Smooth Projective Hashing and Two-Message Oblivious Transfer

Yael Tauman Kalai

Massachusetts Institute of Technology*
tauman@mit.edu,
http://www.mit.edu/∼tauman

Abstract. We present a general framework for constructing two-message oblivious transfer protocols using a modification of Cramer and Shoup's notion of smooth projective hashing (2002). Our framework is actually an abstraction of the two-message oblivious transfer protocols of Naor and Pinkas (2001) and Aiello et al. (2001), whose security is based on the Decisional Diffie Hellman Assumption. In particular, we give two new oblivious transfer protocols. The security of one is based on the N'th-Residuosity Assumption, and the security of the other is based on both the Quadratic Residuosity Assumption and the Extended Riemann Hypothesis. Our security guarantees are not simulation based, and are similar to those of previous constructions.

When using smooth projective hashing in this context, we must deal with maliciously chosen smooth projective hash families. This raises new technical difficulties, and in particular it is here that the Extended Riemann Hypothesis comes into play.

1 Introduction

In [CS98], Cramer and Shoup introduced the first CCA2 secure encryption scheme, whose security is based on the Decisional Diffie Hellman (DDH) Assumption. They later presented an abstraction of this scheme based on a new notion which they called "smooth projective hashing" [CS02]. This abstraction yielded new CCA2 secure encryption schemes whose security is based on the Quadratic Residuosity Assumption or on the N'th Residuosity Assumption [Pa99].[1] This notion of smooth projective hashing was then used by Genarro and Lindell [GL03] in the context of key generation from humanly memorizable passwords. Analogously, their work generalizes an earlier protocol for this problem [KOY01], whose security is also based on the DDH Assumption.

In this paper, we use smooth projective hashing to construct efficient two-message oblivious transfer protocols. Our work follows the above pattern, in that it generalizes earlier protocols for this problem [NP01,AIR01] whose security is

* Supported in part by NSF CyberTrust grant CNS-0430450.

[1] The N'th Residuosity Assumption is also referred to in the literature as the Decisional Composite Residuosity Assumption and as Paillier's Assumption.

based on the DDH assumption. Interestingly, using smooth projective hashing in this context raises a new issue. Specifically, we must deal with maliciously chosen smooth projective hash families. This issue did not arise in the previous two applications because these were either in the public key model or in the common reference string model.

1.1 Oblivious Transfer

Oblivious transfer is a protocol between a *sender*, holding two strings γ_0 and γ_1, and a *receiver* holding a choice bit b. At the end of the protocol the receiver should learn the string of his choice (i.e., γ_b) but learn nothing about the other string. The sender, on the other hand, should learn nothing about the receiver's choice b.

Oblivious transfer, first introduced by Rabin [Rab81], is a central primitive in modern cryptography. It serves as the basis of a wide range of cryptographic tasks. Most notably, any secure multi-party computation can be based on a secure oblivious transfer protocol [Y86,GMW87,Kil88]. Oblivious transfer has been studied in several variants, all of which have been shown to be equivalent. The variant considered in this paper is the one by Even, Goldreich and Lempel [EGL85] (a.k.a. 1-out-of-2 oblivious transfer), shown to be equivalent to Rabin's original definition by Crépeau [Cre87].

The study of oblivious transfer has been motivated by both theoretical and practical considerations. On the theoretical side, much work has been devoted to the understanding of the hardness assumptions required to guarantee oblivious transfer. In this context, it is important to note that known constructions for oblivious transfer are based on relatively strong computational assumptions – either specific assumptions such as factoring or Diffie Hellman (cf. [Rab81,BM89,NP01,AIR01]) or generic assumption such as the existence of enhanced trapdoor permutations (cf. [EGL85,Gol04,Hai04]). Unfortunately, oblivious transfer cannot be reduced in a black box manner to presumably weaker primitives such as one-way functions [IR89]. On the practical side, research has been motivated by the fact oblivious transfer is considered to be the main bottleneck with respect to the amount of computation required by secure multiparty protocols. This makes the construction of efficient protocols for oblivious transfer a well-motivated task.

In particular, constructing round-efficient oblivious transfer protocols is an important task. Indeed, [NP01] (in Protocol 4.1) and [AIR01] independently constructed a *two-message* (1-round) oblivious transfer protocol based on the DDH Assumption (with weaker security guarantees than the simulation based security). Their work was the starting point of our work.

1.2 Smooth Projective Hashing

Smooth projective hashing is a beautiful notion introduced by Cramer and Shoup [CS02]. To define this notion they rely on the existence of a set X (actually a distribution on sets), and an underlying $\mathcal{NP}$-language $L \subseteq X$ (with an associated $\mathcal{NP}$-relation R). The basic hardness assumption is that it is infeasible to distinguish between a random element in L and a random element in $X \setminus L$. This is called a *hard subset membership problem*.

A *smooth projective hash family* is a family of hash functions that operate on the set X. Each function in the family has two keys associated with it: a hash key k, and a projection key $\alpha(k)$. The first requirement (which is the standard requirement of a hash family) is that given a hash key k and an element x in the domain X, one can compute $H_k(x)$. There are two additional requirements: the "projection requirement" and the "smoothness requirement."

The "projection requirement" is that given a projection key $\alpha(k)$ and an element in $x \in L$, the value of $H_k(x)$ is uniquely determined. Moreover, computing $H_k(x)$ can be done efficiently, given the projection key $\alpha(k)$ and a pair $(x, w) \in R$. The "smoothness requirement," on the other hand, is that given a random projection key $s = \alpha(k)$ and any element in $x \in X \setminus L$, the value $H_k(x)$ is statistically indistinguishable from random.

1.3 Our results

We present a methodology for constructing a two-message oblivious transfer protocol from any (modification of a) smooth projective hash family. In particular, we show how the previously known (DDH based) protocols of [NP01,AIR01] can be viewed as a special case of this methodology. Moreover, we show that this methodology gives rise to two new oblivious transfer protocols; one based on the N'th Residuosity Assumption, and the other based on the Quadratic Residuosity Assumption along with the Extended Riemann Hypothesis.

Our protocols, similarly to the protocols of [NP01,AIR01], are not known to be secure according to the traditional simulation based definition. Yet, they have the advantage of providing a certain level of security even against malicious adversaries without having to compromise on efficiency (see Section 3 for further discussion on the guaranteed level of security).

The basic idea. Given a smooth projective hash family for a hard subset membership problem (which generates pairs X, L according to some distribution), consider the following two-message protocol for *semi-honest* oblivious transfer. Recall that the sender's input is a pair of strings γ_0, γ_1 and the receiver's input is a choice bit b.

$R \to S$: Choose a pair X, L (with an associated NP-relation R_L) according to the specified distribution. Randomly generate a triplet (x_0, x_1, w_b) where $x_b \in_R L$, $(x_b, w_b) \in R_L$, and $x_{1-b} \in_R X \setminus L$. Send (X, x_0, x_1).

$S \to R$: Choose independently two random keys k_0, k_1 for $\mathcal{H}$ and send $\alpha(k_0)$ and $\alpha(k_1)$ along with $y_0 = \gamma_0 \oplus H_{k_0}(x_0)$ and $y_1 = \gamma_1 \oplus H_{k_1}(x_1)$.

R: Retrieve γ_b by computing $y_b \oplus H_{k_b}(x_b)$, using the witness w_b and the projection key $\alpha(k_b)$.

The security of the receiver is implied by the hardness of the subset membership problem on X. Specifically, guessing the value of b is equivalent to distinguishing between a random element in L and a random element in $X \setminus L$. The security of the sender is implied by the smoothness property of the hash family $\mathcal{H}$. Specifically, given a random projection key $\alpha(k)$ and any element in $x \in X \setminus L$, the value $H_k(x)$ is statistically indistinguishable from random. Thus, the message y_{1-b} gives no information about γ_{1-b} (since $x_{1-b} \in X \setminus L$). Note that the functionality of the protocol is implied by the projection property.

