Efficient Hybrid Proxy Re-Encryption for Practical Revocation and Key Rotation

Steven Myers[⋆] and Adam Shull[⋆⋆]

Indiana University, Bloomington, IN, USA

Abstract. We consider the problems of i) using public-key encryption to enforce dynamic access control on clouds; and ii) key rotation of data stored on clouds. Historically, proxy re-encryption, ciphertext delegation, and related technologies have been advocated as tools that allow for revocation and the ability to cryptographically enforce *dynamic* access control on the cloud, and more recently they have suggested for key rotation of data stored on clouds. Current literature frequently assumes that data is encrypted directly with public-key encryption primitives. However, for efficiency reasons systems would need to deploy with hybrid encryption. Unfortunately, we show that if hybrid encryption is used, then schemes are susceptible to a key-scraping attack.

Given a proxy re-encryption or delegation primitive, we show how to construct a new hybrid scheme that is resistant to this attack and highly efficient. The scheme only requires the modification of a small fraction of the bits of the original ciphertext. The number of modifications scales linearly with the security parameter and logarithmically with the file length: it does not require the entire symmetric-key ciphertext to be re-encrypted!

Beyond the construction, we introduce new security definitions for the problem at hand, prove our construction secure, discuss use cases, and provide quantitative data showing its practical benefits and efficiency. We show the construction extends to identity-based proxy re-encryption and revocable-storage attribute-based encryption, and thus that the construction is robust, supporting most primitives of interest.

1 Introduction

Data storage on the cloud is now a major business. Examples include Dropbox, Box, Google Drive, iCloud and many more. All of these services provide some degree of sharing and access control that allow one to share files with others, but they all come at the price that all of one's data is either i) encrypted under a key that the cloud has access to or ii) placed on the cloud in plaintext. This is necessary because the service provider acts as an all-trusted reference monitor that decides who can access data. It makes the plaintext data available only to those that are supposed to have access. This makes data held by such providers

[⋆] Email: `samyers@indiana.edu`

[⋆⋆] Email: `amshull@indiana.edu`

privy to insider and data exfiltration attacks that can put the data of large numbers of users at risk. Similarly, many content providers wish to provide content easily through the cloud to subscribing customers, but remove content access in scenarios where subscriptions terminate.

Cryptography seemingly provides natural solutions to the problem of untrusted cloud access control; tools from traditional public-key through attribute-based and predicate encryption allow one to store data on a public cloud, with cryptography enforcing access control functions. However, as detailed by Garrison et al. [17], these cryptographic techniques are not yet well-suited for even traditional dynamic access control policies, such as $RBAC_0$, where users may have their access rights to data changed or revoked over time. In particular, their work highlights the need for more efficient revocation mechanisms in such schemes.

Consider a typical cryptographic access control scenario where a file is encrypted under a public key, and those that have read access are given the secret key. Now if a user's access is revoked from a file that is shared amongst many on an untrusted server, the typical cryptographic solution involves providing new secret keys to all users that should continue to have access to the file, and then re-encrypting the file. When the server is not trusted with the underlying data but can be trusted to perform computation, proxy re-encryption or ciphertext delegation can be used. Otherwise, a user or administrator needs to download the revoked resources, process them by decrypting and then re-encrypting them under new keys, and then transmitting them back to the server. Such a process is expensive both in terms of bandwidth and time. Also, when the administrator is a thin client, it imposes an expensive computational cost. This is particularly true with devices such as smartphones that frequently share large data files, such as video.

Given that we typically can trusts clouds to compute correctly—even if we can't trust them against data exfiltration attacks—proxy re-encryption and ciphertext delegation are clearly solutions that need to be considered. In proxy re-encryption [7,22,4], a proxy (such as the cloud) converts a ciphertext from one key to another key without accessing the underlying plaintext. In proxy re-encryption, revocation works as follows: The administrator of the access control scheme generates new public- and secret-key pairs, passes the new secret keys to those who should have read access to the data, and sends the public key to those that should have write access. The administrator then generates a proxy re-encryption key and asks the cloud to proxy re-encrypt the data with said key. Here the cloud is trusted to compute on the data, but it is never exposed to sensitive data. Garrison et al. [17] give a scheme which implements $RBAC_0$ access controls on the cloud, and suggest it can be modified to use proxy re-encryption schemes to allow revocation on the cloud. Ciphertext delegation in attribute-based encryption (ABE) refers to the ability to re-encrypt a ciphertext so that it is harder to decrypt, without using any secret information. Sahai et al. [35] show how this can be used to achieve revocable-storage ABE.

Another related scenario is that of key rotation for encrypted files stored on the cloud. Key rotation is the process by which files encrypted and stored must be re-keyed on a timely basis. This ensures that if keys are accidentally leaked or otherwise revealed, the files' data remain secure. For example, corporations that regularly have consultants and other external visitors often give them temporary access. Corporations need to ensure that on a regular and timely basis such access is revoked. Because of frequent breakdowns in communication channels on the human side, files may not be re-encrypted under new keys when the consultant has moved on. Similarly, there is fear that as time progresses, keys are inadvertently exposed, or the chance of key-theft goes up. In either case, later exfiltration of the encrypted data can lead to its exposure. Therefore, it is typically recommend that regular key rotations take place. This ensures that there are no lingering access permissions to individuals who should no longer possess them, and to lock out those who have maliciously acquired a copy of a key. Key rotation is recommended across a wide range of industries and organizations. For example, NIST [5] recommends regular and planned rotation, as does the Open Web Application Security Project (OWASP) [30], and the payment card industry [31] requires it on a periodic basis for customer data. Proxy re-encryption schemes allows for such key rotation for files stored on the cloud or other untrusted servers. Even on the cloud, the cost of re-keying large databases can be expensive.

Work has been done on revocation and/or proxy re-encryption for symmetric-key encryption [38], public-key encryption [7,22,4], identity-based encryption [19,22], and attribute-based encryption [25]. The work in the symmetric case [38], while recognizing the problem and attack we addresses, provides non-standard security properties that are difficult to reconcile with modern security notions. The concern with all the asymmetric-primitive–based proxy re-encryption schemes is that while they exist, they all involve computing a number of expensive asymmetric primitives such that the number of such expensive computations scales at least linearly with the length of the file being encrypted. The work of Wang et al. [39] considers revocation for ABE when hybrid encryption is used. However, to grapple with the attack described next, they use a key-homomorphic pseudo-random function by Boneh et al. [10] for hybrid encryption, which has significant efficiency issues: it requires a number of exponentiations in cyclic groups that is linear in the file's length.

Key Scraping Attacks on Cryptographic Dynamic Access Control via Proxy Re-Encryption or Ciphertext Delegation The traditional solution for efficiently encrypting large files with asymmetric cryptography is to use hybrid encryption: files are encrypted first with the faster and more efficient symmetric-key encryption, and the short symmetric key is then encrypted with the asymmetric encryption scheme. Further, one can apply this with proxy re-encryption and ciphertext delegation schemes. However, the hybrid construction has a serious flaw for the motivating scenarios we describe: dynamic cryptographic access control and key rotation. In these scenarios, the adversary is initially a user that is supposed to have access to a file, and that access is later supposed to be revoked. Consider when data is stored using a hybrid proxy encryption scheme, and then the proxy re-encryption procedure is applied to the ciphertext—the naive usage can allow revoked users to retain access to re-encrypted files. Note that the asymmetric ciphertext which encrypts the symmetric key is renewed and encrypted under a new key, but the symmetric ciphertext is not modified at all. The unchanged symmetric key can be used to retrieve the plaintext from the "re-encrypted" ciphertext. The result is that a practical adversary, with minimal resources in bandwidth and storage, can download and decrypt symmetric keys for all the files that it is ever granted access to, and maintain that access in perpetuity if hybrid proxy re-encryptions are the only method ever used to revoke access. This concern has been observed independently by both Garrison et al. [17] and Wang et al. [39]. Garrison et al. consider it in the scenario of attempting to implement efficient $RBAC_0$ with proxy re-encryption on an untrusted cloud, to measure efficiency and practicality. Wang et al. consider it in the context of their Sieve system, which allows users to encrypt their data, provide it to the cloud, and the discriminantly allow different service providers access to data, which can later be revoked.

To make the problem more concrete, consider the following scenario based on Garrison et al. [17]: Content files are stored on a cloud and are hybrid-encrypted using a hybrid proxy re-encryption scheme with public-key encryption algorithm E and a symmetric-key encryption algorithm $\mathsf{E}^{\mathsf{Sym}}$. Alice has access to a large number of files $\{f_i\}_i$ that are encrypted on the cloud in the ciphertexts $\big\{\big(\mathsf{E}(\mathsf{pk}_{Sub_0}, k_i), \mathsf{E}^{\mathsf{Sym}}(k_i, f_i)\big)\big\}_i$. Alice has the secret key, sk_{Sub_0}, corresponding to public key pk_{Sub_0}, as she belongs to an initial group of subscribers, and the subscribers all have access to sk_{Sub_0}, the secret key for this role.[1] She does not have the resources to download all of the content files she has access to. She is removed from the subscriber group, so the cloud proxy re-encrypts all data under a new public-key pk_{Sub_1}, denoting the new group of valid subscribers, and to which Alice does not have the key. The result is that the cloud now serves $\big\{\mathsf{E}(\mathsf{pk}_{Sub_1}, k_i), \mathsf{E}^{\mathsf{Sym}}(k_i, f_i)\big\}_i$, and Alice cannot directly access the content in the subscription service.

However, while it may not be reasonable to assume that Alice can download all of the files she has access to on the cloud service while she is a subscriber, due to their collective size or rate limits on the outgoing service provider's network connection, it is more reasonable to assume that at some point Alice downloads and decrypts all of the symmetric keys $\{k_i\}_i$. Even for millions of files, this would

[1] In Garrison et al.[17] this key is accessed indirectly through another ciphertext specific to a given user which is encrypted under the user's personal public key. We simplify to keep the example simple.

We stress that while in traditional PKI settings, only one person has a given secret-key, in cryptographic access control settings this is not necessarily the case. This is further reflected in cryptographic systems more directly related to access control such as attribute-based encryption and predicate encryption, where a given set of credentials or a given access policy can result in the multiple users being given the same corresponding key.

require less than a gigabyte of storage/bandwidth, and she could use these keys to decrypt all of $\left\{\mathsf{E}^{\mathsf{Sym}}(k_i, f_i)\right\}_i$. Therefore, even if the symmetric keys are re-encrypted via proxy re-encryption, it is reasonable to assume that Alice would maintain the ability to decrypt the symmetric portion of the proxy hybrid re-encrypted files on the cloud. *One needs to ensure with hybrid re-encryption that ciphertexts are re-encrypted at both the public-key and symmetric-key ciphertext portions.*

Use Cases Cryptographic cloud-based dynamic access control has many applications from the storage and sharing of personal videos, data, and personal sensor information, to household IoT sensor data, digital content subscription services, and sharing of medical and sensitive corporate data. In all of these scenarios, there are use and business cases where it makes sense to place data on the cloud, but due to fear of data exfiltration or insider attacks on the cloud, the data should be stored in encrypted format. For example, a large number of works on attribute-based, predicate, and functional encryption are motivated by such use cases. However, only a small percentage of these papers consider dynamic revocation. Yet, being able to revoke, and revoke efficiently, can be extremely important in many real-world scenarios (e.g., termination of employment with cause, or a relationship ending on bad terms). Our scheme allows very efficient revocation that applies against all adversaries that do not go to the effort of downloading essentially all of the files that are to be revoked—and if an adversary has downloaded a file then revocation of said file doesn't provide much benefit in any case. Further, we simultaneously protect against key-scraping attacks. While other techniques have been proposed for this problem, such as that of the Sieve system in [39], our solution is significantly more efficient on any reasonably-sized file due to the need for such schemes to compute exponentiation for each block of the file, and our construction provides comparable security in real-world scenarios.

While one could use the cloud to provide access control against scraping attacks, by for example monitoring a user who accesses the encrypted symmetric-keys portion of too many files, this has several downsides. It implies that the cloud needs to have user accounts, and is aware of and actively records the history of such accesses, and implements access control denial when such occasions occur. The cloud thus monitors which files the users accesses, which portions, and how frequently, which for privacy, security and anonymity reasons may be very undesirable.

Consider the concrete use case of a subscription content service. With a traditional hybrid encryption scheme a malicious user may be tempted to download symmetric keys for the entire content service—performing a scraping attack—so that all the content could be accessed at a late time after the user stopped paying for the service. Our proposed scheme would limit the user to the material they could download while paying for the service. Note that a service can easily limit the download rate to be useful for legitimate users, but make mass download attacks of limited use, by simply limiting its network rate connections.

For example, a library might limit downloads to a few tens of books of data a day, and a streaming service might limit a user to the equivalent of 48 hours of high-definition video in a given day. This doesn't provide much limit on how much of the library of content a legitimate user might actually access. However, a key-scraping attack under such a rate-limit would permit access to a significant fraction of all content of the service. While one can ask the cloud to monitor which files are accessed and which portions, we are then placing more trust into the cloud to perform access control, and maintaining lists of which content users have been accessing has clear privacy concerns.

With respect to the scenario of key rotation of data stored on the cloud, our construction's ability to efficinetly rotate keys lowers its cost, and this can allow for more efficient and less costly key rotations on large data stores, or alternately may permit for more frequent key rotations due to lowered costs. Other systems, such as those proposed by Boneh et al. [10], which permit updating of symmetric encryptions through key-homomorphisms also fulfill this function, but their computational costs are significantly more expensive requiring, for each "block" of the file, exponentiations on cyclic groups where discrete log is hard.

Overview of our Contributions We define a new security notion aimed at proxy re-encryption which provides security against the previously described attacks in the revocation and key-rotation scenarios. In situations where users will have access to data that is to be later revoked, there is clearly nothing that can be done about an adversary that is willing to download and store all files for potential use later. Such an adversary can continue to access the file after revocation by using its local copy, regardless of whether the file is re-encrypted in the cloud. Therefore, we focus our definition on the realistic scenario that assumes that the adversary has not previously *stored* nor viewed the entire original ciphertext. In the increasingly typical setting of cloud storage, this corresponds to the adversary not downloading the entire ciphertext prior to having its access removed. In particular, our security definition specifically rules out symmetric key-scraping attacks. More specifically, our security definition's goal is that an adversary that has access to i) potentially the vast majority—but not all—of a ciphertext before any proxy re-encryption, and ii) the original decryption key, cannot learn anything about the underlying plaintext, even after giving the adversary access to a new proxy re-encrypted version—assuming the adversary does not possess the corresponding new secret key.

We provide a solution by giving a construction that satisfies our definition while performing proxy re-encryption in the hybrid model. Further, the cost of the proxy re-encryption does not scale directly with the length of the file, but rather as a linear function of the security parameter, and a logarithmic function of the length of the file. The real-world savings are significant, and ensure that when bulk revocations occur—as is often the case with key rotations and changes to access control policies—large numbers of large files can be revoked efficiently. We provide quantitative analysis showing the savings of our technique.

While our construction is susceptible to collusion between the cloud and a revoked user, this is unavoidable when the cloud is trusted to perform the proxy re-encryption because it can simply retain all the old files and then give them to the revoked user for decryption. Such collusion is unlikely in the case of a commercial cloud storage provider, as it would likely violate the service agreement and is much riskier than simply reading data stored unencrypted on the cloud.

