Chapter 1

Introduction

One of the core goals of cryptography is to be able to offer security and privacy without sacrificing functionality. To do this, cryptographers define notions of both correctness and security. We can then build systems that we prove satisfy both of these definitions. Correctness states that the system provides the functionality we want, while security is essentially the notion that we can effectively hide information from an adversary. Depending on the notion of security, this adversary may be all powerful (information-theoretic or computationally unbounded), may run only in polynomial time (computationally bounded), or even bounded by a specific polynomial runtime like $O(n^2)$ (a fine-grained adversary). This thesis will focus on realizing three different functionalities, each one hiding different information, and three different notions of security.

In the first part of this thesis, we discuss results in the area of Topology Hiding Computation (THC). THC is a generalization of secure multiparty computation. In this model, parties are in an incomplete but connected communication network, each party with its own private inputs. The functionality these parties want to realize is evaluating some function (computation) over all of their inputs. On the security and privacy side, these parties want to hide both their private inputs and who their neighbors are. So, the goal is for these parties to run a protocol, communicating only with their neighbors, so that by the end of the protocol, they learn the output of the function on their inputs and *nothing else*, including information about what the communication network looks like other than their own neighborhood.

It turns out that THC is a difficult notion to realize, and so being able to achieve it from many different standard cryptographic assumptions is useful. In this thesis, we show how to get THC from the Learning-With-Errors assumption, adding to the list of standard assumptions already known to work.

This subfield is also mired in impossibility results. First, there is an impossibility result for THC stating that it is impossible to design THC against adversaries that can "turn off" parties during the protocol [MOR15a]. In essence, an adversary that has this power can always learn something about the graph. So, protocols for this model are designed to limit the amount of information leaked as much as possible. We present another impossibility result when considering almost any kind of asynchronous model: channels between parties now have unknown delays on them, and an adversary

may be able to control the delay on some of those edges (though messages must be eventually be delivered in unbounded polynomial time).

In the second part of the thesis, we focus on Adversarially Robust Property Preserving Hashes (PPH). PPHs are a generalization of collision resistant hash functions (CRHFs). Recall that a CRHF is a family of hash functions $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ that is compressing ($m < n$), and has the property that no Probabilistic Polynomial-Time (PPT) adversary given h can find two inputs $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$ except with negligible probability. Looking at CRHFs from a slightly different perspective, their functionality is to preserve the equality predicate between two inputs while compressing them, preventing an adversary from coming up with inputs where the equality predicate is not preserved (even though these inputs exist, they are hard to find). A PPH is also a compressing function that preserves some other property. For example, we focus on the property of "gap-hamming" distance: if two inputs are close in hamming distance, then their hashes are also close, and if two inputs are far, their hashes are also far. Note that this example is of a promise predicate since we do not care about inputs that are neither close nor far.

Property preserving hashes were a new cryptographic primitive that we designed, and so it was important to understand what was possible and what was not in our new model. We found strong connections between One-Way Communication Complexity (OWC complexity) and our primitive. Fortunately for us, OWC is a rich, well-studied area of complexity. In many cases, we could not directly port OWC results to our setting, but we could use their proof techniques and get lower bounds. Once we had these lower bounds, we had a much better sense of what was and was not possible, and could construct PPHs more easily.

In the final chapter, for a different perspective, we design a public-key cryptography functionality against weak adversaries. This functionality allows for a party to publish a public key while they keep the secret key, and any other party to use that public key to encrypt a message that only the party with the secret key can decrypt, *hiding* that message from any eavesdroppers. Unlike in standard models for cryptography, however, our eavesdroppers much less powerful: their runtime is bounded by an explicit polynomial instead of "probabilistic polynomial-time", e.g. can only run in time $O(n^{100})$. This, of course, is only interesting as a cryptographic notion if the adversary has the same or more time to run than the honest parties. So, we get a notion of cryptography we call *fine-grained*: for examples, honest parties might only need to run in $O(n^2)$ time, while any adversary running in time $O(n^{100})$ time still cannot glean any useful information with some probability. Motivating the study of this cryptography is the fact that it does not rely on any of the normal cryptographic assumptions, including $P \neq NP$, $BPP \neq NP$, or even the assumption that one-way functions exist.