Technical difficulty. Notice that when considering malicious receivers, the security of the sender is no longer guaranteed. The reason is that there is no guarantee that the receiver will choose $x_{1-b} \in X \setminus L$. A malicious receiver might choose $x_0, x_1 \in L$ and learn both values. To overcome this problem, we extend the notion of a hard subset membership problem so that it is possible to verify that at least one of x_0, x_1 belongs to $X \setminus L$. This should work even if the set X is maliciously chosen by the receiver.

It turns out that implementing this extended notion in the context of the DDH assumption is straightforward [NP01,AIR01]. Loosely speaking, in this case X is generated by choosing a random prime p, and choosing two random elements g_0, g_1 in $\mathbb{Z}_p^*$ of some prime order q. The resulting set X is defined by $X \triangleq \{(g_0^{r_0}, g_1^{r_1}) : r_0, r_1 \in \mathbb{Z}_q\}$, the corresponding language L is defined by $L \triangleq \{(g_0^r, g_1^r) : r \in \mathbb{Z}_q\}$, and the witness of each element $(g_0^r, g_1^r) \in L$ is its discrete logarithm r. In order to enable the sender to verify that two elements x_0, x_1 are not both in L, we instruct the receiver to generate x_0, x_1 by choosing at random two distinct elements $r_0, r_1 \in \mathbb{Z}_q$, setting $x_b = (g_0^{r_0}, g_1^{r_0})$, $w_b = r_0$, and $x_{1-b} = (g_0^{r_0}, g_1^{r_1})$. Notice that x_b is uniformly distributed in L, x_{1-b} is uniformly distributed in $X \setminus L$, and the sender can easily check that it is not the case that both x_0 and x_1 are in L by merely checking that they agree on their first coordinate and differ on their second coordinate.

Implementing this verifiability property in the context of the N'th Residuosity Assumption and the Quadratic Residuosity Assumption is not as easy. This part contains the bulk of technical difficulties of this work. In particular, this is where the Extended Riemann Hypothesis comes into play in the context of Quadratic Residuosity.

2 Smooth Projective Hash Functions

Our definition of smooth projective hashing differs from its original definition in [CS02]. The main difference (from both [CS02] and [GL03]) is in the definition of the smoothness requirement, which we relax to Y*-smoothness*, and in the definition of a subset membership problem, where we incorporate an additional requirement called Y*-verifiability*.

Notation. The security parameter is denoted by n. For a distribution $\mathcal{D}$, $x \leftarrow \mathcal{D}$ denotes the action of choosing x according to $\mathcal{D}$, and $x \in$ *support*(D) means that the distribution $\mathcal{D}$ samples the value x with positive probability. We denote by $x \in_R S$ the action of uniformly choosing an element from the set S. For any two random variables X, Y, we say that X and Y are ϵ-close if $Dist(X, Y) \leq \epsilon$, where $Dist(X, Y)$ denotes the statistical difference between X and Y.[2] We say that the ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are statistically indistinguishable if there exists a negligible function $\epsilon(\cdot)$ such that for every $n \in \mathbb{N}$, the random variables X_n and Y_n are $\epsilon(n)$-close.[3] Recall that a function $\nu : \mathbb{N} \to \mathbb{N}$ is said to be negligible if for every polynomial $p(\cdot)$ and for every large enough n, $\nu(n) < 1/p(n)$.

Hard subset membership problems. A subset membership problem $\mathbf{M}$ specifies a collection $\{I_n\}_{n\in\mathbb{N}}$ of distributions, where for every n, I_n is a probability distribution over *instance descriptions*. Each instance description Λ specifies two finite non-empty sets $X, W \subseteq \{0,1\}^{poly(n)}$, and an NP-relation $R \subset X \times W$, such that the corresponding language $L \triangleq \{x : \exists w \; s.t. \; (x, w) \in R\}$ is non-empty. For every $x \in X$ and $w \in W$, if $(x, w) \in R$, we say that w is a *witness* for x. We use the following notation throughout the paper: for any instance description Λ we let $X(\Lambda)$, $W(\Lambda)$, $R(\Lambda)$ and $L(\Lambda)$ denote the sets specified by Λ.

Loosely speaking, subset membership problem $\mathbf{M} = \{I_n\}_{n\in\mathbb{N}}$ is said to be *hard* if for a random instance description $\Lambda \leftarrow I_n$, it is hard to distinguish random members of $L(\Lambda)$ from random non-members.

Definition 1 (Hard subset membership problem). *Let $\boldsymbol{M} = \{I_n\}_{n\in\mathbb{N}}$ be a subset membership problem as above. We say that $\boldsymbol{M}$ is* hard *if the ensembles $\{\Lambda_n, x_n^0\}_{n\in\mathbb{N}}$ and $\{\Lambda_n, x_n^1\}_{n\in\mathbb{N}}$ are computationally indistinguishable, where $\Lambda_n \leftarrow I_n$, $x_n^0 \in_R L(\Lambda_n)$, and $x_n^1 \in_R X(\Lambda_n) \setminus L(\Lambda_n)$.*[4]

Projective hash family. We next present the notion of a projective hash family with respect to a hard subset membership problem $\mathbf{M} = \{I_n\}_{n\in\mathbb{N}}$. Let $\mathcal{H} = \{H_k\}_{k\in K}$ be a collection of hash functions. K, referred to as the key space, consists of a set of keys such that for each instance description $\Lambda \in \mathbf{M}$,[5] there is a subset of keys $K(\Lambda) \subseteq K$ corresponding to Λ. For every Λ and for every $k \in K(\Lambda)$, H_k is a hash function from $X(\Lambda)$ to $G(\Lambda)$, where $G(\Lambda)$ is some finite non-empty set. We denote by $G = \bigcup_{\Lambda\in\mathbf{M}} G(\Lambda)$. We define a *projection key* function $\alpha : K \to S$, where S is the space of projection keys. Informally, a family $(\mathcal{H}, K, S, \alpha, G)$ is a projective hash family for $\mathbf{M}$ if for every instance description $\Lambda \in \mathbf{M}$ and for every $x \in L(\Lambda)$, the projection key $s = \alpha(k)$ uniquely

[2] Recall that $Dist(X, Y) \triangleq \frac{1}{2}\sum_{s\in S} |Pr[X = s] - Pr[Y = s]|$, or equivalently, $Dist(X, Y) \triangleq \max_{S'\subset S} |Pr[X \in S'] - Pr[Y \in S']|$, where S is any set that contains the support of both X and Y.

[3] For simplicity, throughout this paper we say that two random variables X_n and Y_n are statistically indistinguishable, meaning that the corresponding distribution ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$ are statistically indistinguishable.

[4] Note that this hardness requirement also implies that it is hard to distinguish between a random element $x \in_R L(\Lambda)$ and a random element $x \in_R X(\Lambda)$. We will use this fact in the proof of Theorem 1.

[5] We abuse notation and let $\Lambda \in \mathbf{M}$ denote the fact that $\Lambda \in support(I_n)$ for some n.

determines $H_k(x)$. (We stress that the projection key $s = \alpha(k)$ is only guaranteed to determine $H_k(x)$ for $x \in L(\Lambda)$, and nothing is guaranteed for $x \in X(\Lambda) \backslash L(\Lambda)$.)

Definition 2 (Projective hash family). *$(\mathcal{H}, K, S, \alpha, G)$ is a projective hash family for a subset membership problem $\boldsymbol{M}$ if for every instance description $\Lambda \in \boldsymbol{M}$ there is a well defined (not necessarily efficient) function f such that for every $x \in L(\Lambda)$ and every $k \in K(\Lambda)$, $f(x, \alpha(k)) = H_k(x)$.*

Efficient projective hash family. We say that a projective hash family is efficient if there exist polynomial time algorithms for: (1) Sampling a key $k \in_R K(\Lambda)$ given Λ; (2) Computing a projection $\alpha(k)$ from Λ and $k \in K(\Lambda)$; (3) Computing $H_k(x)$ from Λ, $k \in K(\Lambda)$ and $x \in X(\Lambda)$; and (4) Computing $H_k(x)$ from Λ, $(x, w) \in R(\Lambda)$ and $\alpha(k)$, where $k \in K(\Lambda)$. Notice that this gives two ways to compute $H_k(x)$: either by knowing the hash key k, or by knowing the projection key $\alpha(k)$ and a witness w for x.

