorithm that can find significantly shorter Private Keys than would be possible were the statistical distance used.

Computational Issues

?

Summary

Global demand for PQIDPKC will probably pick up over the next few years.

Ring-based techniques are a clear leader for PQIDPKC, at least from today's vantage point, but they seem to require parameters that are quite different from (and quite a bit larger than) the kinds of parameters used for more 'basic' ring-based cryptography.

There is probably quite a lot of unexplored 'design-space'!

Recommended Reading

Ducas, Lyubashevsky, Prest (2014)

- Efficient Identity-Based Encryption over NTRU Lattices

Gama, Nguyen (2008)

- Predicting lattice reduction

Lyubashevsky, Peikert, Regev (2013)

- A Toolkit for Ring-LWE Cryptography

Bernstein, Chuengsatiansup, Lange, van Vredendaal (2016)

- NTRU-Prime

MIKEY SAKKE Internet RFCs

- Only joking: I wouldn't recommend reading any Internet RFCs…