However, due to its fine-grained nature many standard cryptographic reductions (like Goldreich-Levin hardcore bits [GL89]) no longer work. And so, in this thesis we demonstrate variations that can work in our setting, including a notion of fine-grained one-way functions, fine-grained hardcore-bits, a fine-grained key exchange, and finally fine-grained public-key cryptography.

1.1 Results

In this thesis, we explore these three different notions of hiding while preserving a functionality:

1. hide private inputs and network topology while preserving computation,
2. hide as much information as possible while preserving a property between inputs,
3. and hide messages from a weak adversary while preserving the communication and fine-grained runtime.

Topology-Hiding Computation To address the first perspective, we explore the realm of topology-hiding computation (THC). THC is a generalization of secure multiparty computation (MPC), where we hide not only each party's input, but also the communication graph; parties know who their neighbors are, but learn nothing else about the structure of the graph. In 2017, we showed that THC is possible against a very weak adversary [ALM17b], but there were still many open questions surrounding the nature of THC. This thesis addresses two of them.

The first is simply what standard cryptographic assumptions can imply THC. Prior work from [BBMM18] shows that THC implies oblivious transfer (OT). This means that we will need to make some kind of cryptographic assumption that will imply public key crypto. Both [AM17] and [ALM17b] used a very common cryptographic assumption: Decisional Diffie-Hellman (DDH). Then, in a later update, [ALM17c] showed that you could achieve this with the Quadratic Residuosity (QR) assumption. In this thesis, we will show how the Learning-With-Errors (LWE) assumption can achieve the same results.

The second open question we address here regards asynchronous networks. All prior work in THC was in the synchronous model: at every round, a party either sent a message and it would be received that same round or, in the fail-stop model ([BBMM18, LZM+18]) not send a message at all. It was an open question whether or not we could achieve THC in an asynchronous network. Unfortunately, for all of the standard notions of asynchronous, we show that THC is impossible. This is primarily because all of these notions give power to the adversary to control when messages are sent. One can immediately notice that an adversary should not be able to see who is communicating with whom, but even taking that out of the equation, an adversary with any control over an edge can learn something about the structure of the graph. This means that if we want to design a THC protocol over a network that is not fully synchronous (as real-world networks tend to be), the model of the network cannot give the adversary control over the timing of sending and receiving messages.

Adversarially Robust Property Preserving Hashes and One-Way Communication Lower Bounds Addressing the second perspective, we look at a new cryptographic primitive: adversarially robust Property Preserving Hashes (PPH). The inspiration here is two-fold. First, collision-resistant hash functions (CRHFs) are

a staple in cryptography: no PPT adversary can produce an $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$ except with negligible probability over h sampled from the CRHF family and coins of the adversary, and hence one can preserve the "equality" predicate on compressed inputs even against adversarially chosen inputs. We want to compress inputs while maintaining some property other than equality between them, even in the presence of adversarially chosen inputs. The properties we considered were those that already had non-robust constructions, in particular locality-sensitive hashing (LSH) [IM98]. An LSH family is a hash function family $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ that compresses inputs ($m < n$) and is *mostly* correct. For example, say we were preserving a gap-ℓ_p norm for a gap between r and cr for some constant $c > 1$ (note that in our case, gap-hamming is equivalent to gap-ℓ_0 norm). Then, we have a threshold τ so that for any pair $x_1, x_2 \in \{0,1\}$

- if $||x_1 - x_2||_p < r$, $||h(x_1) - h(x_2)||_p < \tau$, and
- if $||x_1 - x_2||_p > cr$ then $||h(x_1) - h(x_2)||_p \geq \tau$

with high probability over our choice of h sampled from $\mathcal{H}$.