We note that one might prefer to derive a solution such that any adversary that initially has access to the encrypted files, but is only willing to expense minimal storage to maintain access, will no longer be able to access the files after revocation or key rotation. We show that this cannot be done without making substantial modifications to each encrypted file. Thus solutions are possible, but they will be costly in terms of disk access, and such costs will scale linearly with file sizes, bypassing our efficiency gains.

Finally, we show how our definitions naturally extend to proxy re-encryption for identity-based encryption and delegation in attribute-based encryption. This demonstrates the construction can be used for dynamic cryptographic access control that supports expressive access policies. It further suggests the construction generalizes to other reasonable definitions of ciphertext delegation and proxy re-encryption for related encryption primitives.

Overview of our Construction We show how the novel use of an All-or-Nothing Transform (AONT) [33,11,12,15] with traditional ideas from hybrid encryption can be used to achieve an efficient hybrid proxy re-encryption scheme for asymmetric primitives that is friendly to revocation. In particular, the re-encryption algorithm only touches a small fraction of the symmetric-key–encrypted ciphertext, resulting in much greater efficiency than producing a fresh encryption of the plaintext. The re-encrypted ciphertext grows slightly in size by an additive length of one public-key encryption, and thus in practice by several hundred to several thousand bits. However, for the use cases discussed above, storage is typically cheap, and so this ciphertext growth adds a negligible cost.

For those versed in the area, the key idea of our construction is to take a traditional hybrid construction where we have an asymmetric proxy-scheme encryption of the symmetric key, and a symmetric-key encryption of the file in question. We then apply an AONT on top of the symmetric ciphertext. Upon proxy re-encryption we use the original proxy re-encryption scheme to update the asymmetric encryption to a new asymmetric key. We then pseudorandomly choose a number of locations in the AONT-transformed symmetric ciphertext to encrypt. We encrypt enough of the AONT's output that with all-but-negligible probability the adversary did not download some of the newly encrypted locations. Therefore, it cannot invert the AONT and decrypt the ciphertext. We then add a new asymmetric encryption of the symmetric key used to choose and encrypt the random bit locations, so that the appropriate decryptor can later invert all the operations and retrieve the appropriate locations. The number of locations to encrypt is roughly i) proportional to the inverse of the fraction of the file the adversary does not look at, and ii) proportional to the number of bits that need to be changed by the AONT, which ensures that chances of inversion are essentially zero.

2 Related Work

Proxy re-encryption was first considered in the public-key realm by Mambo and Okamoto [27], but first defined by Blaze et al. [7], where they introduced the notion of asymmetric and symmetric proxies for public-key encryption, digital signatures, and identification protocols. Symmetric and asymmetric proxies refer to the trust between the individuals in proxy schemes, and not symmetric and asymmetric-key cryptography. In symmetric proxies, if Alice is delegating to Bob, then not only must Alice trust Bob (because Bob can decrypt her ciphertexts), but also Bob must trust Alice since in such schemes Alice can combine her private key along with the proxy re-encryption key to compute Bob's secret key. In an asymmetric scheme, there is no such trust, and Bob's key is not at risk of being computed.

Jakobsson [23] provides the first steps towards an asymmetric proxy, but his scheme requires an honest quorum to perform the proxy-transformation and ensure that no information about the underlying plaintext is learned. Ivan and Dodis [22] gave the first asymmetric proxy re-encryption scheme for traditional public-key encryption, and the first for identity-based encryption. They also formalize and rename the notions of asymmetric and symmetric proxies to unidirectional and bidirectional, which prevents confusion with the associated terms in cryptography. Further, they formalize security definitions for proxy re-encryption, where they extend traditional notions of CPA and CCA security to hold against adversaries that do not have access to decryption keys/oracles of not only the delegator, but also that of any of the delegatees. Later, Ateniese et al. [4] pushed further, extending security definitions to include security of the delagator's secret key against colluding coalitions of delegatees, noting distinctions between transitivity and transferability of decryption rights, and providing constructions that meet all but one (they do not guarantee non-transferability) of their security notions. Further, they show their scheme is efficient and provide performance benchmarks showing that proxy re-encryption can be practical.

Green and Ateniese [19] gave the first proxy scheme for identity-based encryption (IBE), where individual identities could develop proxy keys without the need to incorporate the master secret key, a key practicality in many scenarios. Since this paper, there has been a number of works extending proxy schemes in IBE [40,37]. This was later extended to attribute-based encryption (ABE) by [25], and there has now, similarly, been a fair amount of work in this area.

Related is the notion of revocable encryption schemes. Such schemes have been constructed for PKI [28,29,1], for IBE [8], and for ABE [35]. In these schemes, certificates/keys can be revoked so that they cannot be used to decrypt ciphertexts encrypted in the future. These schemes differ from ours in that they are concerned with revoking access to future ciphertexts, while our scheme is used to revoke access to existing ciphertexts. However, the ABE scheme [35] also provides a mechanism for revoking access to previously encrypted ciphertexts by delegating the ciphertext to a later time. Since this scheme only delegates the ABE portion of the ciphertext and not the symmetric-key-encrypted portion, this scheme also has the security weakness our construction addresses.

Ateniese et al. [4] also provide a description of a secure file system scheme that uses proxy re-encryption. In their scheme, all files are encrypted under symmetric keys, and all the symmetric keys are encrypted under a master public key. When a user requests a file it has access to, the proxy will re-encrypt the file's symmetric key to the user's public key, giving the user access to the symmetric key and thus to the file's contents. However, this scheme does not consider what happens when a user is revoked; in particular, symmetric keys are never changed.

Proxy re-encryption has been considered in the symmetric realm by Syalim et al. [38]. Their scheme, like ours, uses an AONT as part of their proxy re-encryption. However, their security analysis only considers the ability of a revoked user to obtain the new encryption/decryption key when it possesses plaintexts, old keys, and both old and new ciphertexts. They do not consider our scenario, where the revoked user has to obtain the plaintext from the old key, part of an old ciphertext, and the new ciphertext. In addition, they focus strictly on the symmetric case, and do not address the use of their scheme as part of hybrid encryption.

Watanabe and Yoshino [41] present a mechanism for efficiently updating symmetric keys. They also use an AONT to improve efficiency. However, their scheme is in the symmetric key setting, and it does not consider revocation, where the adversary previously had legitimate access to the file.

Boneh et al. [10] show how to use key-homomorphic pseudorandom functions to implement symmetric-key proxy re-encryption. A pseudorandom function $F : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$ for keyspace $\mathcal{K}$ is key-homomorphic if $F(k_1, x)$ and $F(k_2, x)$ can be combined to produce $F(k_1 \oplus k_2, x)$ for some group operation $\oplus$. Using a key-homomorphic PRF allows the symmetric key to be updated easily, providing a much cleaner solution than the one used in our scheme. However, current constructions of key-homomorphic PRFs are far too inefficient to be used in practice, and their constructions would require asymmetric operations that scale directly with the length of the file being encrypted.

We note that the problem we are addressing may, on first appearance, may have similarities to the Bounded Retrieval Model [14,16], and the recent work of Bellare et al. [6] which tries to make symmetric keys secure against exfiltration. However, while similar in that they both restrict the amount of a secret (or ciphertext) that can be downloaded, they are quite different in that in our setting i) the adversary is later given full access to another transformed version of the ciphertext, ii) the adversary initially has access to the secret key for the ciphertext and is typically able to download the entirety of the symmetric key embedded in the ciphertext, and iii) the legitimate user is expected to download the entire ciphertext in order to decrypt.

3 Notation

Given a string s over a given alphabet, we denote by $|s|$ the length of the string. A function μ is negligible if it grows slower than any inverse polynomial. Let $\mathcal{D}_1 = \{D_{1,i}\}_{i\in\mathbb{N}}$ and $\mathcal{D}_2 = \{D_{2,i}\}_{i\in\mathbb{N}}$ be two indexed sequences of distributions, then $\mathcal{D}_1 \approx \mathcal{D}_2$ denotes that the two sequences are computationally indistinguishable [24]. Let $[N]$ denote $\{1, \ldots, N\}$ and let $\left\{ {N \atop \ell} \right\}$ be the set of all ℓ-element subsets of $[N]$. For $y \in \{0,1\}^N$ and $L \in \left\{ {N \atop \ell} \right\}$, we use $[y]_L$ to denote the $N - \ell$ bits of y that are not in L.

4 Background Definitions

In this section we overview the definitions of the component primitives necessary for our construction. We assume the reader is familiar with the concept of symmetric-key encryption, and chosen-plaintext attack (CPA) security for such encryption. Definitions are included in Appendix A for those that are not.

4.1 All-Or-Nothing Transforms

All-or-nothing transforms were introduced by Rivest [33], as a primitive which intuitively presented a type of function that had the property that without access to essentially the entire output, no party could retrieve any bit of the underlying input. However, with the full output of the function, the input was easily retrievable. The notion was formalized by Boyko [11] in the Random Oracle Model, and later by Canetti et al. [12] in the standard model. Later Dodis et al. [15] extended the definition to include more powerful adaptive adversaries that are allowed to adaptively choose which output bit positions the adversary wishes to see.

Definition 1 (Adaptive and Non-Adaptive AONTs [12,15]). *A randomized polynomial time computable function $T : \{0,1\}^n \to \{0,1\}^N$ is a (non-adaptive) ℓ-AONT if it satisfies conditions 1 and 2. It is an Adaptive ℓ-AONT if it satisfies conditions 1 and 3.*

1. *T is efficiently invertible, i.e., there is a polynomial time machine I such that for any $x \in \{0,1\}^n$ and any $y \leftarrow T(x)$, we have $I(y) = x$.*
2. *(Non-adaptive) For any $L \in \left\{ {N \atop \ell} \right\}$ and any PPT adversary $\mathcal{A}$, we have:*
$$|\Pr[\mathcal{A}(x_0, x_1, [T(x_0)]_L) = 1] - \Pr[\mathcal{A}(x_0, x_1, [T(x_1)]_L) = 1]| \leq \varepsilon(N)$$
for some negligible function ε.
3. *(Adaptive) For any PPT adversary $\mathcal{A}$ with oracle access to string $y = T(x_b)$ who can read at most $N - \ell$ bits of y, we have:*
$$\left|\Pr\left[\mathcal{A}^{T(x_0)}(x_0, x_1) = 1\right] - \Pr\left[\mathcal{A}^{T(x_1)}(x_0, x_1) = 1\right]\right| \leq \varepsilon(N)$$
for some negligible function ε.

Construction of AONTs The first construction of an AONT was given by Rivest [33], which requires two passes of the input with a block cipher. However, it is only known to satisfy a weaker definition of security than Defn. 1. Boyko [11] showed that Optimal Asymmetric Encryption Padding (OAEP) satisfies the non-adaptive version of Defn. 1 in the random oracle model. Extending the work of Canetti et al.[12] and Dodis et al. [15] we show that OAEP is also an adaptively secure AONT in the Random Oracle Model. We note that OAEP is quite efficient, requiring two cryptographic hashes over the length of the input.

Lemma 1. *Let $G : \{0,1\}^k \to \{0,1\}^n$, and $H : \{0,1\}^n \to \{0,1\}^k$ be random oracles. Define the probablistic function $f_{OAEP} : \{0,1\}^n \to \{0,1\}^{n+k}$ as $f_{OAEP}(x;r) = \langle G(r) \oplus x, H(G(r) \oplus x) \oplus r) \rangle$, where $r \in_R \{0,1\}^k$. Let $\ell \leq k$, then f_{OAEP} is an adaptive 2ℓ-AONT, with security $q/2^{\ell-2}$ for an adversary that makes at most $q < 2^{\ell-1}$ adaptive queries to G or H.*

The proof is given in Lemma 25 from Appendix G.

4.2 Public-Key Proxy Re-Encryption

We begin with the traditional unidirectional proxy re-encryption (PRE) public-key encryption primitive. There is some variation in the definitions for PRE schemes, such as bidirectionality vs. unidirectionality and single-hop vs multi-hop. Since unidirectional multi-hop schemes are the most versatile, we use the definition from [32], as that is the only secure unidirectional multi-hop scheme that we know of. Our results apply to bidirectional and/or single-hop schemes as well, with the resulting scheme inheriting the properties of the underlying PRE scheme.

Definition 2 (Public-Key Proxy Re-Encryption). *A proxy public-key re-encryption primitive consists of five probabilistic polynomial time algorithms:*
$\mathsf{G}(\mathbf{1}^\lambda) \to (\mathsf{pk}, \mathsf{sk})$ *A key generation algorithm that given a security parameter generates a public- and secret-key pair.*
$\mathsf{E}(\mathsf{pk}, \mathbf{M}) \to \mathbf{C}$. *Public-key encryption—takes a public key and message and generates a ciphertext.*
$\mathsf{D}(\mathsf{sk}, \mathbf{C}) \to \mathbf{M}$. *Decryption—takes a secret key and ciphertext and returns the underlying message.*
$\mathsf{RG}(\mathsf{pk}_\mathbf{i}, \mathsf{sk}_\mathbf{i}, \mathsf{pk}_\mathbf{j}, \mathsf{sk}_\mathbf{j}) \to \mathsf{rk}_{\mathbf{i}\to\mathbf{j}}$. *Takes two public/secret key pairs—a source and a destination*[2]*—and creates a re-encryption key that can transform a ciphertext encrypted under the source's public key pk_i to one encrypted under the destination's public key pk_j.*
$\mathsf{RE}(\mathsf{rk}_{\mathbf{i}\to\mathbf{j}}), \mathbf{C}_\mathbf{i}) \to \mathbf{C}_\mathbf{j}$. *Takes a re-encryption key and a ciphertext encrypted under the re-encryption key's corresponding source public key, and translates it into a ciphertext under the destination's public key.*

[2] Many PRE schemes are "non-interactive": the destination secret key is not needed to produce the re-encryption key. We present the definition for an "interactive" scheme to match that of [32]. All of our results apply to non-interactive schemes as well. We note in our use cases the "interactive" definition does not require interaction amongst parties.

Correctness A proxy re-encryption scheme is correct if all encryptions and proxy re-encryptions decrypt properly. Formally, for every message M and every set of public/secret key pairs $\{(\mathsf{pk}_{i_u}, \mathsf{sk}_{i_u}) \leftarrow \mathsf{G}\}_{u \in \{0,\ldots,r\}}$ and re-encryption keys $\{\mathsf{rk}_{i_u \to i_{u+1}} \leftarrow \mathsf{RG}(\mathsf{sk}_{i_u}, \mathsf{pk}_{i_u}, \mathsf{pk}_{i_{u+1}})\}_{u \in \{0,\ldots,r-1\}}$, we have

$$\mathsf{D}\big(\mathsf{sk}_{i_r}, \mathsf{RE}\big(\mathsf{rk}_{i_{r-1} \to i_r}, \ldots \mathsf{RE}\big(\mathsf{rk}_{i_0 \to i_1}, \mathsf{E}\big(\mathsf{pk}_{i_0}, M\big)\big) \ldots\big)\big) = M.$$

Unidirectional, Multi-Hop, PRE CPA-Security This security notion establishes the basic concept of chosen-plaintext-attack (CPA) security for proxy re-encryption. The unidirectionality property tells us that the proxy can only be computed in one direction. The multi-hop property ensures that someone who has received a proxy re-encrypted ciphertext from an original source is able to again proxy re-encrypt to a new source, and this process can be repeated an unlimited number of times.

The security game allows the adversary to query public keys for which it has the corresponding secret key—in which case we say that the index of the public key is corrupted—and public keys for which it does not have the secret key—in which case the index is uncorrupted. The challenge ciphertext must be encrypted under a key with uncorrupted index. The adversary can query any re-encryption or re-encryption key that does not go from an uncorrupted to a corrupted index.

