Appendix A
Assumptions

Definition A.1 (Bilinear Map [4]) A *bilinear group* is a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ such that:

- $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ are cyclic groups of prime order p.
- $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is bilinear, i.e. for all $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}$, $e({g_1}^a, {g_2}^b) = e(g_1, g_2)^{ab}$.
- e is an *admissible* bilinear map, i.e.
  - e is efficiently computable;
  - if g_1 and g_2 are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, i.e. $\langle g_1 \rangle = \mathbb{G}_1$ and $\langle g_2 \rangle = \mathbb{G}_2$, then $\mathbb{G}_T$ is generated by $e(g_1, g_2)$
- the *Discrete Logarithm Problem* is hard to be computed in $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$.

The function e is called *bilinear map*, or *pairing*.

Bilinear maps can be divided into symmetric maps where $\mathbb{G} = \mathbb{G}_1 = \mathbb{G}_2$ and asymmetric maps where an isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ is not efficiently computable. There are different hardness assumptions that can be based on bilinear maps.

Assumption A.1 (Decision Linear [1]) Let $(p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda)$ output a cyclic bilinear group of order p with bilinear map
$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.
The decision linear assumption holds if for randomly chosen $g_b, g_1, g_2 \in \mathbb{G}$ and $r_0, r_1, r_2 \in \mathbb{Z}_p$

$$|\Pr\left[A((p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda), g_0, g_1, g_2, g_1^{r_1}, g_2^{r_2}, g_0^{r_1+r_2})\right] -$$
$$\Pr\left[A((p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda), g_0, g_1, g_2, g_1^{r_1}, g_2^{r_2}, g_0^{r_0})\right]| = \mathsf{negl}(\lambda).$$

D. Demirel et al., *Privately and Publicly Verifiable Computing Techniques*,
SpringerBriefs in Computer Science, DOI 10.1007/978-3-319-53798-6

Assumption A.2 (q-Strong Diffie-Hellman Inversion [2]) Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda)$ output an asymmetric bilinear map with random generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ and let $q = poly(\lambda)$. The q-Strong Diffie-Hellman Inversion (q-DHI) holds, if for every PPT adversary A

$$\Pr\left[A(g_1, g_1^z, g_2^z, \ldots, g_1^{z^q}, g_2^{z^q}) = g_1^{\frac{1}{z}} \;\middle|\; z \xleftarrow{\$} \mathbb{Z}_p\right] = \mathsf{negl}(\lambda).$$

Assumption A.3 (External Decisional Diffie-Hellman in $\mathbb{G}_1$ [7]) Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda)$ output an asymmetric bilinear map with random generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. The External Decisional Diffie-Hellman Assumption (XDDH) holds in $\mathbb{G}_1$, if for every PPT adversary A

$$\begin{aligned}&|\Pr\left[A(g_1, g_2, g_1^a, g_1^b, g_1^{ab}) = 1 \;\middle|\; a, b \xleftarrow{\$} \mathbb{Z}_p\right]\\ &- \Pr\left[A(g_1, g_2, g_1^a, g_1^b, g_1^c) = 1 \;\middle|\; a, b, c \xleftarrow{\$} \mathbb{Z}_p\right]| = \mathsf{negl}(\lambda).\end{aligned}$$

Assumption A.4 (Flexible Diffie-Hellman Inversion [7]) Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda)$ output an asymmetric bilinear map with random generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. The Flexible Diffie-Hellman Inversion Assumption (FDHI) holds, if for every PPT adversary A

$$\Pr\left[W \in \mathbb{G}_1 \backslash \{1\}, W' = W^{\frac{1}{z}} : (W, W') \leftarrow A(g_1, g_2, g_2^z, g_2^v, g_1^{\frac{z}{v}}, g_1^r, g_1^{\frac{r}{v}}) \;\middle|\; z, v, r \xleftarrow{\$} \mathbb{Z}_p\right]$$
$$= \mathsf{negl}(\lambda).$$