We can use almost the same language to define PPHs for some property $P : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, except our probability will be negligible and taken over both our sampled hash function and the coins of a PPT adversary. We also generalize the predicate evaluation: instead of using the same predicate on hashed values, we can use an "evaluation function" called Eval. So, even in the presence of (PPT) adversarially chosen values of x_1 and x_2,

- if $P(x_1, x_2) = 0$, then $\mathsf{Eval}(h(x_1), h(x_2)) = 0$, and
- if $P(x_1, x_2) = 1$, then $\mathsf{Eval}(h(x_1), h(x_2)) = 1$

with all but negligible probability. In other words, for any PPT adversary given h sampled from $\mathcal{H}$, the probability that the adversary produces x_1 and x_2 such that $P(x_1, x_2) \neq \mathsf{Eval}(h(x_1), h(x_2))$ is negligible.

The link between robust PPH and LSH was very clear, but less obvious was the link between PPH and one-way communication (OWC): in essence, compressing an input as much as possible for a OWC protocol is akin to compressing an input as much as possible while preserving some property of that input as we would want to do for a hash function. These connections led to the following results, which will be included in the thesis.

- OWC is a rich field with a lot of work detailing lower bounds on the complexity of certain predicates. These lower bounds *almost* directly translated to impossibility results for PPHs. However, we needed to be careful because OWC lower bounds dealt with constant error, where as cryptographers, we care about negligible error. We characterized these OWC lower bounds and how they mapped to PPH lower bounds. We showed that a class of predicates we call "reconstructing predicates" could not have PPHs. This class includes useful predicates like GreaterThan, Index, and ExactHamming.

- With these lower bounds and previous results from LSH in mind, we turned to constructing a PPH for Gap-Hamming: the promise predicate where we can distinguish between pairs of points that are $(1-\epsilon)d$ *close* in hamming distance or $(1+\epsilon)d$ *far*. For all pairs within the gap, we "don't care" how our PPH behaves. We build two constructions for this predicate, each with different drawbacks and benefits. Our first construction relies only on collision-resistant hashing. Our second uses a new assumption related to the hardness of syndrome decoding [AHI+17].
- Rounding out this paper and its myriad of definitions, we demonstrate that even for very simple predicates, we require collision resistance, and even if we weaken our main definition of a robust PPH, we still need one-way functions.

These results were published in ITCS 2019 [BLV19].

In unpublished follow-up research with Cyrus Rashtchian, we explored the intricacies of the CRHF method for constructing gap-hamming PPHs. We fleshed out the relationship between d, the "center" of our gap, and the size of the hash function output. We also discussed how to use this kind of construction to get a gap-ℓ_1-norm PPH and found that with standard methods of transforming ℓ_1 into ℓ_0, it could not work.

1.1.1 Fine-Grained Cryptography and Barriers

Addressing the final perspective, this thesis will discuss our work in Fine-Grained cryptography. With my coauthors Lincoln and Vassilevska Williams, we built the first fine-grained key exchange built from fine-grained assumptions. The premise was to explore what cryptography could be like in "Pessiland," a world in which there are no cryptographic one-way functions (which also implies no public key cryptography). We looked at this through the lens of fine-grained complexity, using both assumptions and reduction techniques from that field.

While designing cryptography for this setting, we came across multiple barriers. The first was that some fine-grained problems would not lend themselves to build cryptography without refuting NSETH [CGI+16]. So, we would need to use a fine-grained problem that was not associated with that barrier. Another form of barrier we ran into was worst-case to average-case reductions for problems that would make for good cryptography. Even now, the only worst-case to average-case reductions for fine-grained problems are for counting versions of these problems. Unfortunately, counting-style problems do not lend themselves well to building cryptography, as there are no longer small enough witnesses. Finally, many standard cryptographic reductions no longer work in a fine-grained world, since they take a non-trivial polynomial amount of time, and so a challenge we faced was to ensure all of our reductions were fine-grained and build different types of primitives based off of these restricted reductions.

