CS 395T: Topics in Cryptography **Spring 2024**

Exercise Set 2

Due: February 9, 2024 at 11:59pm (Submit on Gradescope) **Instructor:** David Wu

Instructions. You **must** typeset your solution in LaTeX using the provided template:

https://www.cs.utexas.edu/~dwu4/courses/sp24/static/homework.tex

You must submit your problem set via Gradescope (accessible through Canvas).

Collaboration Policy. You may discuss your general approach with other students, but you may not share written documents. You should not search online for solutions to these problems. If you do consult external sources, you must cite them in your submission. You must include the names of all of your collaborators with your submission. Refer to the official course policies for the full details.

Problem 1: Computing on Secret-Shared Inputs [12 points]. In this problem, we will explore how to perform computations on secret-shared inputs. For integers $t \leq N \ll p$, we will say that $(s_1, \ldots, s_N)$ is a t-out-of-N secret sharing of a value $s \in \mathbb{F}_p$ if there exists a polynomial $f : \mathbb{F}_p \to \mathbb{F}_p$ of degree at most $t-1$ where $f(0) = s$ and for all $i \in [N]$, $f(i) = s_i$.

Suppose $t < N/2$. Let $(u_1, \ldots, u_N)$ be a t-out-of-N secret sharing of $u \in \mathbb{F}_p$ and $(v_1, \ldots, v_N)$ be a t-out-of-N secret sharing of $v \in \mathbb{F}_p$. First, observe the following properties (verify these yourselves if you are not convinced):

- **Addition:** $(u_1 + v_1, \ldots, u_N + v_N)$ is a t-out-of-N secret sharing of $u + v$
- **Scalar multiplication:** For every constant $c \in \mathbb{F}_p$, $(cu_1, \ldots, cu_N)$ is a t-out-of-N secret sharing of cu.
- **Multiplication:** $(u_1 v_1, \ldots, u_N v_N)$ is a $(2t-1)$-out-of-N secret sharing of the product uv.

Suppose $(w_1, \ldots, w_N)$ is a $(2t-1)$-out-of-N secret sharing of a secret w, and suppose party i holds w_i. Your goal is to design a secure *degree-reduction* protocol that outputs a t-out-of-N secret sharing $(w'_1, \ldots, w'_N)$ of w. Your protocol should satisfy the following properties:

- Each party i sends a single (secret) value $x_{i,j} \in \mathbb{F}_p$ to party j. The value $x_{i,j}$ is a function of the sender's share w_i and the recipient's index j.
- Using the incoming messages $x_{i,j}$ for all $i \neq j$ and its share w_j, party j computes a new share w'_j.
- The shares $(w'_1, \ldots, w'_N)$ are a t-out-of-N secret sharing of w.
- The protocol *perfectly* hides the value of w even if $t-1$ users collude (i.e., the $t-1$ colluding parties share the set of messages they received from the other parties together with their individual shares of w). You can assume that all parties follow your protocol as written (i.e., they are "honest-but-curious.")

Show that your protocol is correct and explain (briefly) why it is secure against $(t-1)$ colluding parties. **Note:** Depending on your construction, the security proof may immediately follow from (perfect) security of Shamir secret sharing. If this is the case for your protocol, then you simply need to state this.

Remark: The protocol you have developed here can be used to multiply secret-shared inputs (namely: each party locally multiplies their shares and then applies the degree-reduction protocol you developed). This can be used as the basis of a protocol for computing *arbitrary* functions on secret-shared data.

Problem 2: IBE and Digital Signatures [12 points]. In class, we saw many similarities between the Boneh-Franklin IBE scheme and the Boneh-Lynn-Shacham signature scheme. It turns out that there is a more general implication between IBE and digital signatures. Let $\Pi_{\mathsf{IBE}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be an *arbitrary* secure identity-based encryption scheme with n-bit identities and 1-bit messages. Show how to use Π_{IBE} to construct a secure digital signature scheme on n-bit messages. Show that your scheme is correct and provide an *informal* sketch for the security of your scheme. The formal proof of security can be a little tedious, so a high-level sketch suffices here. Your sketch should include the following details: (1) how the reduction algorithm constructs the verification key and answers the signature adversary's signing queries; and (2) why a successful forgery by the signing adversary breaks security of the IBE scheme. Your explanation for (2) can be informal and does *not* need a formal calculation.

Optional Feedback. Please answer the following *optional* questions to help design future exercise sets. You do not need to answer these questions. However, we do encourage you to provide us feedback on how to improve the course experience.

(a) How long did you spend on this exercise set?

(b) Do you have any feedback for this exercise set?

(c) Do you have any feedback on the course so far?

(d) Are there specific topics that you are interested in seeing in this course?

Challenge Problem: Attacks on Power Diffie-Hellman [Optional]. Let $\mathbb{G}$ be a group of prime order p with generator g. Normally, a (generic) discrete log algorithm in $\mathbb{G}$ would require time $\tilde{O}(\sqrt{p})$. Suppose instead that in addition to the discrete log challenge (g, g^α), the adversary is also given $\left(g^{\alpha^2}, g^{\alpha^3}, \ldots, g^{\alpha^N}\right)$. Suppose there exists $d \leq N$ such that $d \mid p-1$. Give an algorithm that recovers α in time $\tilde{O}\left(\sqrt{p/d} + \sqrt{d}\right)$. This problem shows that the hardness of problems like the N-bilinear Diffie-Hellman exponent assumption generally degrades with the assumption size N. This attack is also relevant when considering the security of oblivious PRFs such as the DDH-based construction we described in class: $\mathsf{PRF}(k, x) := H(x)^k$.

Challenge Problem: Rebalancing Broadcast Encryption [Optional]. In the Boneh-Gentry-Waters broadcast encryption scheme we described in class, the public key contained $O(N)$ group elements while the secret keys and the ciphertexts contained $O(1)$ group elements, where N is the number of users. Show how to modify the construction to obtain a broadcast encryption scheme where the public key and the ciphertext contains $O(\sqrt{N})$ group elements and the secret keys still contain $O(1)$ group elements. Your construction shows how to reduce the length of the public key at the expense of longer ciphertexts in the context of broadcast encryption.