Assumption A.5 (Co-computational Diffie-Hellman Assumption [8]) Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda)$ output an asymmetric bilinear map with random generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$. The co-computational Diffie-Hellman assumption (co-CDH) holds if for all PPT adversaries A

$$\Pr\left[g_1^{ab} \leftarrow A(g_1, g_1^a g_2, g_2^b) \;\middle|\; a, b \leftarrow \mathbb{Z}_p\right] = \mathsf{negl}(\lambda)$$

Assumption A.6 (q-PDH [9]) The q-power Diffie-Hellman (q-PDH) assumption holds if for all non-uniform PPT adversaries A

$$\Pr\left[y = g^{s^{q+1}} \;\middle|\; \begin{array}{c}(p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda)\\ g \leftarrow \mathbb{G}\backslash\{1\}\\ s \leftarrow \mathbb{Z}_p^*\\ \sigma \leftarrow (p, \mathbb{G}, \mathbb{G}_T, e, g, g^s, \ldots, g^{s^q}, g^{s^{q+2}}, g^{s^{2q}})\\ y \leftarrow A(\sigma)\end{array}\right] = \mathsf{negl}(\lambda).$$

Assumption A.7 (q**-PKE [9])** The q power of knowledge assumptions holds if for all adversaries A there exists a non-uniform PPT extractor χ_A such that

$$\Pr\left[\begin{array}{c} \hat{c} = c^a \\ \wedge\, c \neq \prod_{i=0}^{q} g^{a_i s^i} \end{array} \middle| \begin{array}{c} (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda) \\ g \leftarrow \mathbb{G}\backslash\{1\} \\ a, s \leftarrow \mathbb{Z}_p^* \\ \sigma \leftarrow (p, \mathbb{G}, \mathbb{G}_T, e, g, g^s, \ldots, g^{s^q}, g^a, g^{a\cdot s}, \ldots, g^{a\cdot s^q}) \\ (c, \hat{c}, a_0, \ldots, a_q) \leftarrow (A||\chi_A)(\sigma, z) \end{array}\right]$$

$$= \mathsf{negl}(\lambda)$$

for any auxiliary information $z \in \{0,1\}^{poly(\lambda)}$ that is generated independently of a.

Note that $(y, z) \leftarrow (A||\chi_A)(x)$ signifies that on output x adversary A outputs y and that χ_A, given the same input x and A's random tape, produces z.

Assumption A.8 (q**-SDH [3])** The q strong Diffie-Hellman (q-SDH) holds if for all ppt adversaries A

$$\Pr\left[y = e(g,g)^{\frac{1}{s+c}}, c \in \mathbb{Z}_p^* \middle| \begin{array}{c} (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow Gen(1^\lambda) \\ g \leftarrow \mathbb{G}\backslash\{1\} \\ s \leftarrow \mathbb{Z}_p^* \\ \sigma \leftarrow (p, \mathbb{G}, \mathbb{G}_T, e, g, g^s, \ldots, g^{s^q}) \\ y \leftarrow A(\sigma) \end{array}\right] = \mathsf{negl}(\lambda).$$

For many purposes just having bilinear maps is not enough. Thus, the following is a useful generalization.

Definition A.2 (Multilinear Map [5]) A k *-linear map* is a tuple $pp = (p, \mathbb{G}_1, \ldots, \mathbb{G}_k, e_{ij} : 1 \leq, i, j \leq k)$ such that:

- $e_{ij} : \mathbb{G}_i \times \mathbb{G}_j \rightarrow \mathbb{G}_{i+j}$
- each $(p, \mathbb{G}_i, \mathbb{G}_j, \mathbb{G}_{i+j}, e_{ij})$ is a bilinear map.

There are also several types of assumptions that can be based on multilinear maps.