Despite these difficulties, we achieved the following results:

- Assuming that an average-case version of Zero-k-Clique requires $n^{k-o(1)}$ time, we constructed a non-interactive key exchange that required honest parties to

run in $O(N)$ time and an adversary to run in time $\tilde{\Omega}(N^{1.5-\epsilon})$, where ϵ decreases with respect to k.[1]

- Generalize the properties of Zero-k-Clique to construct this key exchange: plantable, list-hard, and splittable.

- Define and construct fine-grained hard-core bits.

This work was published in CRYPTO 2019 [LLW19].

In unpublished follow-up work, we also show how to use another property of Zero-k-Clique to construct a key exchange where the adversary now needs $\Omega(N^{2-\epsilon})$ time. This N^2 gap is the best we are able to do since we base these constructions on Merkle puzzles [BM09].

1.2 Related Works

1.2.1 Related Work to Topology Hiding Computation

Hiding the network topology is more of a concern in the literature on *anonymous communication* [Cha81, RR98, SGR97]. Here, the goal is to hide the identity of the sender and receiver in a message transmission. A classical technique to achieve anonymity is the so-called mix-net technique, introduced by Chaum [Cha81]. Here, *mix* servers are used as proxies which shuffle messages sent between peers to disable an eavesdropper from following a message's path. The onion routing technique [SGR97, RR98] is perhaps the most known instantiation of the mix-technique. Another anonymity technique known as *Dining Cryptographers networks*, in short DC-nets, was introduced in [Cha88] (see also [Bd90, GJ04]).

Synchronous Networks. Existing constructions to achieve *topology-hiding communication* over an incomplete network focus mainly on the cryptographic setting for passive corruptions. The first protocol was given in [MOR15b]. Here, the authors provide a feasibility result for a broadcast protocol secure against static, passive corruptions. At a very high level, [MOR15b] uses a series of nested multi-party computations, in which each node is emulated by a secure computation of its neighbor. This emulation then extends to the entire graph recursively. In [HMTZ16], the authors improve this result and provide a construction that makes only black-box use of encryption and where the security is based on the DDH assumption. However, both results are feasible only for graphs with logarithmic diameter. Topology hiding communication for certain classes of graphs with large diameter was described in [AM17]. This result was finally extended to allow for arbitrary (connected) graphs in [ALM17a].

The fail-stop setting was first considered in [MOR15b] where the authors give a construction for a fail-stop adversary with a very limited corruption pattern: the adversary is not allowed to corrupt any complete neighborhood of a party and is

[1]The tilde in $\tilde{\Omega}$ ignores any *subpolynomial* factors.

also not allowed an abort pattern that disconnects the graph. In this work, the authors also prove that topology-hiding communication without leakage is impossible if the adversary disconnects the graph. A more recent work, [BBMM18], provides a construction for a fail-stop adversary with one bit/non-negligible leakage but requires that parties have access to secure hardware modules which are initialized with correlated, pre-shared keys.

In the information-theoretic setting, the main result is negative [HJ07]: any MPC protocol in the information-theoretic setting inherently leaks information about the network graph. They also show that if the routing table is leaked, one can construct an MPC protocol which leaks no additional information.

Other Network Models. Katz et al. [KMTZ13] introduce eventual-delivery and channels with a fixed known upper bound. These functionalities implement communication between two parties, where the adversary can set, for each message, the delay after which it is delivered.

Cohen et al. [CCGZ16] define different channels with probabilistic delays, for example point-to-point channels (the SMT functionalities) and an all-to-all channel (parallel SMT, or PSMT). However, their PSMT functionality cannot be easily modified to model THC, since the delivery time is sampled once for all parties. One could modify the SMT functionalities and use their parallel composition, but we find our formulation simpler and much better suited for THC.