Definition 3 (Basic Unidirectional, Multi-Hop, PRE CPA-Security Game [3]). *Let λ be the security parameter. Let $\mathcal{A}$ be an oracle TM, representing the adversary. The* PRE-CPA *game consists of an execution of $\mathcal{A}$ in two phases, which are executed in order, as described in Fig. 1 (pg. 15).*

Within each phase, $\mathcal{A}$ has access to oracles (described below) and each can be queried in any order, poly(λ) times, unless otherwise specified.

Phase 1: *This phase consists of two oracles. On the ith query of either type the oracle computes $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{G}$ and then depending on the query:*

- **Uncorrupted Key Generation** $\mathcal{O}_{\mathsf{ukey}}$*: Output pk_i for $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{G}(1^\lambda)$. Index i is denoted as uncorrupted.*
- **Corrupted Key Generation** $\mathcal{O}_{\mathsf{ckey}}$*: Output $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{G}(1^\lambda)$. Index i is denoted as corrupted.*

Phase 2: *This phase consists of oracles producing re-encryption keys and re-encryptions of ciphertexts, as well as the challenge oracle. Note that the indices correspond to those of the keys produced in the first phase, and these oracles are based on state established in the first phase.*

- **Re-Encryption Key Generation** $\mathcal{O}_{\mathsf{rkey}}(i, j)$*: If $i = j$, or if i is uncorrupted and j is corrupted, then output $\perp$. Otherwise, output $\mathsf{rk}_{i \to j} \leftarrow \mathsf{RG}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_j, \mathsf{sk}_j)$.*
- **Re-Encryption** $\mathcal{O}_{\mathsf{renc}}(i, j, C)$*: If $i = j$, or if i is uncorrupted and j is corrupted, then output $\perp$. Otherwise, output $\mathsf{RE}(\mathsf{rk}_{i \to j}, C)$ where $\mathsf{rk}_{i \to j} \leftarrow \mathsf{RG}(\mathsf{sk}_i, \mathsf{pk}_i, \mathsf{pk}_j)$.*

- **Challenge** $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, i^*)$*: If* i^* *is corrupted, output* $\perp$*. Otherwise, output* $C^* \leftarrow \mathsf{E}(\mathsf{pk}_{i^*}, M_b)$*. The oracle can only be called once.*

Definition 4. *A Proxy Re-Encryption scheme* Π *is Unidirectional, Multi-Hop, PRE CPA-Secure if for all oracle P.P.T. adversaries A, there exists a negligible function* negl*:*

$$\Pr[\mathsf{PRE\text{-}CPA}_{A,\Pi}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

5 Revocation- and Key-Rotation-Friendly Proxy Re-Encryption

In this section we provide the definition of our new proxy re-encryption primitive, and our proposed construction.

5.1 $(1-\varepsilon)$-Revocable, Unidirectional, Multi-Hop, PRE CPA-Security

We modify the above security definition of traditional unidirectional, multi-hop PRE security to incorporate abilities that adversaries have in practice in the revocation and re-keying scenarios: initial access to files and their decryption keys, but a lack of inclination or capability to download all of these files. In particular, they may download the symmetric keys used in a file's hybrid encryption. The goal is now that after a file is re-encrypted the adversary cannot, at this point, decrypt the ciphertext.

The new definitions are a modification of the previous security game given in Defn. 3, differing in the definition of the challenge query. While in Defn. 3 the challenge ciphertext is produced as a fresh encryption under the challenge index, in this definition the challenge ciphertext can be produced from a series of re-encryptions. The challenge query contains a list of (possibly corrupted) indices $[i_0^*, \ldots, i_r^*]$ through which the message will be successively re-encrypted. The adversary can receive portions of these ciphertexts so long as the total size is sufficiently less than the size of each ciphertext (e.g., 90% of the size), representing the assumption that the adversary did not download the entire file prior to revocation. Then the challenge ciphertext is re-encrypted from i_r^* to an uncorrupted index j^*, representing the revocation of the adversary. At this point the adversary receives the entire challenge ciphertext encrypted under index j^*.

Definition 5 ($(1-\varepsilon)$-{Static, Adaptive}-Revocable, Unidirectional, Multi-Hop, PRE CPA-Security Game). *We define games* $(1-\varepsilon)$-$\mathsf{Stat\text{-}Revoke\text{-}PRE\text{-}CPA}_{A,\Pi}(\lambda)$ *and* $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}_{A,b}(\lambda)$ *as being identical to* $\mathsf{PRE\text{-}CPA}_{\mathcal{A},\Pi}(\lambda)$ *given in Defn. 3 except with the following change to the challenge query oracle in Phase 2:*

- **Challenge** $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, [i_0^*, \ldots, i_r^*], j^*, \mathsf{bitPos})$.

The adversary can call $(M_0,\ M_1,\ [i_0^*,\ldots,i_r^*],\ j^*,\ \mathsf{bitPos})$ *for* any *distinct values of* $[i_0^*,\ldots,i_r^*]$. *They represent the multiple hops, prior to the final hop, through which the challenge ciphertext is proxy re-encrypted. These keys may be corrupted, to model the fact that an adversary typically has decryption keys prior to revocation. However, now* j^* *must be an uncorrupted index distinct from each index in* $[i_0^*,\ldots,i_r^*]$. *The input* bitPos *will be used to indicate the bits of ciphertexts created prior to revocation that the adversary receives;* bitPos *differs in its use between the adaptive and static cases. In both the static and adaptive games the following are computed:*

- $\{C_u^*\}_{0\le u\le r}$ *where* $C_0^* = \mathsf{E}(\mathsf{pk}_{i_0^*}, M_b)$, *and for* $u > 0$, $C_u^* = \mathsf{RE}(\mathsf{rk}_{i_{u-1}^*\to i_u^*}, C_{u-1}^*)$
- $C^{**} = \mathsf{RE}(\mathsf{rk}_{i_r^*\to j^*}, C^*)$ *for* $\mathsf{rk}_{i_r^*\to j^*} = \mathsf{RG}(\mathsf{pk}_{i_r^*}, \mathsf{sk}_{i_r^*}, \mathsf{pk}_{j^*}, \mathsf{sk}_{j^*})$

Here each C_u^* *represents a ciphertext before revocation and* C^{**} *represents the ciphertext after revocation. Let* $N' = \max_{0\le u\le r} |C_u^*|$.

In the static game, $(1-\varepsilon)$-$\mathsf{Stat\text{-}Revoke\text{-}PRE\text{-}CPA}$, *the adversary provides* bitPos *which consists of* $(1-\varepsilon)N'$ *pairs* (u, v) *for* $0 \le u \le r$ *and* $0 \le v < C_u^*$. *The output of the query is* C^{**} *and the* v*th bit of* C_u^* *for each pair* (u, v) *specified by* bitPos. *This challenge oracle can only successfully be called once.*

In the adaptive game, $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}$, *the Challenge oracle is stateful. The adversary selects* bitPos *one pair* (u, v) *at a time and receives the* v*th bit of ciphertext* C_u^*, *so it can choose each pair based on the previous bits it received. Once the adversary has received* $(1-\varepsilon)N'$ *total bits of* $\{C_u^*\}_{0\le u\le r}$, *the oracle outputs* C^{**}. *After this it refuses to respond. Similarly, the oracle refuses to respond if queries change any of the calling values other than* bitPos.

Definition 6. *A Proxy Re-Encryption scheme* Π *is* $(1-\varepsilon)$-$\{$*Static, Adaptive*$\}$-*Revocable, Unidirectional, Multi-Hop, PRE CPA-Secure if for all oracle P.P.T. adversaries* $\mathcal{A}$, *there exists a negligible function* negl *s.t. both hold:*

1. $\Pr[(1-\varepsilon)\text{-}\{\mathsf{Stat},\mathsf{Adap}\}\text{-}\mathsf{Revoke\text{-}PRE\text{-}CPA}_{\mathcal{A},\Pi}(\lambda) = 1] \le \frac{1}{2} + \mathsf{negl}(\lambda)$
2. $\Pr[\mathsf{PRE\text{-}CPA}_{\mathcal{A},\Pi}(\lambda) = 1] \le \frac{1}{2} + \mathsf{negl}(\lambda)$.

Note that we need the scheme to satisfy both the traditional definition and the revocable definition, as it is possible to construct revocation schemes that produces secure re-keyed ciphertexts, but where the originals are insecure.

6 CPA-Secure Hybrid Public-Key Proxy Re-Encryption Scheme

We now give a hybrid construction of a CPA-secure public-key proxy re-encryption scheme $\Pi_{hyb} = \left(\mathsf{G}^{\mathsf{Hyb}}, \mathsf{E}^{\mathsf{Hyb}}, \mathsf{D}^{\mathsf{Hyb}}, \mathsf{RG}^{\mathsf{Hyb}}, \mathsf{RE}^{\mathsf{Hyb}}\right)$, assuming the existence of a public-key proxy re-encryption scheme $\Pi = (\mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$, a symmetric-key encryption scheme $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$, and an AONT T.

```
$\mathsf{PRE\text{-}CPA}_{\mathcal{A},\Pi}(\lambda)$

$\sigma \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ukey}},\mathcal{O}_{\mathsf{ckey}}}(\lambda)$          $\triangleright$ (Phase 1)
$b \leftarrow \{0,1\}$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{rkey}},\mathcal{O}_{\mathsf{renc}},\mathcal{O}_{\mathsf{chal}}}(\sigma)$   $\triangleright$ (Phase 2)
Output 1 iff $b = b'$
```

Fig. 1. Proxy Re-Encryption CPA-Security Experiment

```
$\mathsf{E}^{\mathsf{Hyb}}(\mathsf{pk}, M)$

$k_0 \leftarrow \mathsf{G}^{\mathsf{Sym}}(1^\lambda)$
$C^{pk} \leftarrow \mathsf{E}(\mathsf{pk}, k_0)$
$C^T \leftarrow T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M)\big)$
return $C = \big(C^{pk}, [\,], C^T\big)$
```

Fig. 2. $\mathsf{E}^{\mathsf{Hyb}}$ Encryption

```
$\mathsf{D}^{\mathsf{Hyb}}(\mathsf{sk},\, C = (C^{pk}, \big[C_1^{bks}, \ldots, C_r^{bks}\big],$
        $C_r^T = C_{r,1}^T \cdots C_{r,N}^T))$

for $u \leftarrow r, \ldots, 1$ do
    $(s_u, k_u) \leftarrow \mathsf{D}\big(\mathsf{sk}, C_u^{bks}\big)$
    $\mathsf{ind}_u \leftarrow \mathsf{Ind}(s_u, \ell^*)$
    $\mathsf{str}_u \leftarrow \mathsf{Ctr}(k_u, \ell^*)$
    for $v \leftarrow 1, \ldots, N$ do
        if $v \in \mathsf{ind}_u$ then
            $C_{u-1,v}^T \leftarrow C_{u,v}^T \oplus \mathsf{str}[v]$
        else
            $C_{u-1,v}^T \leftarrow C_{u,v}^T$
        end if
    end for
end for
$C^T \leftarrow C_{0,1}^T \cdots C_{0,N}^T$
$k_0 \leftarrow \mathsf{D}\big(\mathsf{sk}, C^{pk}\big)$
return $M \leftarrow \mathsf{D}^{\mathsf{Sym}}\big(k_0, T^{-1}(C^T)\big)$
```

Fig. 3. $\mathsf{D}^{\mathsf{Hyb}}$ Decryption

```
$\mathsf{RE}^{\mathsf{Hyb}}(\mathsf{rk}_{i\to j}{=}\big(\mathsf{pk}_j, \mathsf{rk}'_{i\to j}\big),\;\; C{=}(C^{pk},$
        $\big[C_1^{bks}, \ldots, C_r^{bks}\big], C^T{=}C_1^T \cdots C_N^T))$

$\widetilde{C^{pk}} \leftarrow \mathsf{RE}\big(\mathsf{rk}'_{i\to j}, C^{pk}\big)$
for $u \leftarrow 1, \ldots, r$ do
    $\widetilde{C_u^{bks}} \leftarrow \mathsf{RE}\big(rk'_{i\to j}, C_u^{bks}\big)$
end for
Choose $s_{r+1}, k_{r+1}$ uniformly.
$C_{r+1}^{bks} \leftarrow \mathsf{E}\big(\mathsf{pk}_j, (s_{r+1}, k_{r+1})\big)$
$\mathsf{ind}_{r+1} \leftarrow \mathsf{Ind}(s_{r+1}, \ell^*)$
$\mathsf{str}_{r+1} \leftarrow \mathsf{Ctr}(k_{r+1}, \ell^*)$
for $v \leftarrow 1, \ldots, N$ do
    if $v \in \mathsf{ind}_{r+1}$ then
        $\widetilde{C_v^T} \leftarrow C_v^T \oplus \mathsf{str}_{r+1}[v]$
    else
        $\widetilde{C_v^T} \leftarrow C_v^T$
    end if
end for
$\widetilde{C^T} \leftarrow \widetilde{C_1^T} \cdots \widetilde{C_N^T}$
return $C = (\widetilde{C^{pk}},\, [\widetilde{C_1^{bks}},\, \ldots,\, \widetilde{C_{r+1}^{bks}}],$
        $\widetilde{C^T})$
```

Fig. 4. $\mathsf{RE}^{\mathsf{Hyb}}$ Re-Encryption

```
$\mathsf{Bounded\text{-}Storage\text{-}Reconst}_{\mathcal{A},\Pi}(f, n)$

$\sigma \leftarrow \mathcal{A}(f)$ for $|\sigma| \le n$
$\widetilde{f} \leftarrow \Pi(f)$
$f' \leftarrow \mathcal{A}(\sigma, \widetilde{f})$
Output 1 iff $f = f'$
```

Fig. 5. Bounded Storage Reconstruction Security Game

6.1 Overview of Construction

The basis of our construction is a standard hybrid encryption scheme with an AONT applied to the symmetric ciphertext portion of the hybrid ciphertext. That is, an initial ciphertext has the form (C^{pk}, C^T), where the components are:

- $C^{pk} = \mathsf{E}(\mathsf{pk}, k_0)$ is an encryption under the user's public key pk of the symmetric key k_0.
- $C^T = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M)\big)$ is the AONT applied to a symmetric-key encryption of the message M.

Now, for each proxy re-encryption, the proxy re-encrypts a randomly selected set of bits of C^T, on top of any previous re-encryptions of bits of C^T. This makes inverting the AONT impossible unless the adversary was lucky enough to have previously queried and stored all of the encrypted bits, and since they are (pseudo-)randomly distributed this is incredibly unlikely. However, to allow decryption, the proxy needs to store the locations of the re-encrypted bits and the key used to encrypt them. This is done by producing a new public-key encryption of the seed used to select the positions and encrypt the bits, and adding this to the ciphertext. As a result, the ciphertext size and encryption time grow additively with the number of re-encryptions, where the summand is the size of a proxy ciphertext.

A ciphertext that is a re-encryption of a previous ciphertext has the form $\big(C^{pk}, \big[C_1^{bks}, \ldots, C_r^{bks}\big], C^T\big)$, where the components are:

- $C^{pk} = \mathsf{E}(\mathsf{pk}, k_0)$ is an encryption under the user's public key pk of the symmetric key k_0.
- Each of $C_1^{bks}, \ldots, C_r^{bks}$ is an encryption under the user's public key of the bit positions and key for a previous re-encryption; there is one for each previous re-encryption.
- C^T is $T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M)\big)$ with some bits re-encrypted, as determined by $\big[C_1^{bks}, \ldots, C_r^{bks}\big]$.