Y-smooth projective hash family. Let Y be any function from instance descriptions $\Lambda \in \mathbf{M}$ to subsets $Y(\Lambda) \subseteq X(\Lambda) \setminus L(\Lambda)$. Loosely speaking, a projective hash family for $\mathbf{M}$ is Y-smooth if for every instance description $\Lambda = (X, W, R)$, for every $x \in Y(\Lambda)$, and for a random $k \in_R K(\Lambda)$, the projection key $\alpha(k)$ reveals (almost) nothing about $H_k(x)$.

Definition 3 (Y-smooth projective hash family). *A projective hash family $(\mathcal{H}, K, S, \alpha, G)$ for a subset membership problem $\boldsymbol{M}$ is said to be* Y-smooth *if for every (even maliciously chosen) instance description $\Lambda = (X, W, R)$ and every $x \in Y(\Lambda)$, the random variables $(\alpha(k), H_k(x))$ and $(\alpha(k), g)$ are statistically indistinguishable, where $k \in_R K(\Lambda)$ and $g \in_R G(\Lambda)$.*[6]

A Y-smooth projective hash family thus has the property that a projection of a (random) key enables the computation of $H_k(x)$ for $x \in L$, but gives almost no information about the value of $H_k(x)$ for $x \in Y(\Lambda)$.

Remark. This definition of Y-smooth projective hash family differs from the original definition proposed in [CS02] in two ways. First, it requires the smoothness property to hold against *maliciously* chosen instance descriptions Λ, whereas in [CS02] the smoothness is only with respect to $\Lambda \in \mathbf{M}$. Second, it requires the smoothness property to hold with respect to every $x \in Y$, whereas in [CS02] the smoothness condition is required to hold for randomly chosen $x \in_R X \setminus L$.

The main reason for our divergence from the original definition in [CS02] is that we need to cope with maliciously chosen Λ. We would like to set $Y = X \setminus L$ (as in [CS02]), and construct a $(X \setminus L)$-smooth projective hash family. However, we do not know how to construct such a family, for which the smoothness condition holds for *every* (even maliciously chosen) Λ.[7] Therefore,

[6] We assume throughout this paper, without loss of generality, that a (maliciously chosen) Λ has the same structure as an honestly chosen Λ.

[7] We note that [CS02,GL03] did not deal with maliciously chosen Λ's, and indeed the smoothness property of their constructions does not hold for maliciously chosen Λ's.

we relax our smoothness requirement and require only Y-smoothness, for some $Y \subseteq X \setminus L$. In both our constructions of Y-smooth projective hash families, $Y(\Lambda) \subset X(\Lambda) \setminus L(\Lambda)$ for maliciously chosen $\Lambda \notin \mathbf{M}$, and $Y(\Lambda) = X(\Lambda) \setminus L(\Lambda)$ for every honestly chosen $\Lambda \in \mathbf{M}$. Jumping ahead, the latter will enable the (honest) receiver to choose $x_b \in_R L(\Lambda)$, $x_{1-b} \in_R X(\Lambda) \setminus L(\Lambda)$ such that x_{1-b} is also in $Y(\Lambda)$. This will enable the (honest) sender to be convinced of its security by checking that either x_0 or x_1 is in $Y(\Lambda)$, and it will enable the (honest) receiver to be convinced that a (dishonest) sender cannot guess the bit b, assuming the underlying subset membership problem is hard. (From now on the reader should think of $Y(\Lambda)$ as equal to $X(\Lambda) \setminus L(\Lambda)$ for every $\Lambda \in \mathbf{M}$.)

Thus, we need a subset membership problem $\mathbf{M}$ such that for every honestly chosen $\Lambda \in \mathbf{M}$ it is easy to sample uniformly from both $L(\Lambda)$ and $X(\Lambda) \setminus L(\Lambda)$. On the other hand, for every (even maliciously chosen) (Λ, x_0, x_1) it is easy to verify that either $x_0 \in Y(\Lambda)$ or $x_1 \in Y(\Lambda)$. To this end we define the notion of a "Y-verifiably samplable" subset membership problem.

Definition 4 (Y-verifiably samplable subset membership problem). *A subset membership problem $\boldsymbol{M} = \{I_n\}_{n\in\mathbb{N}}$ is said to be* Y-verifiably samplable *if the following conditions hold.*

1. Problem samplability: *There exists a probabilistic polynomial-time algorithm that on input 1^n, samples an instance $\Lambda = (X, W, R)$ according to I_n.*
2. Member samplability: *There exists a probabilistic polynomial-time algorithm that on input an instance description $\Lambda = (X, W, R) \in \boldsymbol{M}$, outputs an element $x \in L$ together with its witness $w \in W$, such that the distribution of x is statistically close to uniform on L.*
3. Non-member samplability: *There exists a probabilistic polynomial-time algorithm $\mathcal{A}$ that given an instance description $\Lambda = (X, W, R) \in \boldsymbol{M}$ and an element $x_0 \in X$, outputs an element $x_1 = \mathcal{A}(\Lambda, x_0)$, such that if $x_0 \in_R L$ then the distribution of x_1 is statistically close to uniform on $X \setminus L$, and if $x_0 \in_R X$ then the distribution of x_1 is statistically close to uniform on X.*
4. Y-Verifiability: *There exists a probabilistic polynomial-time algorithm $\mathcal{B}$, that given* any *triplet (Λ, x_0, x_1), verifies that there exists a bit b such that $x_b \in Y(\Lambda)$. This should hold even if Λ is maliciously chosen. Specifically:*
   - *For every Λ and every x_0, x_1, if both $x_0 \notin Y(\Lambda)$ and $x_1 \notin Y(\Lambda)$ then $\mathcal{B}(\Lambda, x_0, x_1) = 0$.*
   - *For every honestly chosen $\Lambda \in \boldsymbol{M}$ and every x_0, x_1, if there exists b such that $x_b \in L(\Lambda)$ and $x_{1-b} \in support(\mathcal{A}(\Lambda, x_b))$, then $\mathcal{B}(\Lambda, x_0, x_1) = 1$.*

For simplicity, throughout the paper we do not distinguish between uniform and statistically close to uniform distributions. This is inconsequential.

3 Security of Oblivious Transfer

Our definition of oblivious transfer is similar to the ones considered in previous works on oblivious transfer in the Bounded Storage Model [DHRS04,CCM98].

A similar (somewhat weaker) definition was also used in [NP01] in the context of their DDH based two message oblivious transfer protocol.

In what follows we let $view_{\hat{S}}(\hat{S}(z), R(b))$ denote the view of a cheating sender $\hat{S}(z)$ after interacting with $R(b)$. This view consists of its input z, its random coin tosses, and the messages that it received from $R(b)$ during the interaction. Similarly, we let $view_{\hat{R}}(S(\gamma_0, \gamma_1), \hat{R}(z))$ denote the view of a cheating Receiver $\hat{R}(z)$ after interacting with $S(\gamma_0, \gamma_1)$.