1.2.2 Related Work to Property Preserving Hashing

The notion of compressing an input while preserving something about it intersects many areas of theoretical computer science. The idea of property-preserving hashing underlies sketching [MP80, MG82, AMS96, CM05, CCF04], compressed sensing [CDS01], and locality-sensitive hashing (LSH) [IM98]. However, much of this work does not deal with any kind of adversary. Much like how Universal Hash Functions [CW77], which can be used to test the equality of data points, cannot be used for that purpose in the presence of adversarially generated inputs, LSHs give no guarantees in this setting either. The answer to Universal Hash Functions came in the form of pseudorandom functions (PRF) [GGM86], universal one-way hash functions (UOWHF) [NY89] and collision-resistant hash functions (CRHF) for different adversarial settings (e.g. oracle-access or direct access to the code of the hash function). We do something similar with respect to LSH and explore properties other than locality.

Along that line, Mironov, Naor and Segev [MNS08] showed *interactive protocols* for sketching in such an adversarial environment, achieving our goal, but requiring interaction. In contrast, we focus on non-interactive hash functions. Hardt and Woodruff [HW13] showed negative results which say that linear functions cannot be robust (even against computationally bounded adversaries) for certain natural ℓ_p distance properties; our work will use non-linearity and computational assumptions to overcome the [HW13] attack. Finally, Naor and Yogev [NY15] study adversarial Bloom filters which compress a set in a way that supports checking set membership; we will use their lower bound techniques in Chapter 3.6.

Secure Sketching. Fuzzy extractors and secure sketching [DORS08] aim to achieve similar goals to PPHs. Both of these seek to preserve the privacy of their inputs. Secure sketches generate random-looking sketches that hide information about the original input so that the original input can be reconstructed when given something close to it. Fuzzy extractors generate uniform-looking keys based off of fuzzy (biometric) data also using entropy: as long as the input has enough entropy, so will the output. As stated above, both guarantee that if inputs are close, they will 'sketch' or 'extract' to the same object. Now, the entropy of the sketch or key guarantees that randomly generated far inputs will not collide, but there are no guarantees about adversarially generated far inputs. For our definition of robust PPHs, we want to make sure that adversarially generated inputs, close or far, will not fool our hash function evaluation.

One-Way Communication. Key to this thesis is One-Way Communication (OWC) [Yao79, KNR95]. OWC is the study of protocols between two parties (e.g. Alice and Bob) that only allow for communication in one direction (e.g. Alice to Bob), with the goal of evaluating a joint function on their inputs. The goal of OWC is to minimize the total communication while maintaining correctness of the protocol with high probability. From another perspective, OWC can be seen as hashing Alice's input to Bob, with the goal of compressing her input as much as possible while preserving correctness. Lower bounds in the OWC model translate directly into impossibility results for PPHs. And so, we will reference OWC results from [Woo04, Woo07, JKS08] (who proved OWC lower bounds for hamming distance with certain parameters), and strengthen them to be lower bounds for PPHs, using strategies from [JKS08] in Chapter 3.3.

1.2.3 Related Work to Fine-Grained Cryptography

There has been much prior work leading up to the results presented in this thesis. First, there are a few results using assumptions from fine-grained complexity and applying them to cryptography. Second, there has been work with the kind of assumptions that we will be using.

Fine-Grained Cryptography

Ball et al. [BRSV17, BRSV18] produce fine-grained wost-case to average-case reductions. Ball et al. leave an open problem of producing a one-way-function from a worst case assumption. They prove that from some fine-grained assumptions building a one-way-function would falsify NSETH [CGI$^+$16][BRSV17]. We avoid their barrier in this paper by producing a construction of both fine-grained OWFs and fine-grained PKE from an *average-case* assumption.