To keep the notation consistent, we write an initial ciphertext as $\big(C^{pk}, [\,], C^T\big)$.

6.2 Notation

We let r be the number of previous re-encryptions of a hybrid ciphertext. We let N denote the output length of T, and let $\ell^* \leq N$ with $\ell^* = \omega(\log(\lambda))$ be the number of bits of the AONT output that are re-encrypted. The value of ℓ^* will depend on the security of the AONT and assumptions about the behavior of the adversary.

We use a deterministic function $\mathsf{Ind}(s, \ell^*)$ that takes a seed s and produces a pseudorandom element of $\left\{ {N \atop \ell^*} \right\}$, i.e. a pseudorandom subset of $\{1, \ldots, N\}$ of size ℓ^*. We use $\mathsf{Ctr}(k, \ell^*)$ to denote the keystream of length ℓ^* produced by pseudorandom generator (e.g., here our notation envisions using counter mode encryption with key k and nonce 0, which is a known PRG). Note that if the underlying block cipher is secure, then $\mathsf{Ctr}(k, \ell^*)$ is pseudorandom. We XOR

$\mathsf{Ctr}(k, \ell^*)$ with the bits of the AONT output that we want re-encrypted. For a keystream str, we let $\mathsf{str}[j]$ represent the jth bit of str that has not yet been XORed with any bit of the AONT output.

We let $\mathsf{rInd}(\ell^*)$ denote a random element of $\left\{{N \atop \ell^*}\right\}$, i.e. a random subset of $\{1, \ldots, N\}$ of size ℓ^*; and, $\mathsf{rStr}(\ell^*)$ is a random string of length ℓ^*. Let $[T(\cdot)]_{\mathsf{ind},\mathsf{str}}$ represent a modified output of T—specifically, the values of bit positions specified by the indices in ind are XORed with keystream str. For example, $T(x)_{\mathsf{ind}=1,3,4,\mathsf{str}=101}$ would output $t_1 \oplus 1, t_2, t_3 \oplus 0, t_4 \oplus 1, t_5, \ldots.$, where t_i is the ith output bit of $T(x)$. Finally, $C_{(1-\varepsilon)}$ represents the fraction $1 - \varepsilon$ of the bits of ciphertext C that the adversary receives;

6.3 Proxy Re-Encryption Construction

Our proxy re-encryption scheme is the five-tuple $(\mathsf{G}^{\mathsf{Hyb}}, \mathsf{E}^{\mathsf{Hyb}}, \mathsf{D}^{\mathsf{Hyb}}, \mathsf{RG}^{\mathsf{Hyb}}, \mathsf{RE}^{\mathsf{Hyb}})$, where $\mathsf{G}^{\mathsf{Hyb}}(1^\lambda) = \mathsf{G}(1^\lambda)$, $\mathsf{E}^{\mathsf{Hyb}}$ is defined in Fig. 2 (pg. 15), $\mathsf{D}^{\mathsf{Hyb}}$ is defined in Fig. 3 (pg. 15), $\mathsf{RG}^{\mathsf{Hyb}}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_j, \mathsf{sk}_j) = (\mathsf{pk}_j, \mathsf{RG}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_j, \mathsf{sk}_j)) = \mathsf{rk}_{i \to j}$, and $\mathsf{RE}^{\mathsf{Hyb}}$ is defined in Fig. 4 (pg. 15).

6.4 Correctness

A valid ciphertext for message M under public key pk with corresponding secret key sk has the form $C = \left(C^{pk}, \left[C_1^{bks}, \ldots, C_r^{bks}\right], C_r^T\right)$ where

- C^{pk} is an encryption under pk of symmetric key k_0
- Each C_u^{bks} is an encryption under pk of (s_u, k_u)
- C_r^T is $T\big(\mathsf{E}^{\mathsf{Sym}}(k, M)$ with the bits at positions $\mathsf{Ind}(s_1)$ encrypted under symmetric key k_1, then the bits at positions $\mathsf{Ind}(s_2, \ell^*)$ encrypted under symmetric key k_2, etc.

The decryption algorithm computes $k_0 = \mathsf{D}(\mathsf{sk}, C^{pk})$ and each $(s_u, k_u) = \mathsf{D}(\mathsf{sk}, C_u^{bks})$; computes C_0^T by decrypting the bits of C_r^T at positions $\mathsf{Ind}(s_r, \ell^*)$ using symmetric key k_r to produce C_{r-1}^T, then decrypting the bits of C_{r-1}^T at positions $\mathsf{Ind}(s_{r-1}, \ell^*)$ using symmetric key k_{r-1} to produce C_{r-2}^T, etc., eventually reaching C_0^T; and finally computes $M = \mathsf{D}^{\mathsf{Sym}}\big(k_0, T^{-1}(C^T)\big)$.

6.5 Basic PRE-CPA Security

We first provide the basic Proxy Re-Encryption CPA security of the scheme (Defn. 4). We note that this is necessary because $(1-\varepsilon)$-$\{\mathsf{Stat}, \mathsf{Adap}\}$-$\mathsf{Revoke}$-$\mathsf{PRE}$-$\mathsf{CPA}$ security game (Defn. 5) provides the adversary the decryption key and oracle access to earlier versions of the ciphertext (pre re-encryption), and therefore traditional security is not implied by the new definition; rather, the new definition provides an additional proxy re-encryption security property.

Theorem 1. *Assume the existence of a CPA-secure public-key proxy re-encryption scheme* $\Pi = (\mathsf{G}, \mathsf{RG}, \mathsf{E}, \mathsf{RE}, \mathsf{D})$*, a CPA-secure symmetric-key encryption scheme* $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$*, and an AONT* T*. Then the construction in Section 6.3 is CPA-secure.*

This is a basic hybrid security proof, which is provided in Appendix B.

6.6 Adaptive-Revocable-PRE-CPA Security

We now provide a proof of $(1-\varepsilon)$-revocable security for the adaptive case, showing that it satisfies Definition 6. We require a minor additional property of the underlying PRE scheme, which we call *re-encryption history independence*, which says that the distribution of a re-encrypted ciphertext does not depend on the keys used in encryption and re-encryption prior to the current key (though it may depend on the number of previous re-encryptions):

Definition 7 (Re-Encryption History Independence). *A public-key proxy re-encryption scheme $\Pi = (\mathsf{G}, \mathsf{RG}, \mathsf{E}, \mathsf{RE}, \mathsf{D})$ has re-encryption history independence if for every set of public/secret key pairs $\{(\mathsf{pk}_0, \mathsf{sk}_0), (\mathsf{pk}'_0, \mathsf{sk}'_0), \ldots, (\mathsf{pk}_{r-1}, \mathsf{sk}_{r-1}), (\mathsf{pk}'_{r-1}, \mathsf{sk}'_{r-1}), (\mathsf{pk}_r, \mathsf{sk}_r)\}$ with re-encryption keys $\mathsf{rk}_{u\to u+1} \leftarrow \mathsf{RG}(\mathsf{pk}_u, \mathsf{sk}_u, \mathsf{pk}_{u+1}, \mathsf{sk}_{u+1})$, $\mathsf{rk}'_{u\to u+1} \leftarrow \mathsf{RG}(\mathsf{pk}'_u, \mathsf{sk}'_u, \mathsf{pk}'_{u+1}, \mathsf{sk}'_{u+1})$ for $u \in [0, \ldots, r-2]$ and $\mathsf{rk}_{r-1\to r} \leftarrow \mathsf{RG}(\mathsf{pk}_{r-1}, \mathsf{sk}_{r-1}, \mathsf{pk}_r, \mathsf{sk}_r)$, $\mathsf{rk}'_{r-1\to r} \leftarrow \mathsf{RG}(\mathsf{pk}'_{r-1}, \mathsf{sk}'_{r-1}, \mathsf{pk}_r, \mathsf{sk}_r)$ and every message M:*

$$\mathsf{RE}(\mathsf{rk}_{r-1\to r}, \ldots \mathsf{RE}(\mathsf{rk}_{0\to 1}, \mathsf{E}(\mathsf{pk}_0, M)) \ldots) \approx \mathsf{RE}(\mathsf{rk}'_{r-1\to r}, \ldots \mathsf{RE}(\mathsf{rk}'_{0\to 1}, \mathsf{E}(\mathsf{pk}'_0, M)) \ldots)$$

where $\approx$ denotes computationally indistinguishable distributions.

Although PRE schemes do not need to have this property to be $\mathsf{PRE\text{-}CPA}$-secure, it is a natural property to have. It does follow from re-encryption key privacy, an additional security property found in the schemes of [3,2,32]. Every PRE scheme we looked at [4,13,19,26,3,2,32] has re-encryption history independence.

Theorem 2. *Assume the existence of a* $\mathsf{PRE\text{-}CPA}$*-secure public-key proxy re-encryption scheme $\Pi = (\mathsf{G}, \mathsf{RG}, \mathsf{E}, \mathsf{RE}, \mathsf{D})$ with re-encryption history independence*[3]*, a symmetric-key encryption scheme $\Pi_{sym} = (\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}})$, and an adaptive ℓ-AONT T. Suppose that for the construction of Π_{hyb} from Section 6.3, C^T comprises at least a fraction $1-\delta$ of the total size of each ciphertext. Then for any $\varepsilon < 1$ with $\varepsilon > \delta$ and any $\ell^* > \frac{\ell}{\varepsilon-\delta}$, this construction is* $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}$*-secure.*

Hybrid Argument We show the computational indistinguishability of a series of games. Each game is the same as the real game except in regards to C^{**}. In each game, the challenge query is $(M_0, M_1, [i^*_0, \ldots, i^*_r], j^*, \mathsf{bitPos})$. We highlight in bold the portions of C^{**} that differ from the previous games, in the descriptions of the respective games that follow:

[3] If the scheme is otherwise secure but lacks re-encryption history independence, this theorem can be proven under a weaker version of $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}$ security—the list of identities $[i^*_0, \ldots, i^*_r]$ in the challenge query has the additional requirement that for $u \in [0, \ldots, r_1]$, if i^*_u is uncorrupted then so is i^*_{u+1}.

Game_0*:* This is the real game, where:

$$C^{**} = \Big(\mathsf{RE}\big(\mathsf{rk}_{i_r^* \to j^*}, C^{pk}\big), \big[C_1^{bks}, \dots, C_r^{bks}, \mathsf{E}\big(\mathsf{pk}_{j^*}, (s, k_1)\big)\big], \big[C^T\big]_{\mathsf{Ind}(s,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\Big) \ .$$

Game_1*:* This is identical to Game_0 except that we replace $\mathsf{E}\big(\mathsf{pk}_{j^*}, (s, k_1)\big)$ with $\mathsf{E}\big(\mathsf{pk}_{j^*}, (0, 0)\big)$, resulting in:

$$C^{**} = \Big(\mathsf{RE}\big(\mathsf{rk}_{i_r^* \to j^*}, C^{pk}\big), \big[C_1^{bks}, \dots, C_r^{bks}, \boldsymbol{\mathsf{E}\big(\mathsf{pk}_{j^*}, (0, 0)\big)}\big], \big[C^T\big]_{\mathsf{Ind}(s,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\Big) \ .$$

Game_2*:* This is identical to Game_1 except that we replace the pseudorandom $\mathsf{Ind}(s, \ell^*)$ with truly random $\mathsf{rInd}(\ell^*)$, resulting in:

$$C^{**} = \Big(\mathsf{RE}\big(\mathsf{rk}_{i_r^* \to j^*}, C^{pk}\big), \big[C_1^{bks}, \dots, C_r^{bks}, \mathsf{E}\big(\mathsf{pk}_{j^*}, (0, 0)\big)\big], \big[C^T\big]_{\boldsymbol{\mathsf{rInd}(\ell^*)},\mathsf{Ctr}(k_1,\ell^*)}\Big) \ .$$

Game_3*:* This is identical to Game_2 except that we replace the keystream $\mathsf{Ctr}(k_1, \ell^*)$ we get from counter mode encryption with a random string $\mathsf{rStr}(\ell^*)$, resulting in:

$$C^{**} = \Big(\mathsf{RE}\big(\mathsf{rk}_{i_r^* \to j^*}, C^{pk}\big), \big[C_1^{bks}, \dots, C_r^{bks}, \mathsf{E}\big(\mathsf{pk}_{j^*}, (0, 0)\big)\big], \big[C^T\big]_{\mathsf{rInd}(\ell^*),\boldsymbol{\mathsf{rStr}(\ell^*)}}\Big) \ .$$

We now provide a series of lemmas that show that any adverary's probabilities of success in two successive games are negligibly close. These are presented in Lemmas 2, 3, and 4. Finally, we show in Lemma 6 that any adversary's chance of success in the final game is negligibly close to 1/2.

Lemma 2. *Suppose that an adversary $\mathcal{A}_1$ for the $(1-\varepsilon)$-*$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}$ *security game has probability of success p_0 in* Game_0 *and probability of success p_1 in* Game_1*. If the underlying proxy re-encryption scheme Π is* $\mathsf{PRE\text{-}CPA}$*-secure, then $p_1 - p_0$ is negligible.*

Proof. We construct an adversary $\mathcal{B}_1$ that plays the $\mathsf{PRE\text{-}CPA}$ security game by simulating adversary $\mathcal{A}_1$. $\mathcal{B}_1$ gives $\mathcal{A}_1$ the public parameters it receives. $\mathcal{B}_1$ responds to queries from $\mathcal{A}_1$ as follows:

- Uncorrupted Key Generation query $\mathcal{O}_{\mathsf{ukey}}$: $\mathcal{B}_1$ makes an uncorrupted key generation query to its oracle and receives pk_i in response. $\mathcal{B}_1$ sends pk_i to $\mathcal{A}_1$.
- Corrupted Key Generation query $\mathcal{O}_{\mathsf{ckey}}$: $\mathcal{B}_1$ makes a corrupted key generation query to its oracle and receives $(\mathsf{pk}_i, \mathsf{sk}_i)$ in response. $\mathcal{B}_1$ sends $(\mathsf{pk}_i, \mathsf{sk}_i)$ to $\mathcal{A}_1$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(i, j)$: $\mathcal{B}_1$ queries (i, j) to its re-encryption key generation oracle, receiving $\mathsf{rk}'_{i \to j}$ in response. Note that $\mathcal{B}_1$ will have pk_i from a previous key generation query. $\mathcal{B}_1$ sends $\mathsf{rk}_{i \to j} = \big(\mathsf{pk}_j, \mathsf{rk}'_{i \to j}\big)$ to $\mathcal{A}_1$.
- Re-Encryption query $\mathcal{O}_{\mathsf{renc}}(i, j, C)$: $\mathcal{B}_1$ queries (i, j) to its re-encryption key generation oracle, receiving $\mathsf{rk}'_{i \to j}$ in response. $\mathcal{B}_1$ then runs $\mathsf{RE}^{\mathsf{Hyb}}\big((\mathsf{pk}_j, \mathsf{rk}'_{i \to j}), C\big)$ from Algorithm 4 and sends the result to $\mathcal{A}_1$.