Definition 5 (Secure implementation of Oblivious Transfer). *A two party protocol (S, R) is said to* securely implement oblivious transfer *if it is a protocol in which both the sender and the receiver are probabilistic polynomial time machines that get as input a security parameter n in unary representation. Moreover, the sender gets as input two strings $\gamma_0, \gamma_1 \in \{0,1\}^{\ell(n)}$, the receiver gets as input a choice bit $b \in \{0,1\}$, and the following conditions are satisfied:*

- Functionality: *If the sender and the receiver follow the protocol then for any security parameter n, any two input strings $\gamma_0, \gamma_1 \in \{0,1\}^{\ell(n)}$, and any bit b, the receiver outputs γ_b whereas the sender outputs nothing.*[8]
- Receiver's security: *For any probabilistic polynomial-time adversary $\hat{S}$, executing the sender's part, for any security parameter n, and for any auxiliary input z of size polynomial in n, the view that $\hat{S}(z)$ sees when the receiver tries to obtain the first message is computationally indistinguishable from the view it sees when the receiver tries to obtain the second message. That is,*
$$\{view_{\hat{S}}(\hat{S}(z), R(1^n, 0))\}_{n,z} \stackrel{\mathsf{c}}{\equiv} \{view_{\hat{S}}(\hat{S}(z), R(1^n, 1))\}_{n,z}$$
- Sender's security: *For any deterministic (not necessarily polynomial-time) adversary $\hat{R}$, executing the receiver's part, for any security parameter n, for any auxiliary input z of size polynomial in n, and for any $\gamma_0, \gamma_1 \in \{0,1\}^{\ell(n)}$, there exists a bit b such that for every $\psi \in \{0,1\}^{\ell(n)}$, the view of $\hat{R}(z)$ when interacting with $S(1^n, \gamma_b, \psi)$, and the view of $\hat{R}(z)$ when interacting with $S(1^n, \gamma_0, \gamma_1)$, are statistically indistinguishable.*[9] *That is,*
$$\{view_{\hat{R}}(S(1^n, \gamma_0, \gamma_1), \hat{R}(z))\}_{n,\gamma_0,\gamma_1,z} \stackrel{\mathsf{s}}{\equiv} \{view_{\hat{R}}(S(1^n, \gamma_b, \psi), \hat{R}(z))\}_{n,\gamma_b,\psi,z}$$

Note that Definition 5 (similarly to the definitions in [DHRS04,NP01]) departs from the traditional, simulation based, definition in that it handles the security of the sender and of the receiver separately. This results in a somewhat weaker security guarantee, with the main drawback being that neither the sender nor the receiver are actually guaranteed to "know" their own input. (This is unavoidable in two message protocols using "standard" techniques).

It is easy to show that Definition 5 implies simulatability for semi honest adversaries (the proof is omitted due to lack of space). More importantly, Definition 5 also gives meaningful security guarantees in face of malicious participants.

---

[8] This condition is also referred to as the completeness condition.

[9] We abuse notation by letting $S(1^n, \gamma_b, \psi)$ denote $S(1^n, \gamma_0, \psi)$ if $b = 0$, and letting it denote $S(1^n, \psi, \gamma_1)$ if $b = 1$.

In the case of a malicious sender, the guarantee is that the damage incurred by malicious participation is limited to "replacing" the input strings γ_0, γ_1 with a pair of strings that are somewhat "related" to the receiver's first message (without actually learning anything about the receiver's choice). In the case of a malicious receiver, Definition 5 can be shown to provide exponential time simulation of the receiver's view of the interaction (similarly to the definition of [NP01]). In particular, the interaction gives no information to an unbounded receiver beyond the value of γ_b. (Again, the proof is omitted due to lack of space.)

4 Constructing 2-Round OT Protocols

Let $\mathbf{M} = \{I_n\}_{n\in\mathbb{N}}$ be a hard subset membership problem which is Y-verifiably samplable, and let $(\mathcal{H}, K, S, \alpha, G)$ be a an efficient Y-smooth projective hash family for $\mathbf{M}$. Recall that the Y-verifiably samplable condition of $\mathbf{M}$ implies the existence of algorithms $\mathcal{A}$ and $\mathcal{B}$ as described in Section 2.

We assume for simplicity that for any n and for any $\Lambda \in I_n$, $G(\Lambda) = \{0,1\}^{\ell(n)}$, and that the two messages γ_0, γ_1, to be transferred in the OT protocol, are binary strings of length at most $\ell(n)$. Let n be the security parameter. Let (γ_0, γ_1) be the input of the sender and let $b \in \{0,1\}$ be the input of the receiver.

$R \to S$: The receiver chooses a random instance description $\Lambda = (X, W, R) \leftarrow I_n$. It then samples a random element $x_b \in_R L$ together with its corresponding witness w_b, using the member samplability algorithm, and invokes Algorithm $\mathcal{A}$ on input (Λ, x_b) to obtain a random element $x_{1-b} \in X \setminus L$. It sends (Λ, x_0, x_1).

$S \to R$: The sender invokes algorithm $\mathcal{B}$ on input (Λ, x_0, x_1) to verify that there exists a bit b such that $x_{1-b} \in Y(\Lambda)$. If $\mathcal{B}$ outputs 0 then it aborts, and if $\mathcal{B}$ outputs 1 then it chooses independently at random $k_0, k_1 \in_R K(\Lambda)$, and sends $\alpha(k_0)$ and $\alpha(k_1)$ along with $y_0 = \gamma_0 \oplus H_{k_0}(x_0)$ and $y_1 = \gamma_1 \oplus H_{k_1}(x_1)$.

R: The receiver retrieves γ_b by computing $y_b \oplus H_{k_b}(x_b)$ using the projection key $\alpha(k_b)$ and the pair (x_b, w_b).

We next prove that the above protocol is secure according to Definition 5. Intuitively, the receiver's security follows from the fact that x_b is uniformly distributed in L, x_{1-b} is uniformly distributed in $X \setminus L$, and from the assumption that it is hard to distinguish random L elements from random $X \setminus L$ elements. The sender's security follows from the assumption that $(\mathcal{H}, K, S, \alpha, G)$ is a Y-smooth projective hash family for $\mathbf{M}$, and from the assumption that one of x_0 or x_1 is in $Y(\Lambda)$ (otherwise, it will be detected by $\mathcal{B}$ and the sender will abort).

Theorem 1. *The above 2-round OT protocol is secure according Definition 5, assuming $\boldsymbol{M}$ is a Y-verifiably samplable hard subset membership problem, and assuming $(\mathcal{H}, K, S, \alpha, G)$ is a Y-smooth projective hash family for $\boldsymbol{M}$.*

Proof. we start by proving the receiver's security. Assume for the sake of contradiction that there exists a (malicious) probabilistic polynomial-time sender $\hat{S}$

such that for infinitely many n's there exists a polynomial size auxiliary input z_n such that $\hat{S}(z_n)$ can predict (with non-negligible advantage) the choice bit b when interacting with $R(1^n, b)$. In what follows, we use $\hat{S}(z_n)$ to break the hardness of $\mathbf{M}$, by distinguishing between $x \in_R L$ and $x \in_R X$. Given an instance description $\Lambda = (X, W, R) \leftarrow (I_n)$ and an element $x \in X$:

1. Choose at random a bit b and let $x_b = x$
2. Apply algorithm $\mathcal{A}$ on input (Λ, x_b) to obtain an element x_{1-b}.
3. Feed $\hat{S}(z_n)$ the message (Λ, x_0, x_1), and obtain its prediction bit b'.
4. If $b' = b$ then predict "$x \in_R L$" and if $b' \neq b$ then predict "$x \in_R L$."

Notice that if $x_b \in_R L$ then $\hat{S}(z_n)$ will predict the bit b with non-negligible advantage (follows from our contradiction assumption). On the other hand, if $x_b \in_R X$ then x_{1-b} is also uniformly distributed in X. In this case it is impossible (information theoretically) to predict b.

We now turn to prove the sender's security. Let $\hat{R}$ be any (not necessarily polynomial time) malicious receiver, and for any $n \in \mathbb{N}$, let z_n be any polynomial size auxiliary information given to $\hat{R}$. Let (Λ_n, x_0, x_1) be the first message sent by $\hat{R}(z_n)$. Our goal is to show that for every $n \in \mathbb{N}$ and for every $\gamma_0, \gamma_1 \in \{0,1\}^{\ell(n)}$, there exists $b \in \{0,1\}$ such that the random variables $view_{\hat{R}}(S(1^n, \gamma_0, \gamma_1), \hat{R}(z_n))$ and $view_{\hat{R}}(S(1^n, \gamma_b, \psi), \hat{R}(z_n))$ are statistically indistinguishable.