Fine-Grained Key Exchanges. Fine-grained cryptography is a relatively unexplored area, even though it had its start in the 1970's with Merkle puzzles: the gap between honestly participating in the protocol versus breaking the security guarantee was only quadratic [Mer78]. Merkle originally did not describe a plausible hardness

assumption under which the security of the key exchange can be based. 30 years later, Biham, Goren, and Ishai showed how to implement Merkle puzzles by making an assumption of the existence of either a random oracle or an exponential gap one way function [BGI08]. That is, Merkle puzzles were built under the assumption that a one-way function exists which takes time $2^{n(1/2+\delta)}$ to invert for some $\delta > 0$. So while prior work indeed succeeded in building a fine-grained key-exchange, it needed a very strong variant of OWFs to exist. It is thus very interesting to obtain fine-grained public key encryption schemes based on a fine-grained assumption (that might even work in Pessiland and below).

Another notion of Fine-Grained Cryptography. In 2016, work by Degwekar, Vaikuntanathan, and Vasudevan [DVV16] discussed fine-grained complexity with respect to both honest parties and adversaries restricted to certain circuit classes. They obtained constructions for some cryptographic primitives (including PKE) when restricting an adversary to a certain circuit class. From the assumption $\mathsf{NC1} \neq \oplus L/\text{poly}$ they show Alice and Bob can be in $AC^0[2]$ while being secure against $\mathsf{NC1}$ adversaries. While [DVV16] obtains some unconditional constructions, their security relies on the circuit complexity of the adversary, and does not apply to arbitrary time-bounded adversaries as is usually the case in cryptography. That is, this restricts the types of algorithms an adversary is allowed to use beyond just how much runtime these algorithms can have. It would be interesting to get similar results in the low-polynomial time regime, without restricting an adversary to a certain circuit class. Our results achieve this, though not unconditionally.

Tight Security Reductions and Fine-Grained Crypto. Another area the world of fine-grained cryptography collides with is that of tight security reductions in cryptography. Bellare et.al. coined the term "concrete" security reductions in [BKR94, BGR95]. Concrete security reductions are parametrized by time (t), queries (q), size (s), and success probability (ϵ). This line of work tracks how a reduction from a problem to a construction of some cryptographic primitive effects the four parameters of interest. This started a rich field of study connecting theory to practical cryptographic primitives (such as PRFs, different instantiations of symmetric encryption, and even IBE for example [BCK96, BDJR97, KW03, BR09]). In fine-grained reductions we also need to track exactly how our adversary's advantage changes throughout our reductions, however, we also track the running time of the honest parties. So, unlike in the concrete security literature, when the hard problems are polynomially hard (perhaps because $P = NP$), we can track the gap in running times between the honest and dishonest parties. This allows us to build one way functions and public key cryptosystems when the hard problems we are given are only polynomially hard.

Similar Assumptions

This paper uses hypotheses on the running times of problems that, while solvable in polynomial time, are variants of natural NP-hard problems, in which the size of the solution is a fixed constant. For instance, k-Sum is the variant of Subset Sum, where we are given n numbers and we need to find exactly k elements that sum to a given

Paper	Assumptions	Crypto	Runtime	Power of Adversary
[Mer78]	Random Oracles*	Key Exchange	$O(N)$	$O(N^2)$
[BGI08]	Exponentially-Strong OWFs	Key Exchange	$O(N)$	$O(N^2)$
[BRSV18]	WC 3-Sum, OV, APSP, or SETH	Proof of Work	$O(N^2)$	N/A
[This thesis]/ [LLW19]	Zero-k-Clique or k-Sum	OWFs, Key Exch. & PKE	$O(N)$ $O(N)$	$O(N^{1+\delta})$ $O(N^{1.5-\delta})$
[DVV16]	$\mathsf{NC1} \neq \oplus L/\text{poly}$	OWFs, and PRGs with sublinear stretch, CRHFs, and PKE	$\mathsf{NC1}$	$\mathsf{NC1}$
	$\mathsf{NC1} \neq \oplus L/\text{poly}$	PKE and CRHFs	$\mathsf{AC}^0[2]$	$\mathsf{NC1}$
	Unconditional	PRGs with poly stretch, Symmetric encryption, and CRHFs	AC^0	AC^0

Figure 1-1: A table of previous works' results in this area. There have been several results characterizing different aspects of fine-grained cryptography. *It was [BGI08] who showed that Merkle's construction could be realized with a random oracle. However, Merkle presented the construction.

target, and Zero-k-Clique is the variant of Zero-Clique, in which we are given a graph and we need to find exactly k nodes that form a clique whose edge weights sum to zero.