- Challenge query $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, [i^*_0, \ldots, i^*_r], j^*)$: $\mathcal{B}_1$ generates random bit b, random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_1$ also makes r corrupted key generation queries, receiving $\{(\mathsf{pk}_{i'_u}, \mathsf{sk}_{i'_u})\}_{0 \le u \le r-1}$ in response. $\mathcal{B}_1$ creates $C^{pk}_0 = \mathsf{E}\Big(\mathsf{pk}_{i^*_0}, k_0\Big)$ and $\hat{C}^{pk}_0 = \mathsf{E}\Big(\mathsf{pk}_{i'_0}, k_0\Big)$, and then queries

  $$C^{pk}_\alpha = \mathcal{O}_{\mathsf{renc}}\Big(i'_{\alpha-1}, i^*_\alpha, \hat{C}^{pk}_{\alpha-1}\Big), \qquad \hat{C}^{pk}_\alpha = \mathcal{O}_{\mathsf{renc}}\Big(i'_{\alpha-1}, i'_\alpha, \hat{C}^{pk}_{\alpha-1}\Big),$$

  where $1 \le \alpha \le r$ and $C^{pk}_{r+1} = \mathcal{O}_{\mathsf{renc}}\Big(i'_r, j^*, \hat{C}^{pk}_r\Big)$.

  Note that each C^{pk}_u is produced by encrypting k_0 under $\mathsf{pk}_{i'_0}$ and then re-encrypting it through $[\mathsf{pk}_{i'_1}, \mathsf{pk}_{i'_{u-1}}, \mathsf{pk}_{i^*_u}]$. By re-encryption history independence, this is indistinguishable from a ciphertext produced by encrypting k_0 under $\mathsf{pk}_{i^*_0}$ and then re-encrypting it through $[\mathsf{pk}_{i^*_1}, \mathsf{pk}_{i^*_{u-1}}, \mathsf{pk}_{i^*_u}]$.

  For each $u \in \{1, \ldots, r\}$, $\mathcal{B}_1$ creates $C^{bks}_{u,u} = \mathsf{E}\Big(\mathsf{pk}_{i^*_u}, (s_u, k_u)\Big)$ and $\hat{C}^{bks}_{u,u} = \mathsf{E}\Big(\mathsf{pk}_{i'_u}, (s_u, k_u)\Big)$, and then queries

  $$\begin{aligned}
  C^{bks}_{u,u+1} &= \mathcal{O}_{\mathsf{renc}}\Big(i'_u, i^*_{u+1}, \hat{C}^{bks}_{u,u}\Big), & \hat{C}^{bks}_{u,u+1} &= \mathcal{O}_{\mathsf{renc}}\Big(i'_u, i'_{u+1}, \hat{C}^{bks}_{u,u}\Big),\\
  C^{bks}_{u,u+2} &= \mathcal{O}_{\mathsf{renc}}\Big(i'_{u+1}, i^*_{u+2}, \hat{C}^{bks}_{u,u+1}\Big), & \hat{C}^{bks}_{u,u+2} &= \mathcal{O}_{\mathsf{renc}}\Big(i'_{u+1}, i'_{u+2}, \hat{C}^{bks}_{u,u+1}\Big), \ \ldots\\
  C^{bks}_{u,r} &= \mathcal{O}_{\mathsf{renc}}\Big(i'_{r-1}, i^*_r, \hat{C}^{bks}_{u,r-1}\Big), & \hat{C}^{bks}_{u,r} &= \mathcal{O}_{\mathsf{renc}}\Big(i'_{r-1}, i'_r, \hat{C}^{bks}_{u,r-1}\Big),\\
  C^{bks}_{u,r+1} &= \mathcal{O}_{\mathsf{renc}}\Big(i'_r, j^*, \hat{C}^{bks}_{u,r}\Big)
  \end{aligned}$$

  Note that each $C^{bks}_{u,v}$ is produced by encrypting (s_u, k_u) under $\mathsf{pk}_{i'_u}$ and then re-encrypting it through $[\mathsf{pk}_{i'_u}, \mathsf{pk}_{i'_{v-1}}, \mathsf{pk}_{i^*_v}]$. By re-encryption history independence, this is indistinguishable from a ciphertext produced by encrypting (s_u, k_u) under $\mathsf{pk}_{i^*_u}$ and then re-encrypting it through $[\mathsf{pk}_{i^*_u}, \mathsf{pk}_{i^*_{v-1}}, \mathsf{pk}_{i^*_v}]$.

  Then $\mathcal{B}_1$ creates $C^T_0 = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)$ and $C^*_0 = \Big(C^{pk}_0, [\,], C^T_0\Big)$. For each $u \in \{1, \ldots, r\}$, $\mathcal{B}_1$ creates $C^T_u = \big[C^T_{u-1}\big]_{\mathsf{Ind}(s_u, \ell^*), \mathsf{Ctr}(k_u, \ell^*)}$ and $C^*_u = \big(C^{pk}_u, \big[C^{bks}_{1,u}, \ldots, C^{bks}_{u,u}\big], C^T_u\big)$. $\mathcal{B}_1$ makes challenge query $((s_{r+1}, k_{r+1}), (0,0), j^*)$, receiving ciphertext C' in response. It gives $\mathcal{A}_1$ the $(1-\varepsilon)|C^*_r|$ bits that it adaptively requests of $\{C^*_u\}_{0 \le u \le r}$ as well as

  $$C^{**} = \Big(C^{pk}_{r+1}, \big[C^{bks}_{1,r+1}, \ldots, C^{bks}_{r,r+1}, C'\big], \big[C^T_r\big]_{\mathsf{Ind}(s_{r+1}, \ell^*), \mathsf{Ctr}(k_{r+1}, \ell^*)}\Big) \ .$$

- Guess b': If $b = b'$ then $\mathcal{B}_1$ guesses that C' is an encryption of (s_{r+1}, k_{r+1}), otherwise $\mathcal{B}_1$ guesses that C' is an encryption of $(0,0)$.

If C' is an encryption of (s_{r+1}, k_{r+1}), then $\mathcal{A}_1$ is in Game_0 (the real game) and has probability of success p_0; thus $\mathcal{B}_1$ is correct with probability p_0. If C'

is an encryption of $(0, 0)$ then $\mathcal{A}_1$ is in Game_1 and has probability of success p_1; thus $\mathcal{B}_1$ is correct with probability $1 - p_1$. Therefore $\mathcal{B}_1$'s probability of success is $\frac{1}{2}(p_0 + 1 - p_1) = \frac{1}{2} + \frac{1}{2}(p_0 - p_1)$. By the $\mathsf{PRE\text{-}CPA}$ security of $(\mathsf{G}, \mathsf{RG}, \mathsf{E}, \mathsf{RE}, \mathsf{D})$, $\frac{1}{2}(p_0 - p_1)$ is negligible, and so $p_0 - p_1$ is negligible. □

Lemma 3. *Suppose that an adversary $\mathcal{A}_2$ for the $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}$ security game has probability of success p_1 in Game_1 and probability of success p_2 in Game_2. If $\mathsf{Ind}(s, \ell^*)$ with random seed s is pseudorandom (indistinguishable from $\mathsf{rInd}(\ell^*)$), then $p_2 - p_1$ is negligible.*

Proof. We construct a distinguisher $\mathcal{D}_1$ that receives a set of indices ind—either $\mathsf{ind} = \mathsf{Ind}(s, \ell^*)$ with random seed s or $\mathsf{ind} = \mathsf{rInd}(\ell^*)$. $\mathcal{D}_1$ simulates adversary $\mathcal{A}_2$. $\mathcal{D}_1$ instantiates the PRE itself and sends $\mathcal{A}_2$ the public parameters. $\mathcal{D}_1$ responds to queries from $\mathcal{A}_2$ as follows:

- Uncorrupted Key Generation query $\mathcal{O}_{\mathsf{ukey}}$: $\mathcal{D}_1$ creates a key pair $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{G}$ and sends pk_i to $\mathcal{A}_2$.
- Corrupted Key Generation query $\mathcal{O}_{\mathsf{ckey}}$: $\mathcal{D}_1$ creates a key pair $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{G}$ and sends $(\mathsf{pk}_i, \mathsf{sk}_i)$ to $\mathcal{A}_2$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(i, j)$: If $i = j$ or if i is uncorrupted and j is corrupted then $\mathcal{D}_1$ sends $\perp$; otherwise, $\mathcal{D}_1$ computes $\mathsf{rk}'_{i\to j} \leftarrow \mathsf{RG}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_j, \mathsf{sk}_j)$. $\mathcal{D}_1$ sends $\mathsf{rk}_{i\to j} = \big(\mathsf{pk}_j, \mathsf{rk}'_{i\to j}\big)$ to $\mathcal{A}_2$.
- Re-Encryption query $\mathcal{O}_{\mathsf{renc}}(i, j, C)$: If $i = j$ or if i is uncorrupted and j is corrupted then $\mathcal{D}_1$ sends $\perp$; otherwise, $\mathcal{D}_1$ computes $\mathsf{rk}'_{i\to j} \leftarrow \mathsf{RG}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_j, \mathsf{sk}_j)$. $\mathcal{D}_1$ then runs $\mathsf{RE}^{\mathsf{Hyb}}\big((\mathsf{pk}_j, \mathsf{rk}'_{i\to j}), C\big)$ from Algorithm 4 and sends the result to $\mathcal{A}_2$.
- Challenge query $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, [i^*_0, \ldots, i^*_r], j^*)$: $\mathcal{D}_1$ generates random bit b, random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{D}_1$ creates $C^{pk}_0 = \mathsf{E}\Big(\mathsf{pk}_{i^*_0}, k_0\Big)$ and

$$C^{pk}_1 = \mathsf{RE}\Big(\mathsf{rk}_{i^*_0\to i^*_1}, C^{pk}_0\Big), C^{pk}_2 = \mathsf{RE}\Big(\mathsf{rk}_{i^*_1\to i^*_2}, C^{pk}_1\Big),$$
$$\ldots, C^{pk}_r = \mathsf{RE}\Big(\mathsf{rk}_{i^*_{r-1}\to i^*_r}, C^{pk}_{r-1}\Big), C^{pk}_{r+1} = \mathsf{RE}\big(\mathsf{rk}_{i^*_r\to j^*}, C^{pk}_r\big)$$

For each $u \in \{1, \ldots, r\}$, $\mathcal{D}_1$ creates $C^{bks}_{u,u} = \mathsf{E}\Big(\mathsf{pk}_{i^*_u}, (s_u, k_u)\Big)$ and computes

$$C^{bks}_{u,u+1} = \mathsf{RE}\Big(\mathsf{rk}_{i^*_u\to i^*_{u+1}}, C^{bks}_{u,u}\Big), C^{bks}_{u,u+2} = \mathsf{RE}\Big(\mathsf{rk}_{i^*_{u+1}\to i^*_{u+2}}, C^{bks}_{u,u+1}\Big),$$
$$\ldots, C^{bks}_{u,r} = \mathsf{RE}\Big(\mathsf{rk}_{i^*_{r-1}\to i^*_r}, C^{bks}_{u,r-1}\Big), C^{bks}_{u,r+1} = \mathsf{RE}\big(\mathsf{rk}_{i^*_r\to j^*}, C^{bks}_{u,r}\big)$$

Then $\mathcal{D}_1$ creates $C^T_0 = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)$ and $C^*_0 = \Big(C^{pk}_0, [\,], C^T_0\Big)$. For each $u \in \{1, \ldots, r\}$, $\mathcal{D}_1$ creates $C^T_u = \big[C^T_{u-1}\big]_{\mathsf{Ind}(s_u,\ell^*),\mathsf{Ctr}(k_u,\ell^*)}$ and $C^*_u = \big(C^{pk}_u, \big[C^{bks}_{1,u}, \ldots, C^{bks}_{u,u}\big], C^T_u\big)$.
$\mathcal{D}_1$ gives $\mathcal{A}_2$ the $(1-\varepsilon)|C^*_r|$ bits that it adaptively requests of $\{C^*_u\}_{0\le u\le r}$ as well as

$$C^{**} = \Big(C^{pk}_{r+1}, \big[C^{bks}_{1,r+1}, \ldots, C^{bks}_{r,r+1}, \mathsf{E}\big(\mathsf{pk}_{j^*}, (0,0)\big)\big], \big[C^T_r\big]_{\mathsf{ind},\mathsf{Ctr}(k_{r+1},\ell^*)}\Big)\ .$$

- Guess b': If $b = b'$ then $\mathcal{D}_1$ guesses that $\mathsf{ind} = \mathsf{Ind}(s, \ell^*)$, otherwise $\mathcal{D}_1$ guesses that $\mathsf{ind} = \mathsf{rInd}(\ell^*)$.

If $\mathsf{ind} = \mathsf{Ind}(s, \ell^*)$, then $\mathcal{A}_2$ is in Game_1 and has probability of success p_1; thus $\mathcal{D}_1$ is correct with probability p_1. If $\mathsf{ind} = \mathsf{rInd}(\ell^*)$, then $\mathcal{A}_2$ is in Game_2 and has probability of success p_2; thus $\mathcal{D}_1$ is correct with probability $1 - p_2$. Therefore $\mathcal{D}_1$'s probability of success is $\frac{1}{2}(p_1 + 1 - p_2) = \frac{1}{2} + \frac{1}{2}(p_1 - p_2)$. By the pseudorandomness of $\mathsf{Ind}(s, \ell^*)$, $\frac{1}{2}(p_1 - p_2)$ is negligible, and so $p_1 - p_2$ is negligible. □

Lemma 4. *Suppose that an adversary $\mathcal{A}_3$ for the $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}$ security game has probability of success p_2 in Game_2 and probability of success p_3 in Game_3. If $\mathsf{Ctr}(k_1, \ell^*)$ with random key k_1 is pseudorandom (indistinguishable from $\mathsf{rStr}(\ell^*)$), then $p_3 - p_2$ is negligible.*

Proof. We construct a distinguisher $\mathcal{D}_2$ that receives a bitstream str—either $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$ with random key k_1 or $\mathsf{str} = \mathsf{rStr}(\ell^*)$. $\mathcal{D}_2$ simulates adversary $\mathcal{A}_3$. $\mathcal{D}_2$ instantiates the PRE itself and sends $\mathcal{A}_3$ the public parameters. $\mathcal{D}_2$ responds to queries from $\mathcal{A}_3$ as follows:

- $\mathcal{D}_2$ responds to $\mathcal{O}_{\mathsf{ukey}}$, $\mathcal{O}_{\mathsf{ckey}}$, $\mathcal{O}_{\mathsf{rkey}}$, and $\mathcal{O}_{\mathsf{renc}}$ queries the same ways as $\mathcal{D}_1$ in Lemma 3.
- Challenge query $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, [i^*_0, \ldots, i^*_r], j^*)$: $\mathcal{D}_2$ generates random bit b and constructs $C^*_0, \ldots, C^*_r$, C^{pk}_{r+1}, and $\left[C^{bks}_{1,r+1}, \ldots, C^{bks}_{r,r+1}\right]$ following the procedure in Lemma 3. $\mathcal{D}_2$ gives $\mathcal{A}_3$ the $(1-\varepsilon)|C^*_r|$ bits that it adaptively requests of $\{C^*_u\}_{0 \le u \le r}$ as well as

$$C^{**} = \left(C^{pk}_{r+1}, \left[C^{bks}_{1,r+1}, \ldots, C^{bks}_{r,r+1}, \mathsf{E}\big(\mathsf{pk}_{j^*}, (0,0)\big)\right], \left[C^T_r\right]_{\mathsf{rInd}(\ell^*),\mathsf{str}}\right) .$$

- Guess b': If $b = b'$ then $\mathcal{D}_2$ guesses that $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$, otherwise $\mathcal{D}_2$ guesses that $\mathsf{str} = \mathsf{rStr}(\ell^*)$.

