**Format:**  
Your report should have a minimum length of 15 pages LNCS Style, excluding references.   
Your report has to include references, but they do not count towards your total page limit. The 'literature list' is the list of references.   
  
Try not to write more than 20 pages unless you feel you have a very good reason.   
  
**Template:**  
The LNCS is template is available for Latex [1] and Word [2] on the Springer page,   
but I suggest you use this quick start version [3].   
  
**How to correctly include references in your literature list:**  
  
Please refer to e.g. the Chicago manual of style [5].  
  
Any valid reference style is accepted, as there are many styles, and as they basically contain the same information.   
  
  
  
**Next steps:**  
I will send you instructions about the peer review process closer to the next deadline.  
  
  
  
Best regards, Benjamin Heitmann.  
  
  
  
[1] <ftp://ftp.springer.de/pub/tex/latex/llncs/latex2e/llncs2e.zip>  
[2] <ftp://ftp.springer.de/pub/tex/latex/llncs/word/splnproc1110.zip>  
[3] <https://github.com/latextemplates/LNCS>  
[4] <http://www.chicagomanualofstyle.org/tools_citationguide.html>

TO DO:

* Use Latex
* Find examples/ empirical results
* Gmw protocol – scapi library LSSR based, passive as comparison
* Reusing results: maybe check in night for adversaries?
* Could be nice to include reuse

1. **Motivation & Introduction:** SMPC, SCALE-MAMBA
2. **Cryptography Basics:** OT, SHE, MAC
3. **SMPC basics & history:** Shamir secret sharing, SPDZ
4. **SCALE-MAMBA**
   * **Architecture:** SCALE & MAMBA
   * **Logic:** Online & Offline Phase
   * **Math:** What is actually happening?
5. **Comparison:** Performance, Security, Use Cases
6. **Outlook:** What brings the future?
7. **Resume**
8. Motivation & introduction

1. Motivation:

* There are a lot of computations that have to be done in a distributed environment in todays technologies and in the future
* Not possible to have all the information in one place anymore
* Sometimes hard to know who to trust
* Problem: want to compute something together but we don’t want to share our information
* A lot of use cases: number of distributed environments is climbing rapidly; for example see crypto currencies (distributed block chain; no one can do something alone, everyone relies on everyone else (or at least on a majority of the other participants)
* A lot of research. Right now more theoretical than practical works; a lot of computational overhead
* Scale-Mamba:
  + Developed by a research team of the KU Leuven
  + Implements a lot of the theoretical foundations
  + Crypto suit, end to end
  + Possible to be safe against corruption with abort
  + Good (compared to others) execution times
  + Improvement of before existing technologies (SPDZ especially, and others)

1. Structure:

This thesis will try to give an overview over Secure Multiparty Computation, especially focusing on Scale-Mamba and all the technologies it builds upon. Therefore, it will explain the basic technologies first in an abstracted manner and later go more into detail.

1. Cryptographic fundaments
2. Message authentication code (MAC)
3. Oblivious transfer (OT)
4. Homomorphic Encryption (HE)
5. Somewhat Homomorphic Encryption (SHE)
6. Access structures?
7. Monotone Span Programs (MSP) ?
8. FHE – fully homomorphic encryption?
9. Different adversary types
   1. Malicious
   2. Honest but corious

* Maybe move this behind the basic introduction of SMPC? Some parts like access structures

1. SMPC basics & history
2. Introduction (History, Yao Garbled Circuit, Shamir Secret Sharing Scheme)
   1. Part of cryptography that was started in the 1970s and is getting much more relevance in the last years
   2. YAOs Millionaire problem
   3. Mental poker
   4. Yaos garbled circuit
   5. Shamirs secret sharing scheme
      1. Choose random polynomial out of finite field to a prime
      2. Degree of polynomial t is the number of tolerated adversaries
      3. Polynomial: f(0) = s
      4. Send shares to each other player
      5. T+1 shares needed to reconstruct secret (with Lagrange interpolation in O(n^2))
      6. These can be used for different operations; for example some calculations can be made on the secrets, and at the end the final result is revealed and calculated and compared; so a result can be calculated without revealing the own secret
   6. Other secret sharing schemes
      1. Replicated (KNF based)
      2. Dx`NF based
3. SMPC basics

* Access structures
* Montone span programs?
* Offline/online phase
* Secret Sharing foundations (mathematical)

1. Early adoptions of LSSR; MASCOT?
2. SPDZ

* Maybe not too many details, because a lot of things are similar to SCALE-MAMBA?

1. Scale-Mamba
2. Why we need it? Problems of SPDZ, other SMPC protocols
3. Architecure

* Written in C++
* Scale: Secure Computation Algorithms from LEuven
  + SMPC foundations
  + Code base based on different projects like SPDZ, HE…?,.
  + Combination of offline & online phase
  + We will focus more on this then on the Compiler part, its more relevant for this thesis
* Mamba: Multiparty AlgorithMs Basic Argot
  + Compiler; similar to python
  + Why necessary? Easily able to write own configurations for SMCP for SCALE-MAMBA

1. Idea, Computational Logic

* Online & Offline phase
* Offline phase:
  + Calculation of beaver triples
  + Precalculations like MAC?
* Online phase:
  + Secret sharing
  + Sacrifice of triples to check calculation results
  + Check for adversaries (MAC?)
  + Abortion
* Options/ Configurations:
  + It is possible to use different setups, depending on the taste of the user

1. Performance, real world usage, problems, upcoming developments,

* Upcoming: Calculating offline data before, and then compiling the Mamba code just in time (especially full threshold access structures need a lot of time)

1. Comparison
   1. What else is out there?
   2. Different use cases – when which SMPC tool is more useful; what are the strengths of SCALE-MAMBA compared to other ones?
   3. Factual comparisons
2. Outlook
   1. What brings the future in general in SMPC?
   2. SMPC is a technology that will shape the future (at least a bit)
3. Resume
   1. SCALE-MAMBA is amazing
   2. We saw how it works in which cases it is more useful
   3. We saw why SMPC is necessary; we saw that we need it (or at least we should use it)

List of abbrviations

LSSR

SMPC

SPDZ

OT

MAC

HE

SHE

SCALE

MAMBA

List of references