6 Conclusion

In this thesis we considered tight reductions in cryptography. Chapter 3 dealt with properties of an implementation of a cryptographic primitive that rule out a tight reduction. We considered security experiments that induce some relation over statements and witnesses and that may be broken by the adversary if it is able to compute a “fresh” witness. To rule out a tight security proof we required the relation to be efficiently computable during the security experiment and witnesses to be efficiently re-randomizable. An interesting open problem is whether these conditions are also necessary to rule out tight reductions.

We have discussed how known implementations of primitives that come along with an (almost) tight security proof circumvent our result. Here, we focused on the re-randomization property. However, it may also be possible to circumvent our result when the security experiment of the considered primitive does not allow efficient verification, e.g., when the considered primitive is a pseudorandom function or a message authentication code. For PRFs a possibility result is the classic Naor-Rheingold PRF [NR97]. Further research could analyze this property in the light of our new work. Moreover, it would be interesting to apply our result to further primitives.

In Chapters 4 and 5 we showed how to construct signature schemes that are tightly secure in the multi user setting with respect to both, sign-queries and number of users. We proposed a tightness preserving compiler that is provably secure in the standard model, as well as a quite efficient solution in the ROM. While the ROM solution supports signatures, public keys and parameters of constant size, it is currently not clear if this can be achieved using our standard model compiler. Namely, one drawback of our compiler is that, when it is implemented with signature schemes that are tightly secure in the multi user setting without corruptions [HJ12, CW13] we either have large pub-
lic parameters or large signatures. An interesting open problem is to construct a signature scheme that is tightly secure in the multi user setting with corruptions from standard assumptions and that supports constant size signatures, public keys and parameters and that is provably secure in the standard model.