Assumption A.9 ((k, l) **Multilinear Diffie Hellman Inversion, [5])** Let pp (see Definition A.2) be the description of a set of multilinear groups and $g_1 \in \mathbb{G}_1$ a random generator. Let $w \in \mathbb{Z}_p$ be random. The (k, l) Multilinear Diffie Hellman Inversion holds, if for all adversaries A

$$\Pr\left[A(g_1, g_1^w, \ldots, g_1^{w^l}) = g_k^{w^{kl+1}}\right] = \mathsf{negl}(\lambda).$$

Assumption A.10 (k **Augmented Power Multilinear Diffie Hellman Assumption [6])** Let pp (see Definition A.2) be the description of a set of multilinear groups, and $g_1 \in \mathbb{G}_1$ a random generator. Let $a, b, x \in \mathbb{Z}_p$ be random. The k Augmented

Power Multilinear Diffie Hellman Assumption holds if for all adversaries A

$$\Pr\left[A(g_1, g_1^a, g_1^b, g_1^{ab}, g_1^x, g_1^{ax}, g_1^{abx}) = g_k^{a^{k-1}(bx)^k}\right] = \mathsf{negl}(\lambda).$$

Other well known hardness assumptions are *factorization based*, i.e. given $N = pq$ the product of two distinct odd primes it should be difficult to find p and q. N is called an RSA modulus.

Assumption A.11 (RSA Assumption) Let (N, e) be a pair of integers, where $N = pq$ for some odd prime numbers p, q, $e \in \mathbb{Z}_N \backslash \{1\}$ and $\gcd(e, \varphi(N)) = 1$. Given an element $z \in \mathbb{Z}_N$, the standard RSA problem is to compute the integer y such that $y^e \equiv z \mod N$.

There is also a stronger variant of this where the attacker can choose e.

Assumption A.12 (Strong RSA Assumption) Let (N, e) be a pair of integers, where $N = pq$ for some odd prime numbers p, q, $e \in \mathbb{Z}_N \backslash \{1\}$ and $\gcd(e, \varphi(N)) = 1$. Given an element $z \in \mathbb{Z}_N$ and the freedom to choose e, the *Strong RSA* problem is to compute the integer y such that $y^e \equiv z \mod N$.

References

1. M. Backes, D. Fiore, R.M. Reischuk, Verifiable delegation of computation on outsourced data, in *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13*, Berlin, 4–8 November 2013, pp. 863–874
2. D. Boneh, X. Boyen, Efficient selective-ID secure identity-based encryption without random Oracles, in *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*, Interlaken, 2–6 May 2004, pp. 223–238
3. D. Boneh, X. Boyen, Short signatures without random Oracles, in *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*, Interlaken, 2–6 May 2004, pp. 56–73
4. D. Boneh, M.K. Franklin, Identity-based encryption from the Weil pairing. SIAM J. Comput. **32**, 586–615 (2003)
5. D. Catalano, D. Fiore, R. Gennaro, L. Nizzardo, Generalizing homomorphic MACs for arithmetic circuits, in *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Proceedings*, Buenos Aires, 26–28 March 2014, pp. 538–555
6. D. Catalano, D. Fiore, B. Warinschi, Homomorphic signatures with efficient verification for polynomial functions, in *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Proceedings, Part I*, Santa Barbara, CA, 17–21 August 2014, pp. 371–389
7. D. Catalano, D. Fiore, L. Nizzardo, Programmable hash functions go private: constructions and applications to (homomorphic) signatures with shorter public keys, in *Advances in Cryptology - CRYPTO 2015 - 35th Annual Cryptology Conference, Proceedings, Part II*, Santa Barbara, CA, 16–20 August 2015, pp. 254–274
8. K. Elkhiyaoui, M. Önen, M. Azraoui, R. Molva, Efficient techniques for publicly verifiable delegation of computation, in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2016*, Xi'an, 30 May–3 June 2016, pp. 119–128

9. B. Parno, J. Howell, C. Gentry, M. Raykova, Pinocchio: nearly practical verifiable computation, in *2013 IEEE Symposium on Security and Privacy, SP 2013*, Berkeley, CA, 19–22 May 2013, pp. 238–252