We assume without loss of generality that either $x_0 \in Y(\Lambda_n)$ or $x_1 \in Y(\Lambda_n)$. If this is not the case, the sender aborts the execution and b can be set to either 0 or 1. Let b be the bit satisfying $x_{1-b} \in Y(\Lambda_n)$. By the Y-smoothness property of the hash family, the random variables $(\alpha(k), H_k(x_{1-b}))$ and $(\alpha(k), g)$ are statistically indistinguishable, for a random $k \in_R K(\Lambda_n)$ and a random $g \in_R G(\Lambda_n)$. This implies that the random variables $(\alpha(k), \gamma_{1-b} \oplus H_k(x_{1-b}))$ and $(\alpha(k), g)$ are statistically indistinguishable, which implies that $view_{\hat{R}}(S(1^n, \gamma_0, \gamma_1), \hat{R}(z))$ and $view_{\hat{R}}(S(1^n, \gamma_b, \psi), \hat{R}(z))$ are statistically indistinguishable.

5 Constructing Smooth Projective Hash Families

We next present two constructions of Y-smooth projective hash families for hard subset membership problems which are Y-verifiably samplable. One based on the N'th Residuosity Assumption, and the other based on the Quadratic-Residuosity Assumption together with the Extended Reimann Hypothesis. A key vehicle in both constructions is the notion of an (ϵ, Y)-universal projective hash family.

Definition 6 (Universal projective hash families). *Let $\boldsymbol{M} = \{I_n\}_{n \in \mathbb{N}}$ be any hard subset membership problem. A projective hash family $(\mathcal{H}, K, S, \alpha, G)$ for $\boldsymbol{M}$ is said to be (ϵ, Y)-*universal *if for every n, every (maliciously chosen) Λ corresponding to the security parameter n, every $x \in Y(\Lambda)$ and every $g \in G(\Lambda)$, $Pr_{k \in_R K(\Lambda)}[H_k(x) = g \mid \alpha(k)] \leq \epsilon(n)$.*

As shown in [CS02], it is possible to reduce the error rate in a (ϵ, Y)-universal projective hash family from ϵ to ϵ^t (via independent repetitions). Once the error rate is reduced to be a negligible function ϵ^t, it is possible to transform the (ϵ^t, Y)-universal projective hash family into a Y-smooth projective hash family by applying the Leftover Hash Lemma. Both transformations preserve efficiency (up to polynomial factors). Due to lack of space we omit the details of these transformations, and we refer the interested reader to [CS02].

We conclude that it suffices to construct subset membership problems which are Y-verifiably samplable and for which there exists an efficient $(\frac{1}{2}, Y)$-universal projective hash family. In the remaining of this paper we present two such constructions – the first based on the N'th Residuosity Assumption, and the second based on the Quadratic-Residuosity Assumption together with the Extended Reimann Hypothesis.

5.1 N'th Residuosity Assumption

The N'th Residuosity Assumption. Let p, q be distinct safe primes; namely $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are odd primes. Let $N = pq$ and let J_{N^2} be the subgroup of $\mathbb{Z}^*_{N^2}$, consisting of all elements with Jacobi symbol 1. Let P be the subgroup consisting of all N'th powers of elements in J_{N^2}. The N'th Residuosity Assumption, originally introduced by Paillier [Pa99], asserts that given only N, it is hard to distinguish random elements of J_{N^2} from random elements of P.[10,11]

Overview of the constructions under the N'th Residuosity Assumption. We would like to use the constructions given in [CS02]. They construct a subset membership problem that generates instances where $X = J_{N^2}$ and $L = P$ (so that the hardness property follows from the N'th Residuosity Assumption). They define a corresponding universal projective hash family by $H_k(x) = x^k (\text{mod } N^2)$, with the projection key of k being $\alpha(k) = g^{Nk} (\text{mod } N^2)$, where $g^N (\text{mod } N^2)$ is an a priori chosen generator for L. In their proof of the universal property, they make strong use of the fact that for honestly chosen N's (N's which are a product of two safe primes), P can also be characterized by $P = \{x \in J_{N^2} : \mathit{order}(x) \text{ is co-prime to } N\}$. In our case we must also consider maliciously chosen N's, in which case this characterization does not remain true.

In order to ensure that for every N (even maliciously chosen), it still holds that every element in L is of order which is co-prime to N, we change the definition of L: rather than taking L to be all the N'th powers elements in J_{N^2}, we take L to be all the T'th powers elements in J_{N^2}, where $T \triangleq N^{\lceil \log N \rceil + 1}$. As we shall see shortly, this ensures that for every (even maliciously chosen) N, every element in L is of order which is co-prime to N, and for every honestly chosen N, this new L is equal to the previous one, and thus it remains hard

[10] Actually, Paillier did not make the restriction to safe primes or to elements in J_{N^2}. We note that the N'th Residuosity Assumption without these restrictions implies the N'th Residuosity Assumption with these restrictions, assuming that safe primes are sufficiently dense, as we do here. We refer the reader to [CS02] for more details.

[11] Jumping ahead, the reason that we restrict our attention to elements in J_{N^2} is that this results with the subgroup L being cyclic. This is an important point that will be elaborated on below.

to distinguish random X elements from random L elements, under the N'th Residuosity Assumption.

The subset membership problem M.

1. *Problem samplability:* For every n, I_n is a samplable distribution that generates an instance description Λ as follows: On input 1^n,
   (a) Generate two random n bit safe primes p, q; namely, primes p and q such that $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are odd primes. Let $N = pq$, $N' = p'q'$, and $T \triangleq N^{\lceil \log N \rceil + 1}$.
   (b) Generate a random (non-square) element $g \in \mathbb{Z}^*_{N^2}$ with Jacobi symbol 1, by choosing a random element $\mu \in_R \mathbb{Z}^*_{N^2}$ and setting $g = -\mu^2 (\bmod\ N^2)$.[12]
   (c) Output $\Lambda = (N, \mu)$, which specifies (X, W, R) in the following way: $X \triangleq J_{N^2}$, $L \triangleq \langle g^T \rangle$ is the subgroup generated by $g^T (\bmod\ N^2)$, $W \triangleq \{0, 1, \ldots, \lfloor N/2 \rfloor\}$, and $R \triangleq \{(g^{Tr}, r) : r \in W\}$.

   Notice that for every (even maliciously chosen) N, it holds that $L \subseteq \{x \in J_{N^2} : \mathit{order}(x) \text{ is co-prime to } N\}$. This is the case since the order of g divides $N\phi(N)$ (which is the order of $Z^*_{N^2}$), p and q divide $\phi(N)$ at most $\lceil \log N \rceil$ times, and they divide N exactly once. Thus p and q divide $N\phi(N)$ at most $\lceil \log N \rceil + 1$ times, and thus they divide the order of g at most $\lceil \log N \rceil + 1$ times. This implies that the order of $g^T (\bmod\ N^2)$ (where $T = N^{\lceil \log N \rceil + 1}$) is co-prime to both p and q, and thus is co-prime to N.
   Moreover, for every honestly chosen N, with overwhelming probability $L = P = \{x \in J_{N^2} : \mathit{order}(x) \text{ is co-prime to } N\}$. This follows from the fact that $|J_{N^2}| = N\phi(N)/2 = 2NN'$, which implies that P is a cyclic group of order $2N'$. Thus, for any random non-square element g in J_{N^2}, g^N is a generator of P with overwhelming probability. Moreover, since the order of g^N is co-prime to N, it follows that $\langle g^T \rangle = \langle g^N \rangle$.
   Let $Y(\Lambda) = \{x \in J_{N^2} : \mathit{order}(x) \text{ is } \textit{not} \text{ co-prime to } N\}$.[13]
2. *Member samplability:* On input $\Lambda = (N, \mu)$, choose a random $r \in_R W$ and output $g^{Tr} \in L(\Lambda)$ together with its corresponding witness r, where $g = -\mu^2 (\bmod\ N^2)$
3. *Non-member samplability:* On input $\Lambda = (N, \mu)$ and $x \in L(\Lambda)$, $\mathcal{A}$ chooses a random $a \in_R \{1, \ldots, N-1\}$, and outputs $x(1 + aN) \in X(\Lambda) \setminus L(\Lambda)$.
   Notice that for every $a \in \{1, ..., N-1\}$, $\mathit{order}(1 + aN)$ divides N (and is different than 1), which implies that $1 + aN \in Y(\Lambda)$.
4. *Y-Verifiability:* On input $(1^n, \Lambda, x_0, x_1)$, $\mathcal{B}$ outputs 1 if and only if $x_0, x_1 \in J_{N^2}$, $x_0 \neq x_1$, and $(x_0/x_1)^N = 1(\mathit{mod}\ N^2)$.