If $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$, then $\mathcal{A}_2$ is in Game_2 and has probability of success p_2; thus $\mathcal{D}_2$ is correct with probability p_2. If $\mathsf{str} = \mathsf{rStr}(\ell^*)$, then $\mathcal{A}_2$ is in Game_3 and has probability of success p_3; thus $\mathcal{D}_2$ is correct with probability $1 - p_3$. Therefore $\mathcal{D}_2$'s probability of success is $\frac{1}{2}(p_2 + 1 - p_3) = \frac{1}{2} + \frac{1}{2}(p_2 - p_3)$. By the pseudorandomness of $\mathsf{Ctr}(k_1, \ell^*)$, $\frac{1}{2}(p_2 - p_3)$ is negligible, and so $p_2 - p_3$ is negligible. □

Adversary's success rate in Game_3 We begin with a technical lemma:

Lemma 5. *Suppose that there are N possible balls. In the first stage, a fraction ε' are selected (by any method). In the second stage, exactly ℓ^* balls are selected uniformly at random. Let ℓ' be the number of balls selected in both stages. Then for any $t > 0$,*

$$\Pr[\ell' \le \ell^*(\varepsilon' - t)] \le e^{-2\ell^* t^2}$$

Its proof can be found in Lemma 9 from Appendix C, but follows from a standard application of Hoeffding's inequality. Next, we look at the success rate of an adversary in Game_3:

Lemma 6. *Suppose that an adversary $\mathcal{A}_4$ for the $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}$ security game has probability of success p_3 in Game_3. Suppose also that the underlying AONT T is a computationally-secure adaptive ℓ-AONT and C^T comprises at least a fraction $1-\delta$ of the total size of each ciphertext. If $\varepsilon > \delta$ and $\ell^* > \frac{\ell}{\varepsilon-\delta}$, then $p_3 < \frac{1}{2} + \mathsf{negl}$ for some negligible function negl.*

Proof. We construct an adversary $\mathcal{B}_2$ that plays the ℓ-AONT security game by simulating adversary $\mathcal{A}_4$. $\mathcal{B}_2$ instantiates the PRE itself and sends $\mathcal{A}_4$ the public parameters. $\mathcal{B}_2$ responds to queries from $\mathcal{A}_4$ as follows:

- $\mathcal{B}_2$ responds to $\mathcal{O}_{\mathsf{ukey}}$, $\mathcal{O}_{\mathsf{ckey}}$, $\mathcal{O}_{\mathsf{rkey}}$, and $\mathcal{O}_{\mathsf{renc}}$ queries the same ways as $\mathcal{D}_1$ in Lemma 3.
- Challenge query $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, [i_0^*, \ldots, i_r^*], j^*, \mathsf{bitPos})$: $\mathcal{B}_2$ generates random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$ and seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_2$ then constructs $C_0^{pk}, \ldots, C_{r+1}^{pk}$ and $C_{u,u}^{bks}, \ldots, C_{u,r+1}^{bks}$ for $u \in \{1, \ldots, r\}$ following the procedure in Lemma 3. $\mathcal{B}_2$ makes AONT challenge query $\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_0), \mathsf{E}^{\mathsf{Sym}}(k_0, M_1)\big)$, receiving oracle access to any $N-\ell$ bits of AONT output $C_0^T = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)$ in response. $\mathcal{B}_2$ responds to $\mathcal{A}_4$ by giving it oracle access to
$$\{C_u^* = \big(C_u^{pk}, \big[C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}\big], C_u^T\big)\}_{0 \le u \le r}$$
for
$$C_u^T = \Big[\cdots \Big[\big[C_0^T\big]_{\mathsf{Ind}(s_1,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\Big] \cdots\Big]_{\mathsf{Ind}(s_u,\ell^*),\mathsf{Ctr}(k_u,\ell^*)} ,$$
with a limit of $(1-\varepsilon)|C_r^*|$ total bits queried. Whenever $\mathcal{A}_4$ queries a bit of C_u^T, $\mathcal{B}_2$ queries the corresponding bit of C_0^T. For each $u' \in \{1, \ldots, u\}$, if the bit is in $\mathsf{Ind}(s_{u'}, \ell^*)$, $\mathcal{B}_2$ XORs it with the corresponding bit of $\mathsf{Ctr}(k_{u'})$. $\mathcal{B}_2$ returns the resulting bit to $\mathcal{A}_4$. When $\mathcal{A}_4$ finsihes its queries, $\mathcal{B}_2$ chooses random $L \in \left\{ {N \atop \ell^*} \right\}$ and queries any bits of $\big[C_0^T\big]_L$ that it hadn't previously queried. If this requires more than $N - \ell$ bits of C_0^T then $\mathcal{B}_2$ aborts the simulation. Otherwise, it produces $\widetilde{C_r^T} \in \{0,1\}^N$ by setting the bits not in L as their corresponding values from C_r^T and choosing the bits in L randomly. $\mathcal{B}_2$ then gives $\mathcal{A}_4$
$$C^{**} = \Big(C_{r+1}^{pk}, \big[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, \mathsf{E}\big(\mathsf{pk}_{j^*}, (0,0)\big)\big], \widetilde{C_r^T}\Big)$$
- Guess b': $\mathcal{B}_2$ receives b' as $\mathcal{A}_4$'s guess and uses the same bit b' for its guess as well.

We now look at the probability that $\mathcal{B}_2$ aborts when responding to $\mathcal{A}_4$'s challenge query. Let ε' be the fraction of C_0^T that $\mathcal{B}_2$ does not query from its oracle; note that $\varepsilon' \geq \varepsilon - \delta$. Let ℓ' be the number of bits of C_0^T that are in L but that $\mathcal{B}_2$ does not query. $\mathcal{B}_2$ aborts if and only if $\ell' < \ell$. To get any upper bound on ℓ', we apply Lemma 5. We let balls in the lemma correspond to bits here. Balls selected in the first stage correspond to bits not in bitPos, and balls selected in the second stage correspond to bits in L. Then ℓ' here is equivalent to ℓ' in the lemma. Thus by Lemma 5, for any $t > 0$, $\Pr[\ell' \le \ell^*(\varepsilon' - t)] \le e^{-2\ell^* t^2}$.

For any fixed t, this probability is negligible for $\ell^* = \omega(\log(\lambda))$. Thus as long as $\ell < \ell^* \cdot \varepsilon'$, the probability that $\mathcal{B}_2$ aborts is negligible. Since $\varepsilon' \ge \varepsilon - \delta$, if $\ell^* > \frac{\ell}{\varepsilon - \delta}$ then this will hold.

If $\mathcal{B}_2$ does not abort then its probability of success is identical to $\mathcal{A}_4$'s probability of success. Since L was chosen randomly and any string XORed with a random string is a random string, $\widetilde{C_r^T} = \left[C_r^T\right]_{\mathsf{rInd}(\ell^*),\mathsf{rStr}(\ell^*)}$.

Thus $\mathcal{A}_4's$ view here is the same as in Game_3, so its probability of success is p_3. Hence $\mathcal{B}_2$'s probability of success, when accounting for the probability that it aborts, is $p_3 - \mathsf{negl}'$ for some negligible function negl'. By the ℓ-AONT security of T, $p_3 - \mathsf{negl}' < \frac{1}{2} + \mathsf{negl}''$ for some negligible function negl''. Therefore $p_3 < \frac{1}{2} + \mathsf{negl}$ for $\mathsf{negl} = \mathsf{negl}' + \mathsf{negl}''$. □

Now we use these lemmas to prove Theorem 2.

Proof. By Lemmas 2, 3, and 4, we see that the adversary's probability of success in Game_0 can only be negligibly greater than its probability of success in Game_3. By Lemma 6, the adversary's success in Game_3 can be at most $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl. Thus the adversary's success in Game_0, which is the real case, can be at most $\frac{1}{2} + \mathsf{negl}'$ for some negligible function negl'. Combined with Theorem 1, this proves that is $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}PRE\text{-}CPA}$-secure. □

For completeness, we provide a proof for the static case in Appendix D, showing that it satisfies Definition 6. Note that while for the adaptive case the AONT must be adaptive, for the static case we can use a non-adaptive AONT.

7 Extensions to Proxy IBE and Revocable ABE

There is strong potential of identity-based proxy re-encryption [19,25,40] and revocable-storage ABE [35] in providing strong dynamic access control on untrusted clouds. Further, they can support expressive access control policies that are more difficult to achieve with traditional public-key encryptions. Therefore, being able to efficiently revoke large files using the hybrid scheme presented here is important.

We note that it is difficult to present a unified theorem that directly shows our construction immediately lifts to these primitives. This is similar to how it is difficult to have a generic hybrid encryption theorem that covers traditional PKE, IBE and ABE even though intuitively one understands the construction goes through. Regardless, in Sections 11 and 12 we show how to extend the traditional definitions to support $(1-\varepsilon)$-revocable security, and show that our hybrid constructions naturally achieve the security definitions.

8 Implementation Issues

A standard block or stream cipher can be used to implement the symmetric-key encryption and pseudo-random generator. AES with CTR mode is a natural choice for both symmetric-key encryption scheme and as the pseudorandom number generator, and is aided by the inclusion of AES in silico on many modern integrated circuits. The choice of the AONT depends on whether or not one needs an adaptive security notion. However, for practical deployment random oracle model constructions seem practical; Boyko [11] gives a tight reduction of OAEP for non-adaptive security, and we provide a loose, but technically easier, proof of adaptive security for OAEP in Lemma 25 from Appendix G.[4] Implementation details for OAEP are well understood due to its use in PKCS #1 [34]. Basing the Random Oracle implementation on standard cryptographic hashes, such as SHA-256, also typically benefits from the its inclusion in modern silicon.

From a practical perspective our construction allows certain overhead computations to be moved to the cloud, where they may be more palatable. For example, a thin client need not compute the AONT on the symmetric ciphertext—this computation does not rely on any secret data. Thus a thin client can upload an appropriate traditional hybrid encryption $(\mathsf{E}_{\mathsf{pk}}(k), \mathsf{E}^{\mathsf{Sym}}{}_k(m))$, where E is part of a proxy re-encryption scheme, and the cloud can compute $T(\mathsf{E}^{\mathsf{Sym}}{}_k(m))$ for the AONT T—the cloud covers the extra encryption costs. Similarly, if a hybrid ciphertext has not been proxy re-encrypted, the cloud can remove the AONT, reducing the decryption cost to that of traditional hybrid encryption.

We note that the more times a file is proxy re-encrypted, the more effort decryption takes, and the longer ciphertexts get. However, in the worst case we can use lazy re-encryption and amortization to solve this problem. For lazy re-encryption, when someone goes to modify a file that has been revoked several times, they will need to decrypt everything, but when they go to update the file, they can produce a new ciphertext (i.e., they do not need to reproduce the re-encryptions), and thus the ciphertext is renewed. In the case where files are not updated frequently, files that have been frequently revoked can be updated via a traditional download, re-encrypt, and upload process; but now the costs are amortized over a deployment-specific acceptable number of revocations.

Finally, a negative effect of this construction is that it limits streaming decryption services (but not streaming upload services, due to the prior observation that the cloud can apply T). However, a moment's thought shows that this is an inherent property of our security definition—if you could download a small portion of the file and start decryption, you could break CPA security.

9 Efficiency

We compare the efficiency of our scheme to a naive approach to hybrid proxy re-encryption. In the naive approach, to re-encrypt a ciphertext the proxy re-

[4] Boyko [11] provides a weaker adaptive definition of AONT security and shows that OAEP meets it; however, this definition is weaker than the definition we require.

encrypts the public-key–encrypted symmetric key using the public-key proxy re-encryption algorithm, creates a new symmetric key and encrypts it under the new public-key, and re-encrypts the already encrypted message under the new symmetric key. We note that this naive solution does not necessarily achieve our security definition, because the adversary may be able to begin decrypting the original ciphertext when it downloads a $(1-\varepsilon)$ fraction of it, and this may break CPA security. Regardless, it provides a reasonable benchmark system.

Since the public-key costs will be the same between our construction and the naive construction, we only compare the costs of symmetric-key encryption/decryption and the AONT. We assume that the scheme is implemented as described in Sect. 8, using OAEP, implemented with SHA-256, as the AONT.

The AONT is only used in our scheme for encryption and decryption. Computing the AONT on an N-bit message, as well as inverting the AONT, requires computing two hash functions: $G : \{0,1\}^{256} \to \{0,1\}^N$ and $H : \{0,1\}^N \to \{0,1\}^{256}$. According to [34], $G(R)$ for $R \in \{0,1\}^{256}$ is computed 256 bits at a time by running the SHA-256 compression function on the input R and a counter, so the compression function runs $N/256$ times. Since the block size of SHA-256 is 512 bits, computing $H(M)$ requires running the compression function $N/512$ times. Thus computing OAEP requires the SHA-256 compression function to run a total of $3N/512$ times, e.g., 49152 times for a 1 MiB file and 50331648 times for a 1 GiB file.

Table 1 compares the number of times the AES block cipher is run for each operation. This depends on ℓ^*, the number of bits that are encrypted in each re-encryption, which in turn depends on several parameters: ℓ, the minimum number of missing bits for the AONT to be secure; ε, the minimum fraction of the ciphertext not downloaded by the adversary; and δ, is the maximum fraction of the ciphertext comprised by the public-key portion. Let ε^* denote $\varepsilon - \delta$, the minimum fraction of the symmetric-key portion of the ciphertext that the adversary has not downloaded. Although Thm. 2 provides an asymptotic security guarantee, we can make it concrete. Lemmas 2, 3, and 4 give tight reductions based on the security of the underlying PRE scheme and the pseudorandomness of AES-CTR. Thus the main consideration is Lemma 6, which depends on the probability that adversary $\mathcal{B}_2$ aborts and the security of the AONT.

By Lemma 5, $\Pr[\ell' \leq \ell^*(\varepsilon' - t)] \leq e^{-2\ell^* t^2}$ for any $t > 0$, where ℓ' is the number of bits of the symmetric-key ciphertext not learned by adversary $\mathcal{A}_4$ and ε' is the fraction of the symmetric-key portion of the ciphertext not downloaded by $\mathcal{A}_4$. Since $\mathcal{B}_2$ will abort if and only if $\ell' < \ell$ and $\varepsilon' \geq \varepsilon^*$, if $\ell \leq \ell^*(\varepsilon' - t)$ then the probability that $\mathcal{B}_2$ aborts will be at most $e^{-2\ell^* t^2}$. For 128 bits of security, we want the probability that $\mathcal{B}_2$ aborts to be at most 2^{128}. Thus we need to select values for ℓ^* and t such that $\ell \leq \ell^*(\varepsilon' - t)$ and $e^{-2\ell^* t^2} \leq 2^{128}$. This system of inequalities has a solution with $t > 0$ if

$$\ell^* \geq \left(4\varepsilon^*\ell + 128\ln(2) + \sqrt{(4\varepsilon^*\ell + 128\ln(2))^2 - 16(\varepsilon^*)^2\ell^2}\right) / \left(4(\varepsilon^*)^2\right) \ .$$

Thus to maximize efficiency, we set ℓ^* as the least integer for which this inequality holds.

OAEP implemented with SHA-256 as described above with $\ell = 260$ will have 128 bits of security as an adaptive ℓ-AONT (Lemma 1), so we use 260 as our value for ℓ. We consider a few different values of ε^*.