With respect to Subset Sum, Impagliazzo and Naor showed how to directly obtain OWFs and PRGs assuming that Subset Sum is hard on average [IN02]. The OWF is $f(\mathbf{a}, \mathbf{s}) = (\mathbf{a}, \mathbf{a} \cdot \mathbf{s})$, where $\mathbf{a}$ is the list of elements (chosen uniformly at random from the range R) and $s \in \{0,1\}^n$ represents the set of elements we add together. In addition to Subset Sum, OWFs have also been constructed from planted Clique, SAT, and Learning-Parity with Noise [Lin17, JP00]. The constructions from the book of Lindell and the chapter written by Barak [Lin17] come from a definition of a "plantable" NP-hard problem that is assumed to be hard on average.

Although our OWFs are equivalent to scaled-down, polynomial-time solvable characterizations of these problems, we also formalize the property that allows us to get these fine-grained OWFs (plantability). We combine these NP constructions and formalizations to lay the groundwork for fine-grained cryptography.

In the public-key setting, there has been relatively recent work taking NP-hard problems and directly constructing public-key cryptosystems [ABW10]. They take a problem that is NP-hard in its worst case and come up with an average-case assumption that works well for their constructions. Our approach is similar, and we also provide evidence for why our assumptions are correct.

In recent work, Subset Sum was also shown to directly imply public-key cryptography [LPS10]. The construction takes ideas from Regev's LWE construction [Reg05], turning a vector of subset sum elements into a matrix by writing each element out base q in a column. The subset is still represented by a 0-1 matrix, and error is handled by the lack of carrying digits. It is not clear how to directly translate this construction into the fine-grained world. First, directly converting from Subset Sum to k-Sum just significantly weakens the security without added benefit. More importantly, the security reduction has significant polynomial overhead, and would not apply in a very pessimistic Pessiland where random planted Subset Sum instances can be solved in quadratic time, say.

While it would be interesting to reanalyze the time-complexity of this construction (and others) in a fine-grained way, this is not the focus of our work. Our goal is to obtain novel cryptographic approaches exploiting the fine-grained nature of the problems, going beyond just recasting normal cryptography in the fine-grained world, and obtaining somewhat generic constructions.

1.3 Outline of Thesis

Because this thesis spans multiple subjects, each chapter is self-contained. In Chapter 2, we present two results in the area of Topology-Hiding Computation (THC): achieving THC from the Learning-With Errors assumption in Section 2.3, and showing that Topology-Hiding Broadcast is impossible if we allow an adversary to control message timings in Section 2.4. Then, in Chapter 3, we introduce the notion of a robust Property Preserving Hash (PPH), defining our notions in Section 3.2. We

discuss impossibility results for which properties cannot be hashed and preserved in Section 3.3, describe constructions for the gap-hamming property in Sections 3.4 and 3.5, and demonstrate that making cryptographic assumptions is necessary for building PPHs for many kinds of properties. Finally, in Chapter4, we define and construct multiple fine-grained cryptographic primitives. First, we define our notions of fine-grained One-Way Functions, key-exchanges, and public-key cryptosystems in Section 4.2. We construct fine-grained One-Way Functions in Section 4.6. Second, we construct fine-grained key-exchanges and fine-grained public key cryptosystems in two ways: from general properties of a fine-grained assumption in Section 4.7, and from assuming the strong Zero-k-Clique-R Hypothesis in Section 4.8.