The fact that **M** is a hard subset membership problem follows from the N'th Residuosity Assumption and from the fact that for every honestly chosen $\Lambda \in \mathbf{M}$, with overwhelming probability $L(\Lambda) = P$.

---

12 Recall that for N which is a product of two safe primes $-1 \in J_N \setminus QR_N$.

13 Notice that for every (even maliciously chosen) Λ, it holds that $Y(\Lambda) \subseteq X \setminus L$, and for honestly chosen Λ it holds that $Y(\Lambda) = X(\Lambda) \setminus L(\Lambda)$ with overwhelming probability.

We next show that $\mathbf{M}$ is Y-verifiably samplable, under the Nth Residuosity Assumption. Fix any $\Lambda = (N, \mu) \in \mathbf{M}$. It is easy to see that the member samplability algorithm samples a random element in L. Moreover, notice that $X = P \cdot H \triangleq \{x \cdot y : x \in P,\ y \in H\}$, where $H \triangleq \langle 1 + N \rangle$. This is the case since for every N which is a product of two safe primes, it holds that $P \cap H = \{1\}$ (since the order of elements in P divide $2N'$, the order of elements in H divide N, and $GCD(2N', N) = 1$). This implies that $|P \cdot H| = |P| \cdot |H| = 2N'N$, which together with the fact that $P \cdot H \subseteq J_{N^2}$ implies that $P \cdot H = J_{N^2}$. Now, recall that $\mathcal{A}(\Lambda, x) = x(1 + aN)$ for some uniformly chosen $a \in \{1, \ldots, N-1\}$. Thus, if $x \in_R X$ then $\mathcal{A}(\Lambda, x) \in_R X$, and if $x \in_R L$ then $\mathcal{A}(\Lambda, x) \in_R X \setminus L$, which implies that the non-member samplability requirement holds.

It remains to show that the Y-verifiability requirement holds. Notice that for every (even maliciously chosen) N and for every $x \neq 1$ such that $x^N = 1 (\text{mod } N^2)$, it holds that $x \in Y(\Lambda)$. Thus, for every distinct x_0, x_1, if $(x_0/x_1)^N = 1 (\text{mod } N^2)$ then $x_0/x_1 \in Y(\Lambda)$, which implies that either $x_0 \in Y(\Lambda)$ or $x_1 \in Y(\Lambda)$.

$(\frac{1}{2}, Y)$-Universal Projective Hashing for $\mathbf{M}$. Consider the projective hash family $(\mathcal{H}, K, S, \alpha, G)$, defined as follows. For every $\Lambda = (N, \mu) \in \mathbf{M}$:

- Let $K(\Lambda) = \{0, 1, \ldots, \lfloor \frac{N^2}{2} \rfloor\}$ and let $K = \bigcup_{\Lambda \in \mathbf{M}} K(\Lambda)$.
- Let $G(\Lambda) = J_{N^2}$ and let $G = \bigcup_{\Lambda \in \mathbf{M}} G(\Lambda)$.
- For every $k \in K(\Lambda)$, let $H_k(x) = x^k (\text{mod } N^2)$.
- For every $k \in K(\Lambda)$, let $\alpha(k) = g^{Tk} (\text{mod } N^2)$, where $T \triangleq N^{\lceil \log N \rceil + 1}$ and $g = -\mu^2 (\text{mod } N^2)$.

Claim. $(\mathcal{H}, K, S, \alpha, G)$ is an efficient $(\frac{1}{2}, Y)$-universal projective hash family for $\mathbf{M}$.

Proof. It is straightforward to verify that all the efficiency requirements hold. As for the projection requirement, this follows from the fact that for every $k \in K(\Lambda)$ and every $x = g^{Tr} (mod\ N^2) \in L(\Lambda)$,

$$H_k(x) = x^k (mod\ N^2) = (g^{Tr})^k (mod\ N^2) = (g^{Tk})^r (mod\ N^2) = \alpha(k)^r (mod\ N^2).$$

We next show that it is $(\frac{1}{2}, Y)$-universal. Fix any (even maliciously chosen) $\Lambda = (N, \mu)$, and let $Z \triangleq \phi(N^2)/GCD(\phi(N^2), T)$. Notice that $GCD(N, Z) = 1$, which implies that for every $y \in Y(\Lambda)$, $y^Z \neq 1 (mod\ N^2)$ (since the order of y is *not* co-prime to N). Also notice that for every hash key k, $\alpha(k) = \alpha(k + Z)$. Fix any $y \in Y(\Lambda)$. Since for every s there are at least two elements $k, k + Z \in K(\Lambda)$ such that $s = \alpha(k) = \alpha(k + Z)$, and since $y^Z \neq 1$, it follows that s does not uniquely determine $H_k(y)$, implying that $(\mathcal{H}, K, S, \alpha, G)$ is a $(\frac{1}{2}, Y)$-universal projective hash family.

5.2 The Quadratic Residuosity Assumption

The Quadratic Residuosity Assumption. Let p, q be distinct safe primes; namely, $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are odd primes. Let $N = pq$, let J_N be the subgroup of $\mathbb{Z}_N^*$ consisting of all elements with Jacobi symbol 1, and let QR_N be the subgroup of $\mathbb{Z}_N^*$ consisting of all quadratic residues (note that $QR_N \subseteq J_N$). The Quadratic Residuosity Assumption asserts that given only N, it is hard to distinguish random elements of J_N from random elements of QR_N.

Overview of the constructions under the Quadratic Residuosity Assumption. We would like to use the constructions given in [CS02]. They construct a subset membership problem that generates instances where $X = J_N$ and $L = QR_N$ (so that the hardness property follows from the Quadratic Residuosity Assumption). They define a corresponding universal projective hash family by $H_k(x) = x^k (\text{mod } N)$, with the projection key of k being $\alpha(k) = g^{2k} (\text{mod } N)$, where $g^2 (\text{mod } N)$ is an a priori chosen generator for L. In their proof of the universal property, they make strong use of the fact that for honestly chosen N's (N's which are a product of two safe primes), QR_N can also be characterized by $QR_N = \{x \in J_N : \mathit{order}(x) \text{ is odd}\}$. In our case we must also consider maliciously chosen N's, in which case this characterization does not remain true.

In order to ensure that for every N (even maliciously chosen), it still holds that every element in L is of odd order, we change the definition of L: rather than taking L to be the set of all squares in J_N, we take L to be the set of all the T'th powers elements in J_N, where $T \triangleq 2^{\lceil \log N \rceil}$. As we shall see shortly, this ensures that for every (even maliciously chosen) N, every element in L is of odd order, and for every honestly chosen N, this new L is equal to the previous one, and thus it remains hard to distinguish random X elements from random L elements, under the Quadratic Residuosity Assumption.

We would like to prove that this subset membership problem, which generates instances with $X = J_N$ and $L = QR_N$ (with overwhelming probability for honestly chosen N's), is Y-verifiably samplable for some $Y \subseteq J_N \setminus QR_N$. However, achieving the non-member samplability property is quite problematic. The crux of the problem is that we cannot efficiently sample an element in $J_N \setminus QR_N$ for maliciously chosen N's.[14] What we do know (under the Extended Reimann Hypothesis) is how to sample $\log^3 N$ elements such that at least one of them is in $J_N \setminus QR_N$ (though we don't know which one).[15] Thus, rather than constructing a Y-verifiably samplable subset membership problem, which is associated with a single algorithm $\mathcal{A}$ for sampling a non-member element, we will construct a subset membership problem with many ($t = \log^3 N$) algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_t$, with the guarantee that *at least one* of them is actually sampling a non-member element. Correspondingly, there will be many verification algorithms $\mathcal{B}_1, \ldots, \mathcal{B}_t$, with the guarantee that for every i it holds that $\mathcal{B}_i(\Lambda, x, \mathcal{A}_i(x)) = 1$, and that

---

[14] Indeed, for N's that are a product of two safe primes $-1 \in J_N \setminus QR_N$, but this is not guaranteed in general.