We assume that $\mathsf{Ind}(s, \ell^*)$ runs using AES as a pseudorandom number generator. Choosing a random bit position out of N possibilities requires $\log_2 N$ bits of randomness. However, since N may not be a power of 2, we will allocate $2 \log_2 N$ bits of randomness to allow the bit selection to be sufficiently close to uniform. In addition, 1 random bit will be required to "hide" the selected bit. Thus a total of $\ell^* \cdot (2 \log_2(N) + 1)$ bits are required for each re-encryption, so the block cipher runs $\ell^* \cdot (2 \log_2(N) + 1)/128$ times per re-encryption. Encryption takes $N/128$ instances of the block cipher, while decryption takes $N/128$ instances of the block cipher to decrypt the underlying encryption plus $\ell^* \cdot (2 \log_2(N) + 1)/128$ instances for each previous re-encryption.

Table 1. Instances of the AES block cipher required for each operation in the naive approach and in our scheme, where r is the number of re-encryptions

File size	ε^*	ℓ^*	Encryption	Re-enc.	Decryption		
					$r = 1$	$r = 10$	$r = 100$
	naive		6.554×10^4	6.554×10^4	1.311×10^5	7.209×10^5	6.619×10^6
1 MiB	0.5	926	3.400×10^2	6.554×10^4	6.588×10^4	6.894×10^4	9.954×10^4
$N = 2^{23}$	0.25	2325	8.537×10^2	6.554×10^4	6.639×10^4	7.407×10^4	1.509×10^5
	0.1	8875	3.259×10^3	6.554×10^4	6.879×10^4	9.812×10^4	3.914×10^5
	naive		6.711×10^7	6.711×10^7	1.342×10^8	7.382×10^8	6.778×10^9
1 GiB	0.5	926	4.847×10^2	6.711×10^7	6.711×10^7	6.711×10^7	6.716×10^7
$N = 2^{33}$	0.25	2325	1.217×10^3	6.711×10^7	6.711×10^7	6.712×10^7	6.723×10^7
	0.1	8875	4.646×10^3	6.711×10^7	6.711×10^7	6.716×10^7	6.757×10^7

Figure 6 shows the effect that file size has on the cost of re-encryption, comparing naive re-encryption and our scheme with various values of ε^*. Naive re-encryption cost is linear in terms of file size, while our re-encryption cost is logarithmic in terms of file size (the number of bits re-encrypted is independent of file size, but the amount of pseudorandomness required to choose the bit positions to re-encrypt is logarithmic in terms of file size). Because the cost in the naive scheme dominates the cost in our scheme for any reasonably-sized file, this is presented as a log-log graph.

Figure 7 shows the effect that the number of previous re-encryptions has on the cost of decryption (for AES), comparing naive re-encryption with our scheme at $\varepsilon^* = 0.1$ for various small file sizes. For both schemes the cost of decryption is linear in the number of previous re-encryptions, but if the file is bigger than just 40 KiB the decryption cost grows faster for naive re-encryption than for our scheme. For files larger than 200 KiB the decryption cost for the naive scheme completely dominates our scheme when more than a few re-encryptions have

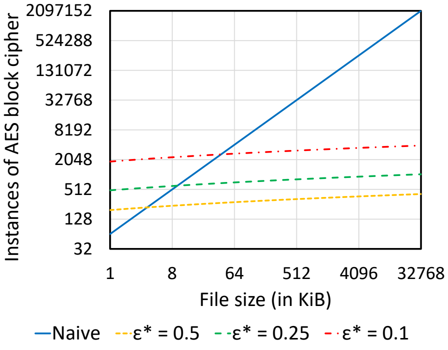

Fig. 6. Cost of re-encryption vs. file size (log-log graph)

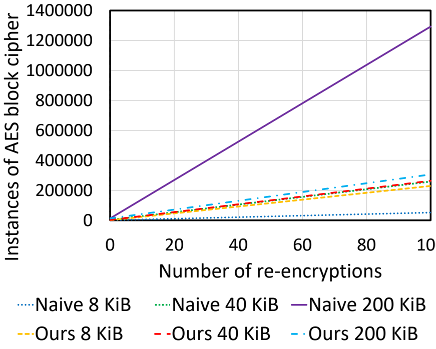

Fig. 7. Cost of decryption vs. number of previous re-encryptions

occurred. Note that while SHA-256 costs will add to the cost of decryption for our scheme only, this cost does not grow with the number of re-encryptions.

While our scheme is slower for encryption (due to computing the hash function), in practice this will only occur once for each file. In addition, it is unlikely that a user will encrypt a large number of files at once, so any computation will likely be minimally disruptive. Re-encryption costs are more significant because re-encryption can occur for a large number of files at the same time, such as when a user leaves or changes jobs, and so all the files the user had access to need to be re-encrypted. In this case, our scheme is several orders of magnitude faster than the naive approach, with the difference greater for larger files. Finally, while our scheme is slower than the naive scheme when decrypting a file that has never been re-encrypted, as noted in the previous section, the cloud can offload these costs from the client. Alternately, once a file has been re-encrypted enough times our scheme becomes significantly faster.

10 Bounded Storage Security and Error Correction

Suppose that an adversary is able to download and view each file in its entirety, but is limited in the amount of information stored about each file. That is, what if we consider a model where the adversary is limited in storage instead of bandwidth? First, we argue that in today's reality, bandwidth is a more expensive requirement than storage, and many security operations centers actively monitor for large data exfiltration.

Regardless, imagine we want to proxy re-encrypt each file so that the adversary cannot reconstruct the original file from its limited storage (smaller than the original file) and the re-encrypted file. We show that in such a scenario, then there can be no solution where we are able to only touch a small number of the bits in the original ciphertext, as our construction does. That is, any solution will have to "re-encrypt" a much larger fraction of the original ciphertext than our construction. This follows naturally from linear error-correcting codes.

We formally define this scenario in Figure 5 (pg. 15) for adversary $\mathcal{A}$, re-encryption scheme Π, and file f.

Theorem 3. *Let Π be a re-encryption scheme such that when given a file of length N, it modifies ℓ bits of f. Then there exists an adversary $\mathcal{A}$ such that for any file f of length N and any $n > \ell$,* $\Pr[\mathsf{Bounded\text{-}Storage\text{-}Reconst}_{\mathcal{A},\Pi}(f, n) = 1]$ *is a non-negligible function of N.*

Proof. Adversary $\mathcal{A}$ can apply a systematic folded Reed-Solomon code, a systematic linear error-correcting code, of message length N and block length $N+n$ to f. It will storing the n syndrome bits in σ. Thus the code maps f to (f, σ), resulting in code rate $R = \frac{N}{N+n}$. Since $\widetilde{f}$ is f with ℓ bits modified, $(\widetilde{f}, \sigma)$ is (f, σ) with ℓ errors, resulting in a fraction $\frac{\ell}{N+n}$ of errors. Note that the n bits the user stores will not incur errors, so all ℓ errors will occur in the original N bits. However, this restriction on possible error locations can only help $\mathcal{A}$ further.

Let $\varepsilon = \frac{n-\ell}{N+n}$. Since $n > \ell$, ε is positive. Then $1 - R - \varepsilon = \ell/(N + n)$. Now $\mathcal{A}$ applies the list decoding algorithm for folded Reed-Solomon codes from [20] to f as described above. Since ε is positive and the fraction of errors is $1-R-\varepsilon$, $\mathcal{A}$ can produce a polynomial-sized list containing $f \,\|\, \sigma$. It can then choose an item from this list at random and output the first N bits, resulting in a non-negligible probability of $\mathcal{A}$ outputting f. □

In this scenario, re-encryption is insecure if it does not touch at least as many bits as the adversary stores. This makes it much less efficient in the scenario where the adversary does not download the entire file, where re-encryption can touch far fewer bits than the adversary downloads and stores while remaining secure.

11 Identity-Based Proxy Re-Encryption

The traditional notion of identity-based encryption [9] was extended to include proxy re-encryption by Green and Ateniese [19].

Definition 8 (Identity-Based Proxy Re-Encryption (IB-PRE) [19]). *An identity-based proxy re-encryption primitive consists of six probabilistic polynomial time algorithms:*

- $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{param}, \mathsf{msk})$. *The setup algorithm generates global parameters. Both λ and* param *are considered implicit parameters to the remaining algorithms, but are suppressed for clarity of notation.* msk *is the master secret key.*
- $\mathsf{G}(\mathsf{msk}, \mathsf{id}) \rightarrow \mathsf{sk}_{\mathsf{id}}$. *Key generation produces a secret key corresponding to the given identity.*
- $\mathsf{E}(\mathsf{id}, M) \rightarrow C$. *Identity-based encryption takes an identity and message and generates a ciphertext.*

- $\mathsf{D}(\mathsf{sk}_{\mathsf{id}}, C) \rightarrow M$. *Decryption takes a secret key and ciphertext and returns the underlying message.*
- $\mathsf{RG}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_j) \rightarrow \mathsf{rk}_{\mathsf{id}_i \rightarrow \mathsf{id}_j}$. *Re-encryption key generation takes a secret key for the source and the identities of the source and destination, and creates a re-encryption key that can transform a ciphertext encrypted under the source's identity* id_i *to one encrypted under the destination's identity* id_j.
- $\mathsf{RE}(\mathsf{rk}_{\mathsf{id}_i \rightarrow \mathsf{id}_j}, C_{\mathsf{id}_i}) \rightarrow C_{\mathsf{id}_j}$. *Re-encryption takes a re-encryption key and a ciphertext encrypted under the re-encryption key's corresponding source identity, and translates it into a ciphertext under the destination's identity.*

11.1 Correctness

An IB-PRE scheme is correct if all encryptions and proxy re-encryptions decrypt properly. Formally, for every message M and every set of identity/secret-key pairs $\{\mathsf{id}_i, \mathsf{sk}_{\mathsf{id}_i} \leftarrow \mathsf{G}(\mathsf{msk}, \mathsf{id}_i)\}_{i \in \{0,\ldots,r\}}$ and re-encryption keys

$$\left\{\mathsf{rk}_{\mathsf{id}_i \rightarrow \mathsf{id}_{i+1}} \leftarrow \mathsf{RG}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_{i+1})\right\}_{i \in \{0,\ldots,r-1\}} ,$$

we have

$$\mathsf{D}\big(\mathsf{sk}_{\mathsf{id}_r}, \mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_{r-1} \rightarrow \mathsf{id}_r}, \ldots \mathsf{RE}(\mathsf{rk}_{\mathsf{id}_0 \rightarrow \mathsf{id}_1}, \mathsf{E}(\mathsf{id}_0, M)) \ldots\big)\big) = M \ .$$

11.2 Unidirectional, Multi-Hop, IB-PRE CPA-Security

Definition 9 (Basic Unidirectional, Multi-Hop, IB-PRE CPA-Security Game [19]). *Let λ be the security parameter. Let $\mathcal{A}$ be an oracle TM, representing the adversary. The* IB-PRE-CPA *game consists of an execution of $\mathcal{A}$ in two phases, which are executed in order, as described in Alg. 1.*

Algorithm 1 Identity-Based Proxy Re-Encryption CPA-Security

```
experiment IB-PRE-CPA$_{\mathcal{A},\Pi}(\lambda)$
    $\mathsf{param} \leftarrow \mathsf{G}(1^\lambda)$
    $(M_0, M_1, id^*, \sigma) \leftarrow A^{\mathcal{O}_{\mathsf{key}}, \mathcal{O}_{\mathsf{rkey}}}(\mathsf{param})$
    $b \leftarrow \{0,1\}$
    $C^* \leftarrow \mathsf{E}(\mathsf{id}^*, M_b)$
    $b' \leftarrow A^{\mathcal{O}_{\mathsf{key}}, \mathcal{O}_{\mathsf{rkey}}}(\sigma, C^*)$
    Output 1 iff $b = b'$
end experiment
```

Within each phase, $\mathcal{A}$ has access to oracles producing secret keys and re-encryption keys, which can be queried in any order, poly(λ) times, as follows:

- *Key Generation* $\mathcal{O}_{\mathsf{key}}(\mathsf{id})$*: Output* $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, \mathsf{id})$.
- *Re-Encryption Key Generation* $\mathcal{O}_{\mathsf{rkey}}(\mathsf{id}_i, \mathsf{id}_j)$. *Output* $\mathsf{rk}_{\mathsf{id}_i \rightarrow \mathsf{id}_j} \leftarrow \mathsf{RG}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_j)$.

Challenge identity id^* *cannot be chosen to allow trivial decryption of the challenge ciphertext from the keys queried in Phase 1. This means that if* $\mathcal{A}$ *queries a series of re-encryption keys going from* id_i *to* id_j*, and it queries* $\mathcal{O}_{\mathsf{key}}(\mathsf{id}_j)$*, then it cannot select* id_i *or* id_j *as the challenge identity. Similarly, in Phase 2, no queries can be made that would allow trivial decryption of the challenge ciphertext.*

Definition 10. *An Identity-Based Proxy Re-Encryption scheme* Π *is Unidirectional, Multi-Hop, IB-PRE CPA-Secure if for all oracle P.P.T. adversaries* A*, there exists a negligible function* negl *such that*

$$\Pr[\mathsf{IB\text{-}PRE\text{-}CPA}_{A,\Pi}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \enspace .$$

11.3 $(1-\varepsilon)$-Revocable, Unidirectional, Multi-Hop, IB-PRE CPA-Security

We now modify the above security definition for IB-PRE in a similar way as for regular PRE.

