**Chapter 8 – Heuristics**

**8.1 - Assumptions**

SQISIGN is built on a few critical assumptions: primality of certain parameters, quadratic residues and non-residues, random lattices, and the importance of the ideal size. Primality is used in several key places when discussing the security of the scheme, but perhaps the most important is in the construction of the finite field where the modulus *p* is a very large prime number, thereby ensuring that the field cannot be subjected to a factorization attack.

Tied together with primality is the idea of quadratic residuosity, where a quadratic residue is a number that can be expressed as the square of an integer within a given modular system. For example, if our prime number *p* is equal to 7, we would use modular arithmetic for the square numbers up to 7. This is going to give us a set of numbers that we can call residues. In this example, our residues would be the set {0,1,2,4} and the remaining set {3,5,6} we can refer to as non-residues. Both of these sets of numbers will be incorporated into SQISIGN in different but important ways. The residues will ensure that the structure of the elliptic curves remains intact by ensuring that certain points fall on the curve, while the non-residues are used in the generation of the isogeny path. This provides an extra layer of security as an attacker would have a very tough time telling the difference between the residues and non-residues.

Another assumption we can make is that the lattices will be random. As has been discusses previously, lattices are used in key generation, signing, and transformations. Randomness of the lattices ensures unpredictability, making it impossible for an attack in which there is an attempt to reconstruct the lattices, ensuring the security of the private key.

The final assumption to be discussed is perhaps the most obvious, the notion of finding the optimal ideal sizes. If an ideal is small the security isn’t as strong, but larger ideals will lead to greater computational overhead, potentially making the scheme far too expensive.

**8.2 - Subroutines**

The strength of a cryptographic protocol often lies in the intricacy of its subroutines, and SQISIGN is no different. The core of the scheme involves the supersingular elliptic curves and isogeny paths, both of which involve very complex algebraic structures. There are several subroutines critical to the security of the protocol: RandomEquivalentPrimeIdeal, KeyGenKLPT, SpecialEichlerNorm, IdealToIsogenyEichler, and other further randomizations. We have discussed most of the mechanics of these subroutines already, so here the focus will be their contributions to the security of the scheme:

* + RandomEquivalentPrimeIdeal generates a random ideal equivalent to a given ideal, ensuring the randomness of the structure adds unpredictability to the protocol. This is a large part in what makes the scheme resistant to a lot of known attacks.
  + KeyGenKLPT, which we’ve covered a few times, ensures that the ideal corresponds to a valid isogeny. Without this subroutine, the private key could be invalid or easy to guess.
  + SpecialEichlerNorm is the subroutine that handles performance maintenance without sacrificing security.
  + IdealToIsogenyEichler is the subroutine that actually converts the ideals to isogenies, ensuring that the protocol preserves the security guarantees and works seamlessly.
  + Last, there are other randomizations that are used in SQISIGN , making it even harder to attack any one part of the scheme.

**Chapter 9 – Security Analysis**

**9.1 – Security Reductions**

There are a few properties that the sigma protocol must satisfy: correctness, special soundness, weak Honest-Verifier Zero Knowledge (wHVZK), Impersonation Under Passive Attack (IMP-PA), and finally Existential Unforgeability Under Chosen Message Attacks (EUF-CMA) after the Fiat-Shamir Transform.

The correctness property says that the verifier will always accept a valid proof. The verifier function is V(x, comm, chall, resp)=1, where x is the public key, comm is the commitment, chall is a random challenge from the verifier, and resp is the response from the prover.

Special soundess means that if a prover can produce two valid responses to the same challenge, they must know the private key. However, if a cheating prover only knows the public key *x* and does not witness the private key *w* they can’t force the verifier to accept.

wHVZK ensures that even if the verifier is honest during the protocol, they can’t get any additional information about the private key. This is only partially handled in SQISIGN in the sense that the scheme doesn’t create “fake” results as some other schemes do, but because there are so many isogenies created during the key generation it would be impossible to extract the real one.

IMP-PA security says that an attacker simply observing the protocol can’t forge the signature or get information about the private key. There is a theorem that states that if a scheme satisfies both special soundness and wHVZK, it also satisfies IMP-PA.

There is another theorem that says any scheme that satisfies IMP-PA and undergoes a Fiat-Shamir transform, that scheme then satisfies EUF-CMA.. This says that even after observing signatures on other chosen messages, an attacker can’t forge a valid signature for any new message.

**9.2 – Resistance to Known Attacks**

Carefully choosing certain parameters in the sigma protocol ensures that (so far) there is no way to break the security that SQISIGN offers.

Theoretically, if one could compute the endomorphism ring or the isogeny path, they could break the protocol (this is true not just of SQISIGN but any isogeny-based scheme). However, no such algorithm exists even in quantum computing.

Key recovery is also not possible as an attacker would have to either randomly guess the private key or reverse-engineer the isogeny from the destination curve (public key). Again, there is no algorithm that can reverse-engineer an isogeny, and while one could theoretically randomly guess the private key, the probability of this is close enough to zero that it is deemed quantum resistant.

Forgery is another attack that has been proven to be ineffective, as even if an attacker observes multiple valid signatures, the randomness and unpredictability in the protocol means they can’t use any of the information to construct a signature for a new message. This is due to the fact that the signature generation depends on the private key and isogeny path, neither of which can be obtained from the public key or other signatures.