Considerations in the usage of a Round5 in a Quantum-resistant hybrid design

Royal Philips N.V., Netherlands.

1 Introduction

Due to the inherent vulnerability of RSA and Elliptic Curve cryptography to attacks by quantum computers and the relatively long time period that public key encryption algorithms must guarantee the confidentiality of their secrets, a transition to quantum-secure alternatives has been initiated by the U.S. Government and the information security community. Standardization bodies such as National Institute of Standards and Technology (NIST) or European Telecommunications Standards Institute are currently in the process of evaluating and standardizing Post- Quantum Cryptography.

Round5 is a leading candidate for NIST PQC key-encapsulation and public-key encryption. Round5 resulted from the merge of the NIST PQC first round candidates Round2 and HILA5 and on January 30th 2019 it was accepted as a NIST PQC second round candidate.

Round5 relies on the General Learning with Rounding (GLWR) problem to unify the well-studied Learning with Rounding (LWR) and Ring Learning with Rounding (RLWR) lattice-problems. Thus, Round5 enables a single description and implementation of multiple algorithms relying on different underlying problems. This gives the user the flexibility to choose the parameter set and algorithm that fits his application best.

Round5 design choices have been made with security and performance in mind. The flexible and unified design allows the user to choose the configuration that fits best her security and performance needs. Learning with Rounding allows for a lower bandwidth overhead than typical Learning with Errors-based proposals. Round5's ring instantiations further rely on prime-order cyclotomic polynomial rings that enjoy well-established proofs of security and offer a large design space, allowing for a fine-tuning of the ring dimension. It is thus easy to scale-up or scale-down Round5's parameters to target different security targets. The usage of power-of-two moduli q and p makes modular operations fast so that Round5 is very efficient on a variety of platforms. Fixed-weight ternary secrets ensure fast operation and low failure probability. Finally, the usage of the strong constant-time XEf error correction code allows Round5 to support the smallest configuration parameters among the NIST lattice-based proposals, and thus, offer the best performance in terms of bandwidth, CPU, and memory usage. Since XEf is constant-time, timing attacks on the error correction are not feasible.

2 Comments on Virgil Securiy quantum-resistant hybrid design

Virgil Security provides solutions to enable secure communications with minimal effort. For instance, Virgil E3Kit allows for the development of end-to-end secure applications. The idea is that an application provider can register in Virgil Security platform, integrate the E3Kit in its application, and let its customers communicate in a secure way. When Alice and Bob – customers of the application provider – join the system of the application provider, they generate some cryptographic keys that are stored in Virgil Security's platform. When Alice wants to talk to Bob, Alice retrieves Bob's public-key from Virgil Security platform and uses it to setup a secure channel. Alice uses her private key to sign the message. Upon reception of the authenticated and encrypted message, Bob retrieves Alice's public-key to verify her identity and decrypts it using his private keys.

Virgil Security has integrated Round5 in its E3Kit designing a quantum-resistant hybrid solution [1]. Virgil security has drafted a white paper [2] on its approach to a quantum-resistant hybrid design.

The rest of this section includes comments on this white paper regarding its rationale, available options, and performance analysis. These comments aim at trying to achieve the best possible understanding on the design of quantum-resistant hybrid solutions and highlighting the existing trade-offs in different PQC algorithms that are currently under review in the NIST PQC standardization project.

2.1 General comments

The white paper of Virgil Security gives a good overview on the overall design. The document would benefit from a new section with a brief introduction about Virgil Security and the type of product/service in which the hybrid solution is integrated including its rationale and needs.

2.2 Comments on "what is PQC"

It is recommended to include a list of *threats* in section "what is PQC". The main one refers to the so called "harvest and decrypt" attack. This is the main threat anyone should worry about today since long-term data confidentiality might be compromised in the future if a passive attacker today collects data, and in the future uses his quantum-computer to break it. Source authentication is considered less important since it would require an active attacker. For communication links being established today, this is not an issue since (as far

[1] https://developer.VirgilSecurity.com/docs/e3kit/fundamentals/supported-algorithms/

[2] https://virgilsecurity.com/wp-content/uploads/2020/03/hybrid-post-quantum-encryption.pdf

as we know) quantum-computers have not been built yet. Source authentication is an issue in areas such as code signing, in particular, if it requires any type of hardware to securely store public-keys. Not having the right solution today might imply a big transition issue in the future. Furthermore:

- Section "what is PQC" could be improved by including a brief description on related work on hybrid solutions/standardization activities. Some examples are as follows:
  - NIST PQC project [3].
  - IETF: General considerations [4].
  - IETF: TLS hybrid [5], [6].
  - IETF: IKEv2 hybrid [7].
  - IETF: Hybrid certificates [8].
- It would be useful to include a list of requirements in section "what is PQC". This list of requirements might be slightly different than, e.g., the list of requirements in the hybrid TLS or IKEv2 documents, however, the list there might be useful as starting point.
- "Re-encrypting currently encrypted (and stored) data" is not the problem. Usually data is stored using symmetric crypto that is not so badly affected, and we can easily start using longer 256 bit keys already today. The problem is data in transit that is protected using asymmetric crypto. If an attacker records the communication links today, then he will be able to break exchanged encrypted data once he has access to a quantum computer.
- When talking about timelines, the example of $Q - T < C$ could be included. Q is the time when the Quantum computer is available; T is today; C is the number of years that the exchanged data needs to remain confidential.