[15] There is a subtle issue here. The above statement is not true if N is a power of a single prime (i.e., if N is of the form $N = p^\alpha$, for some prime p and some $\alpha \geq 1$), since in this case $J_N \setminus QR_N = \emptyset$. Fortunately, we can assume from now on (without loss of generality) that N is never of that form, since this can be checked in polynomial time.

at least one of the $\mathcal{B}_i$'s outputs 1 on input (Λ, x_0, x_1) only if either $x_0 \in Y(\Lambda)$ or $x_1 \in Y(\Lambda)$.

The idea would be to use this subset membership problem to construct an oblivious transfer protocol as follows:

$R \rightarrow S$: On input $b \in \{0,1\}$, the receiver chooses a random instance description Λ together with t pairs $(x_0^1, x_1^1), \ldots, (x_0^t, x_1^t)$, and corresponding t witnesses $w_b^1, \ldots, w_b^t$, such that for each $i \in \{1, \ldots, t\}$ it holds that $x_b^i \in_R L(\Lambda)$, $(x_b^i, w_b^i) \in R(\Lambda)$, and $x_{1-b}^i = \mathcal{A}_i(\Lambda, x_b^i)$. It sends $(x_0^1, x_1^1), \ldots, (x_0^t, x_1^t)$.

$S \rightarrow R$: The sender first checks that $\mathcal{B}_i(\Lambda, x_0^i, x_1^i) = 1$ for all $i \in \{1, \ldots, t\}$. If this check does not pass then he aborts. If the check does pass then the sender splits his input (γ_0, γ_1) into t random shares $(\gamma_0^1, \gamma_1^1) \ldots, (\gamma_0^t, \gamma_1^t)$. He then chooses t pairs of random hash keys $(k_0^1, k_1^1), \ldots (k_0^t, k_1^t)$, and sends for each $i \in \{1, \ldots, t\}$ the projection keys $\alpha(k_0^i)$ and $\alpha(k_1^i)$ together with the values $y_0^i = H_{k_0^i}(x_0^i) \oplus \gamma_0^i$ and $y_1^i = H_{k_1^i}(x_1^i) \oplus \gamma_1^i$.

R: The receiver retrieves γ_b by computing $y_b^i \oplus H_{k_b^i}(x_b^i)$, using the projection key $\alpha(k_b^i)$ and the pair (x_b^i, w_b^i), and by computing the XOR of all these values.

The sender's security is ensured since we know (under the Extended Reimann Hypothesis) that one of the $\mathcal{B}_i$'s outputs 1 only if one of the elements x_0^i or x_1^i is in $Y(\Lambda)$, which implies that at least one of the γ_b^i is statistically hidden, which in turn implies that γ_b is statistically hidden. The receiver's security follows from the fact that for every i and for $\Lambda \leftarrow I_n$, it is hard to distinguish between $x_0 \in_R L(\Lambda)$ and $\mathcal{A}_i(\Lambda, x_0)$.

The subset membership problem M. Our subset membership problem $\mathbf{M} = \{I_n\}_{n \in \mathbb{N}}$ is based on the one defined in [CS02]. However, we incorporate here several modifications.

1. *Problem samplability.* For every n, I_n is a samplable distribution that generates an instance description Λ as follows: On input 1^n,
   (a) Generate two random n bit safe primes p, q; namely, primes p and q such that $p' = \frac{p-1}{2}$ and $q' = \frac{q-1}{2}$ are odd primes. Let $N = pq$ and $T \triangleq 2^{\lceil \log N \rceil}$.
   (b) Choose a random element $\mu \in_R \mathbb{Z}_N^*$, and output $\Lambda = (N, \mu)$, which specifies (X, W, R) in the following way: $X \triangleq J_N$, $L \triangleq \langle \mu^T \rangle$ is the subgroup generated by $\mu^T (\text{mod } N)$, $W \triangleq \{0, 1, \ldots, \lfloor N/4 \rfloor\}$, and $R \triangleq \{(\mu^{Tr}, r) : r \in W\}$.

   Notice that $L \subseteq \{x \in J_N : \mathit{order}(x) \text{ is odd}\}$, for every (even maliciously chosen) N. This is the case since the order of μ divides $\phi(N)$ (which is the order of Z_N^*), and 2 divides $\phi(N)$ at most $\lceil \log N \rceil$ times. Thus, 2 divides the order of μ at most $\lceil \log N \rceil$ times. This implies that the order of $\mu^T (\text{mod } N)$ (where $T = 2^{\lceil \log N \rceil}$) is co-prime to 2, and thus is odd.

   Moreover, for every honestly chosen N, with overwhelming probability $L = QR_N = \{x \in J_N : \mathit{order}(x) \text{ is odd}\}$. This follows from the fact that QR_N is a cyclic group of order N', which implies that a random element in QR_N generates QR_N with overwhelming probability. Moreover, since the order of every element in QR_N is co-prime to 2, it follows that $\langle \mu^T \rangle = \langle \mu^2 \rangle$.

For every $\Lambda = (N, \mu)$, let $Y(\Lambda) = J_N \backslash QR_N$. Then for every (even maliciously chosen) Λ, it holds that $Y(\Lambda) \subseteq \{x \in J_N : order(x) \text{ is even}\}$.
2. *Member samplability:* On input $\Lambda = (N, \mu)$, choose a random $r \in W$, and output $\mu^{Tr} (\text{mod } N)$ together with its corresponding witness r.
3. *Non-member samplability $\mathcal{A}_i$:* On input $\Lambda = (N, \mu)$ and $x \in X(\Lambda)$, if $i \in J_N$ then $\mathcal{A}_i(\Lambda, x)$ outputs the element $i \cdot x (\text{mod } N)$. If $i \notin J_N$ then $\mathcal{A}_i(\Lambda, x)$ outputs x.[16]
4. *Y-Verifiability $\mathcal{B}_i$:* On input (Λ, x_0, x_1), if $i \in J_N$, then $\mathcal{B}_i(\Lambda, x_0, x_1)$ outputs 1 when both $x_0, x_1 \in J_N$ and $x_b / x_{b-1} = i (\text{mod } N)$ for some $b \in \{0, 1\}$. If $i \notin J_N$ then $\mathcal{B}_i(\Lambda, x_0, x_1)$ always outputs 1.

We would like to prove that $\mathbf{M}$ is a Y-verifiably samplable hard subset membership problem. The hardness of $\mathbf{M}$ follows from the fact that with overwhelming probability over $\Lambda = (N, \mu) \leftarrow I_n$, it holds that $L(\Lambda) = QR_N$. In order to prove that $\mathbf{M}$ is Y-verifiably samplable, we need to prove that $\mathbf{M}$ satisfies the following three properties: member samplablility, non-member samplability, and Y-verifiability. It is easy to see that the member samplability property holds. In order to see that the non-member samplability property holds it suffices to notice that under the Quadratic Residuosity Assumption, for every large enough n, for $\Lambda = (N, \mu) \leftarrow I_n$, and for every $i = 1, \ldots, \log^3 N$, it is hard to distinguish between $x \in_R L(\Lambda)$ and $x' = \mathcal{A}_i(\Lambda, x)$. In order to show that the Y-verifiability property holds, it suffices to show that the Y-verifiability property holds for a single i. This we show under the Extended Riemann Hypothesis, using the following well known result from algebraic number theory.

Lemma 1 ([BS96], 8.5.9). *Assume the Extended Riemann Hypothesis. Let H be a non-trivial subgroup of Z_N^* of index d, and let C be a coset of H. Then the least prime whose residue belongs to C is $O(d^2 \log^2 N)$.*

Assume the Extended Riemann Hypothesis. We first use Lemma 1 to prove that for every (maliciously chosen) N one of the elements in $\{1, ..., \log^3 N\} \cap J_N$ is also an element in $J_N \setminus QR_N$.[17]

Consider any $N = p_1^{a_1} ... p_k^{a_k}$. Let G be the subgroup of Z_N^* consisting of all elements which are squares modulo p_1. Let $H \triangleq J_N \cap G$. Notice that both G and J_N are subgroups of Z_N^* of index 2, and that H is a subgroup of Z_N^* of index 4. Now let g be any element in J_N which is *not* a square modulo p_1 (i.e., $g \in J_N \setminus G$), and let $C = gH$ be a coset of H. According to Lemma 1, the Extended Riemann Hypothesis implies that one of the elements in $\{1, 2, \ldots, x\}$, where $x = O(d^2 \log^2 N) < \log^3 N$, must be an element in C. Notice that all elements in C are non-squares modulo p_1, which implies that $C \subseteq J_N \setminus QR_N$. Thus, we conclude that one of the elements in $\{1, 2, \ldots, x\} \subset \{1, 2, \ldots, \log^3 N\}$ is in $J_N \setminus QR_N$.

[16] For $i \notin J_N$, x can be distinguished from $i \cdot x$, since it is easy to check whether an element in $\mathbb{Z}_N^*$ has Jacobi symbol 1. Thus, in this case we simply let $\mathcal{A}_i(\Lambda, x)$ output x, to make sure that it is hard to distinguish x from $\mathcal{A}_i(\Lambda, x)$.

[17] In what follows we use our assumption that N is not a power of a single prime (if N is a power of a single prime then $J_N \setminus QR_N = \emptyset$).

Fix any (even maliciously chosen) $\Lambda = (N, \mu)$, and let $i \in \{1, ..., \log^3 N\} \cap J_N$ be an element in $J_N \setminus QR_N$. It is easy to see that for every x_0, x_1, if both are not in $Y(\Lambda) = J_N \setminus QR_N$ then $\mathcal{B}_i(\Lambda, x_0, x_1)$ outputs 0. Moreover, for any x_0, x_1, such that $x_b \in L(\Lambda)$ and $x_{1-b} \in \mathcal{A}_i(\Lambda, x_b)$ (for some b), it holds that $x_{1-b} = i \cdot x_b$ and $x_0, x_1 \in J_N$ (since $i \in J_N$), and thus $\mathcal{B}_i(\Lambda, x_0, x_1)$ outputs 1.

$(\frac{1}{2}, Y)$-universal projective hash family for M. Consider the projective hash family $(\mathcal{H}, K, S, \alpha, G)$, defined as follows. For every $\Lambda = (N, \mu) \in \mathbf{M}$:

- Let $K(\Lambda) = \{0, 1, \ldots, \lfloor \frac{N}{2} \rfloor\}$ and let $K = \bigcup_{\Lambda \in \mathbf{M}_i} K(\Lambda)$.
- Let $G(\Lambda) = J_N$ and let $G = \bigcup_{\Lambda \in \mathbf{M}_i} G(\Lambda)$.
- For every $k \in K(\Lambda)$, let $H_k(x) = x^k (\text{mod } N)$.
- For every $k \in K(\Lambda)$, let $\alpha(k) = \mu^{Tk} (\text{mod } N)$, where $T \triangleq 2^{\lceil \log N \rceil}$.

Claim. The hash family $(\mathcal{H}, K, S, \alpha, G)$ is an efficient $(\frac{1}{2}, Y)$-universal projective hash family for $\mathbf{M}$.

Proof. It is straightforward to verify that all efficiency requirements hold. As for the projection requirement, it follows easily from the fact for every $k \in K(\Lambda)$, and for every $x \in L(\Lambda)$:

$$H_k(x) = x^k (\text{mod } N) = (\mu^{Tr})^k (\text{mod } N) = (\mu^{Tk})^r (\text{mod } N) = \alpha(k)^r (\text{mod } N)$$

We next prove that this projective hash family is $(\frac{1}{2}, Y)$-universal. Fix any (even maliciously chosen) $\Lambda = (N, \mu)$, and fix any $x \in Y(\Lambda) = J_N \setminus QR_N$. As was previously mentioned, x is of even order. Let $Z \triangleq \phi(N)/GCD(\phi(N), T)$. Note that Z is an odd number, and that $\mu^{TZ} = 1 (\text{mod } N)$. Also note that for every s there are (at least) two distinct elements $k, k + Z \in K(\Lambda)$ such that $s = \alpha(k) = \alpha(k + Z)$. Thus, in order to prove that the $(\frac{1}{2}, Y)$-universal property holds, it remains to prove that $x^Z \neq 1 (\text{mod } N)$, which follows immediately from the fact that x is of even order.

Acknowledgements. First and foremost I would like to thank Alon Rosen. Although he refused to be a co-author of this paper, Alon's comments, suggestions and involvement played an essential part in the creation of this work. I would like to thank Ronald Cramer for pointing out and explaining the notion of smooth projective hashing. I would like to thank Shien Jin Ong for pointing out a crucial mistake that I had in the initial stage of this work. I would like to thank Eric Bach for pointing out Lemma 1. Finally, I would like to thank Shai Halevi and Shafi Goldwasser for their great comments and simplifying suggestions.

References

[AIR01] William Aiello, Yuval Ishai, Omer Reingold. Priced Oblivious Transfer: How to Sell Digital Goods. In *EUROCRYPT 2001*, pages 119-135, 2001.

[BS96] E.Bach and J. Shallit. Algorithmic Number Theory, Vol. 1: Efficient Algorithms. MIT Press, 1996.

[BM89] M. Bellare and S. Micali. Non-Interactive Oblivious Transfer and Applications. In *CRYPTO '89*, pages-547-557, 1989.

[CS98] R. Cramer and V.Shoup. A Practical Public Key Cryptosystem Provably Secure Against Adaptive Chosen Ciphertext Attack. In *CRYPTO 1998*, pages 13-25, 1998.

[CS02] R. Cramer and V.Shoup. Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In *Eurocrypt 2002*, Springer-Verlag (LNCS 2332), pages 45-64, 2002.

[Cre87] C. Crépeau. Equivalence between two flavours of oblivious transfers. In *CRYPTO 1987*, pages 350-354, 1987.

[CCM98] C. Cachin, C. Crépeau, Julien Marcil. Oblivious Transfer with a Memory-Bounded Receiver. In *FOCS 1998*, pages 493-502, 1998.

[DHRS04] Y. Z. Ding, D. Harnik, A. Rosen, R. Shaltiel. Constant-Round Oblivious Transfer in the Bounded Storage Model. In *TCC 2004*, pages 446-472,2004.

[Fr98] John B. Fraleigh. A first course in abstract algebra , 7th edition, Addison-Wesley 1998.

[EGL85] S. Even and O. Goldreich and A. Lempel. A Randomized Protocol for Signing Contracts. In *Communications of the ACM 28:6*, pages 637–647, 1985.

[GL03] R. Gennaro and Y. Lindell. A Framework for Password-Based Authenticated Key Exchange. In *EUROCRYPT 2003*, pages 524-543, 2003.

[Gol04] O. Goldreich. Foundations of Cryptography - Volume 2 (Basic Applications). Cambridge University Press, 2004.

[GMW87] O. Goldreich and S. Micali and A. Wigderson. How to Play any Mental Game - A completeness Theorem for Protocols with Honest Majority. In *STOC 1987*, pages 218-229, 1987.

[Hai04] Iftach Haitner. Implementing Oblivious Transfer Using Collection of Dense Trapdoor Permutations. In *TCC 2004*, pages 394-409, 2004.

[IR89] R. Impagliazzo and S. Rudich. Limits on the Provable Consequences of One-Way Permutations. In *STOC 89*, pages 44–61, 1989.

[KOY01] J. Katz, R. Ostrovsky, M. Yung. Efficient Password-Authenticated Key Exchange Using Human-Memorable Passwords. In *EUROCRYPT 2001*, pages 475-494, 2001.

[Kil88] J. Kilian. Founding Cryptography on Oblivious Transfer. 20th ACM Symposium on the Theory of Computing, pages 20–31, 1988.

[NP01] M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *SODA 2001*, pages 448-457,2001

[Pa99] P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *EUROCRYPT 1999*, pages 223-238, 1999.

[Rab81] M. O. Rabin. How to Exchange Secrets by Oblivious Transfer. TR-81, Harvard, 1981.

[Y86] A. C. Yao. How to Generate and Exchange Secrets. In *FOCS 1986*, pages 162-167, 1986.