2.3 PQC: Round5 and Falcon

Virgil Security has selected both Round5 and Falcon for integration into Virgil E3Kit. Some comments are as follows:

- The initial paragraph describing the potential candidates can be improved giving further details: what would be the impact of very long keys in Virgil Security's system? Can they be cached? if no, why not? what are the security requirements? (CPA/CCA)? what is the failure rate of your system? what are the consequences for the desired failure rate of the chosen algorithms? what would be the impact of SIDH on CPU performance, e.g., on mobile devices: battery usage, security issues (DoS)? This part could also benefit from a more systematic description. For instance, the figures in page 3 include

[3] https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
[4] https://datatracker.ietf.org/doc/draft-hoffman-c2pq/
[5] https://tools.ietf.org/html/draft-whyte-qsh-tls13-06
[6] https://tools.ietf.org/html/draft-campagna-tls-bike-sike-hybrid-03
[7] https://tools.ietf.org/html/draft-tjhai-ipsecme-hybrid-qske-ikev2-02
[8] https://tools.ietf.org/html/draft-truskovsky-lamps-pq-hybrid-x509-01

a given classification (Goppa, QC code, Isogeny, Structured/unstructured, multivariate, symmetric). It would be useful to list all of them in this initial paragraph discussing pros/cons.
- The choice of Round5 and Falcon comes suddenly. Why lattice-based? why structured solutions? It would be useful to have a list of requirements / decision criteria first. CPA or CCA KEM, security level, failure rate, performance,... Maybe some of these terms are too technical for a white paper, but still, they can be pretty useful to assess the decision. An end user will also appreciate a deeper description.
- In the context of Round5, it would be useful to explain the rationale for the choice of R5ND_5CCA_0d and the specific configuration used (SHAKE or AES), side-channel countermeasures, etc. It would also be useful to state which alternative Round5 parameter sets could be used in other applications or depending on the user. In particular, a non-ring parameter set could be useful to integrate. It is clear that not all end users might have the knowledge to make such a complex decision; however, some users might have that knowledge, and prefer a solution with less structure.
- Instead of having an hyperlink in the text, please, consider using a footnote or a standard reference.
- "... based on the lattice problem which is hard to solve even by quantum computers" Clarify the problem? Even when a quantum computer is available.
- Figures in page 3 are for Category 1, but the choices made, at least for Round5, are for level 5. It would be better to include the right figures.
- Probably due to a copy-paste issue, the quality of the figures in page 3 is low.
- We chose one of the + ones (Falcon): it would be good to highlight it in the figure.
- The texts of Round5 and Falcon are copied and there are some mistakes. For instance, "CPU and memory usage" does not fit in the reading flow. It would be good to list which security properties they provide and which algorithm in the specification is used.
- Notation and font type/size are not consistent. For instance, the table font looks different than the text. Sometimes kb is used and sometimes kbyte.

2.4 Virgil Security's Hybrid Algorithm Approach

This section describes some technical aspects. Further details are required:

- Why hybrid for both confidentiality and authentication? Needs might be different.
- Why hybrid with a single algorithm?
- Something is in the figure wrong. Likely, the top right blue box should be an ECC KEM.
- Although this is a white paper, it would be useful to have a more clear description of what is what and how it is used in the system. For instance:

which R5-KEM is used? are the security guarantees (semantic security of R5-KEM and ECC KEM equal? or different? If so, why? Which method do you use to derive **a**, what are the encapsulated key and the shared key in Round5? has anything been changed? Which HKDF is used? sk1 and sk2 are not defined; + is not defined; in the hybrid TLS document above there is a discussion on methods to derive keys. Is this method one of them?; links to the standards that are used should be included (ECIES, AES-GCM,...); How does the ECIES fit in the figure? in which parts? the right box that starts with "per recipient" is about (?) the information that is protected when Alice talks to Bob. If this is the case, it should be explained. The document would be better if it is explained how frequently keys are exchanged, etc. The diagram is about hybrid encryption, it would be better to have a description for the hybrid certificates; it would be beneficial to show how the hybrid solution is used in the system (high level diagram including user registration, ...)

2.4.1 Suggestions

- Include timing numbers for the chosen configurations
- Include timing numbers for different Round5 configurations (e.g., classic, hybrid with R5ND_5CCA_5d, hybrid with R5N1_5CCA_0d). In the timing numbers, it would be useful to compare CPU timing but also latency due to the key exchange)
- Round5 has a non-ring parameter set with a small ciphertext. Since public-keys are retrieved from Virgil Security's platform, and this seems to be a one-time task, this parameter set might offer a communication overhead similar to the chosen ring configuration, at an acceptable computational overhead. Note that the public-keys are pretty big and this is a disadvantage. It would be interesting to test it in practice.
- If enough data is available, it might be feasible to cluster data based on type of connection (mobile,...)