Definition 11 ($(1-\varepsilon)$-{Static, Adaptive}-Revocable, Unidirectional, Multi-Hop, IB-PRE CPA-Security Game). *We define the new games* $(1-\varepsilon)$-$\mathsf{Stat\text{-}Revoke\text{-}IB\text{-}PRE\text{-}CPA}_{A,\Pi}(\lambda)$ *and* $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}IB\text{-}PRE\text{-}CPA}_{A,b}(\lambda)$ *as a modification of* $\mathsf{IB\text{-}PRE\text{-}CPA}_{\mathcal{A},\Pi}(\lambda)$ *given in Defn. 9, as described in Alg. 2.*

Algorithm 2 $(1-\varepsilon)$-{Static, Adaptive}-Revocable Identity-Based Proxy Re-Encryption CPA-Security

```
experiment $(1-\varepsilon)$-$\{\mathsf{Stat}, \mathsf{Adap}\}$-$\mathsf{Revoke\text{-}IB\text{-}PRE\text{-}CPA}_{\mathcal{A},\Pi}(\lambda)$
    $\mathsf{param} \leftarrow \mathsf{G}(1^\lambda)$
    $(M_0, M_1, [\mathsf{id}_0^*, \ldots, \mathsf{id}_r^*], \mathsf{id}^{**}), \sigma \leftarrow A^{\mathcal{O}_{\mathsf{key}}, \mathcal{O}_{\mathsf{rkey}}}(\mathsf{param})$
    $b \leftarrow \{0, 1\}$
    $C_0^* \leftarrow \mathsf{E}(\mathsf{id}_0^*, M_b)$
    For $0 < u \leq r$, $C_u^* \leftarrow \mathsf{RE}(\mathsf{rk}_{\mathsf{id}_{u-1}^* \to \mathsf{id}_u^*}, C_{u-1}^*)$
    $\sigma' \leftarrow A^{\mathcal{O}_{\mathsf{key}}, \mathcal{O}_{\mathsf{rkey}}, \lfloor \mathsf{C}^* \rceil}(\sigma)$
    $C^{**} \leftarrow \mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_r^* \to \mathsf{id}^{**}}, C^*\big)$ for $\mathsf{rk}_{\mathsf{id}_r^* \to \mathsf{id}^{**}} = \mathsf{RG}\big(\mathsf{sk}_{\mathsf{id}_r^*}, \mathsf{id}_r^*, \mathsf{id}^{**}\big)$
    $b' \leftarrow A^{\mathcal{O}_{\mathsf{key}}, \mathcal{O}_{\mathsf{rkey}}}(\sigma', C^{**})$
    Output 1 iff $b = b'$
end experiment
```

Note that the experiment provides a new oracle, $\lfloor \mathsf{C}^* \rceil$*, that provides random access to* $(1-\varepsilon)N'$ *bits of* $\{C_u^*\}_{0 \leq u \leq r}$*, where* $N' = \max_{0 \leq u \leq r} |C_u^*|$*. Again, this access is either adaptive or static, and is equivalent to the adversary selecting* bitPos *in Defn. 5. Once the adversary is done querying* $\lfloor \mathsf{C}^* \rceil$*, it is given complete access to* C^{**}*. The adversary can output* any *distinct values of* $[\mathsf{id}_0^*, \ldots, \mathsf{id}_r^*]$*. However,* id^{**}*, which must be distinct from each identity in* $[\mathsf{id}_0^*, \ldots, \mathsf{id}_r^*]$*, must also be an identity that cannot be trivially decrypted through either a direct query* *to $\mathcal{O}_{\mathsf{key}}$, or through a sequence of $\mathcal{O}_{\mathsf{rkey}}$ queries and then a $\mathcal{O}_{\mathsf{key}}$ query. Similarly, after the adversary has been given the challenge ciphertext C^{**} we disallow any queries that would allow trivial decryption of C^{**}.*

Definition 12. *An identity-based proxy re-encryption scheme Π is $(1-\varepsilon)$-$\{$Static, Adaptive$\}$-Revocable, Unidirectional, Multi-Hop, IBEPRE CPA-Secure if for all oracle P.P.T. adversaries $\mathcal{A}$, there exists a negligible function* negl *such that*

$$\Pr[(1-\varepsilon)\text{-}\{\mathsf{Stat},\mathsf{Adap}\}\text{-}\mathsf{Revoke\text{-}IB\text{-}PRE\text{-}CPA}_{\mathcal{A},\Pi}(\lambda)=1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

and

$$\Pr[\mathsf{IB\text{-}PRE\text{-}CPA}_{\mathcal{A},\Pi}(\lambda)=1] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \ .$$

11.4 Identity-Based Proxy Re-Encryption Construction

We now give a hybrid construction of an identity-based proxy re-encryption scheme $\Pi_{hyb} = \left(\mathsf{Setup}^{\mathsf{Hyb}}, \mathsf{G}^{\mathsf{Hyb}}, \mathsf{E}^{\mathsf{Hyb}}, \mathsf{D}^{\mathsf{Hyb}}, \mathsf{RG}^{\mathsf{Hyb}}, \mathsf{RE}^{\mathsf{Hyb}}\right)$, assuming the existence of an identity-based proxy re-encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$, a symmetric-key encryption scheme $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$, and an AONT T. Our scheme is the IB-PRE version of the PRE construction from Section 6.3. It uses the same notation as Section 6.2 and has the following six algorithms:

- $\mathsf{Setup}^{\mathsf{Hyb}}(1^\lambda) = \mathsf{Setup}(1^\lambda)$
- $\mathsf{G}^{\mathsf{Hyb}}(\mathsf{msk}, \mathsf{id}) = \mathsf{G}(\mathsf{msk}, \mathsf{id})$
- $\mathsf{E}^{\mathsf{Hyb}}$, which is described in Alg. 3
- $\mathsf{D}^{\mathsf{Hyb}}$, which is described in Alg. 4
- $\mathsf{RG}^{\mathsf{Hyb}}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_j)$, which outputs $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j} = \mathsf{RG}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_j)$.
- $\mathsf{RE}^{\mathsf{Hyb}}$, which is described in Alg. 5

Algorithm 3 $\mathsf{E}^{\mathsf{Hyb}}$ Encryption Pseudocode

```
procedure $\mathsf{E}^{\mathsf{Hyb}}(\mathsf{id}, M)$
    $k_0 \leftarrow \mathsf{G}^{\mathsf{Sym}}(1^\lambda)$
    $C^{pk} \leftarrow \mathsf{E}(\mathsf{id}, k_0)$
    $C^T \leftarrow T(\mathsf{E}^{\mathsf{Sym}}(k_0, M))$
    return $C = (C^{pk}, [\,], C^T)$
end procedure
```

11.5 Correctness

A valid ciphertext for message M under public key id with corresponding secret key $\mathsf{sk}_{\mathsf{id}}$ has the form $C = \left(C^{pk}, \left[C_1^{bks}, \ldots, C_r^{bks}\right], C_r^T\right)$ where

Algorithm 4 $\mathsf{D}^{\mathsf{Hyb}}$ Decryption Pseudocode

```
procedure $\mathsf{D}^{\mathsf{Hyb}}\big(\mathsf{sk}, C = \big(C^{pk}, \big[C_1^{bks}, \ldots, C_r^{bks}\big], C_r^T = C_{r,1}^T \cdots C_{r,N}^T\big)\big)$
    for $u \leftarrow r, \ldots, 1$ do
        $(s_u, k_u) \leftarrow \mathsf{D}\big(\mathsf{sk}, C_u^{bks}\big)$
        $\mathsf{ind}_u \leftarrow \mathsf{Ind}(s_u, \ell^*)$
        $\mathsf{str}_u \leftarrow \mathsf{Ctr}(k_u, \ell^*)$
        for $v \leftarrow 1, \ldots, N$ do
            if $v \in \mathsf{ind}_u$ then
                $C_{u-1,v}^T \leftarrow C_{u,v}^T \oplus \mathsf{str}[v]$
            else
                $C_{u-1,v}^T \leftarrow C_{u,v}^T$
            end if
        end for
    end for
    $C^T \leftarrow C_{0,1}^T \cdots C_{0,N}^T$
    $k_0 \leftarrow \mathsf{D}\big(\mathsf{sk}, C^{pk}\big)$
    return $M \leftarrow \mathsf{D}^{\mathsf{Sym}}\big(k_0, T^{-1}\big(C^T\big)\big)$
end procedure
```

Algorithm 5 $\mathsf{RE}^{\mathsf{Hyb}}$ Re-Encryption Pseudocode

```
procedure $\mathsf{RE}^{\mathsf{Hyb}}\big(\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j}, C = \big(C^{pk}, \big[C_1^{bks}, \ldots, C_r^{bks}\big], C^T = C_1^T \cdots C_N^T\big)\big)$
    $\widetilde{C^{pk}} \leftarrow \mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j}, C^{pk}\big)$
    for $u \leftarrow 1, \ldots, r$ do
        $\widetilde{C_u^{bks}} \leftarrow \mathsf{RE}\big(rk_{\mathsf{id}_i \to \mathsf{id}_j}, C_u^{bks}\big)$
    end for
    Choose $s_{r+1}, k_{r+1}$ uniformly at random
    $C_{r+1}^{bks} \leftarrow \mathsf{E}(\mathsf{id}_j, (s_{r+1}, k_{r+1}))$
    $\mathsf{ind}_{r+1} \leftarrow \mathsf{Ind}(s_{r+1}, \ell^*)$
    $\mathsf{str}_{r+1} \leftarrow \mathsf{Ctr}(k_{r+1}, \ell^*)$
    for $v \leftarrow 1, \ldots, N$ do
        if $v \in \mathsf{ind}_{r+1}$ then
            $\widetilde{C_v^T} \leftarrow C_v^T \oplus \mathsf{str}_{r+1}[v]$
        else
            $\widetilde{C_v^T} \leftarrow C_v^T$
        end if
    end for
    $\widetilde{C^T} \leftarrow \widetilde{C_1^T} \cdots \widetilde{C_N^T}$
    return $C = \Big(\widetilde{C^{pk}}, \Big[\widetilde{C_1^{bks}}, \ldots, \widetilde{C_{r+1}^{bks}}\Big], \widetilde{C^T}\Big)$
end procedure
```

- C^{pk} is an encryption under id of symmetric key k_0
- Each C_u^{bks} is an encryption under id of (s_u, k_u)
- C_r^T is $T\big(\mathsf{E}^{\mathsf{Sym}}(k, M)\big)$ with the bits at positions $\mathsf{Ind}(s_1)$ encrypted under symmetric key k_1, then the bits at positions $\mathsf{Ind}(s_2, \ell^*)$ encrypted under symmetric key k_2, etc.

The decryption algorithm computes $k_0 = \mathsf{D}\big(\mathsf{sk}_{\mathsf{id}}, C^{pk}\big)$ and each $(s_u, k_u) = \mathsf{D}\big(\mathsf{sk}_{\mathsf{id}}, C_u^{bks}\big)$; computes C_0^T by decrypting the bits of C_r^T at positions $\mathsf{Ind}(s_r, \ell^*)$ using symmetric key k_r to produce C_{r-1}^T, then decrypting the bits of C_{r-1}^T at positions $\mathsf{Ind}(s_{r-1}, \ell^*)$ using symmetric key k_{r-1} to produce C_{r-2}^T, etc., eventually reaching C_0^T; and finally computes $M = \mathsf{D}^{\mathsf{Sym}}\big(k, T^{-1}\big(C^T\big)\big)$.

11.6 Security Theorems

We have the following security theorems, which are the IB-PRE versions of Theorems 1, 2, and 11, respectively. The proofs are analogous to the proofs from the corresponding PRE theorems presented in Appendix B, Section 6, and Appendix D, respectively. See Appendix E for details.

Theorem 4. *Assume the existence of a* IB-PRE-CPA*-secure identity-based proxy re-encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$*, a CPA-secure symmetric-key encryption scheme* $\Pi_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$*, and an AONT* T*. Then the construction in Section 11.4 is* IB-PRE-CPA*-secure.*

Theorem 5. *Assume the existence of a* IB-PRE-CPA*-secure identity-based proxy re-encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$ *with re-encryption history independence, a symmetric-key encryption scheme* $\Pi_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$*, and an adaptive* ℓ*-AONT* T*. Suppose that for the construction from Section 11.4,* C^T *comprises at least a fraction* $1 - \delta$ *of the total size of each ciphertext. Then for any* $\varepsilon < 1$ *with* $\varepsilon > \delta$ *and any* $\ell^* > \frac{\ell}{\varepsilon - \delta}$*, this construction is* $(1 - \varepsilon)$-Adap-Revoke-IB-PRE-CPA*-secure.*

Theorem 6. *Assume the existence of a* IB-PRE-CPA*-secure identity-based proxy re-encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$ *with re-encryption history independence, a symmetric-key encryption scheme* $\Pi_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$*, and an* ℓ*-AONT* T*. Suppose that for the construction from Section 11.4,* C^T *comprises at least a fraction* $1-\delta$ *of the total size of each ciphertext. Then for any* $\varepsilon < 1$ *with* $\varepsilon > \delta$ *and any* $\ell^* > \frac{\ell}{\varepsilon - \delta}$*, this construction is* $(1-\varepsilon)$-Stat-Revoke-IB-PRE-CPA*-secure.*

12 Revocable-Storage Attribute-Based Encryption

Revocable-Storage Attribute-Based Encryption [35] combines the notion of attribute-based encryption (ABE) [36,18] with identity-based revocation [8]. It allows for users' keys to be revoked and also provides for ciphertext delegation—re-encrypting ciphertexts to more restrictive policies—so that revoked users cannot access existing data. This provides a natural use for our techniques, as the existing data will need to be encrypted under a new symmetric key. We focus on key-policy ABE (KP-ABE), where each key is associated with a policy and each ciphertext with a set of attributes, though our techniques could also be used with ciphertext-policy ABE (CP-ABE).

Definition 13 (Revocable-Storage KP-ABE [35]). *A revocable-storage KP-ABE scheme with time bound T consists of six probabilistic polynomial time algorithms:*

- $\mathsf{Setup}(1^\lambda) \to (\mathsf{param}, \mathsf{msk})$*. The setup algorithm generates global parameters. Both* λ *and* param *are considered implicit parameters to the remaining algorithms, but are suppressed for clarity of notation.* msk *is the master secret key.*
- $\mathsf{G}(\mathsf{msk}, P, \mathsf{id}) \to \mathsf{sk}_{P,\mathsf{id}}$*. Key generation produces a secret key corresponding to the given policy* P *and identity* id*.*
- $\mathsf{E}(S, M, t) \to C_{S,t}$*. Attribute-based encryption encrypts message* M *under attribute set* S *at time* $t \leq T$*.*
- $\mathsf{KU}(\mathsf{msk}, \mathsf{rl}, t) \to \mathsf{ku}_t$*. The key update algorithm produces key update information for revocation list* rl *(a set of revoked identities) at time* t*.*
- $\mathsf{D}(\mathsf{sk}_{P,\mathsf{id}}, \mathsf{ku}_{t'}, C_{S,t}) \to M$*. Decryption takes a secret key* $\mathsf{sk}_{P,id}$*, key update information* $\mathsf{ku}_{t'}$*, and a ciphertext* $C_{S,t}$*, and returns a message or* $\perp$*.*
- $\mathsf{CTU}(C_{S,t}) \to C_{S,t+1}$*. The ciphertext update algorithm transforms a ciphertext encrypted under time* t *to a ciphertext encrypted at time* $t+1$*.*

12.1 Correctness

A revocable-storage KP-ABE scheme is correct if for every message M, identity id and revocation list r such that $\mathsf{id} \notin \mathsf{rl}$, every policy P and attribute set S such that $P(S) = 1$, and times $t' \geq t + r$ for $r \geq 0$,

$$\mathsf{D}(\mathsf{G}(\mathsf{msk}, P, \mathsf{id}), \mathsf{KU}(\mathsf{msk}, \mathsf{rl}, t'), \mathsf{CTU}(\cdots \mathsf{CTU}(\mathsf{E}(S, M, t)) \cdots)) = M \ .$$

Note that this is weaker than the correctness requirement from [35], which additionally requires that for every message M, attribute set S, and time t,

$$\mathsf{E}(S, M, t+1) \equiv \mathsf{CTU}(\mathsf{E}(S, M, t))$$

where $\equiv$ denotes equal distributions. This means that encrypting a message at time $t+1$ results in the same distribution as encrypting the same message at time t and then updating it to time $t+1$. Our scheme will not achieve this requirement—the size of the ciphertext grows with each update, and most bits of the ciphertext are not changed with each update. However, this requirement for equal distributions is only needed when a ciphertext at time t is updated to time $t+1$ multiple times, as then the randomness may be correlated, possibly breaking security. Since this will not occur when the scheme is used as intended for revocation, we do not believe that the requirement is necessary.

12.2 Revocable Storage KP-ABE Security

Definition 14 (Basic Revocable-Storage KP-ABE Security Game [35]). *Let λ be the security parameter. Let $\mathcal{A}$ be an oracle TM, representing the adversary. The* $\mathsf{RS\text{-}KP\text{-}ABE}$ *game consists of an execution of $\mathcal{A}$ in two phases, which are executed in order, as described in Alg. 6.*

Algorithm 6 Revocable-Storage KP-ABE Security

```
experiment $\mathsf{RS\text{-}KP\text{-}ABE}_{\mathcal{A},\Pi}(\lambda)$
    $\mathsf{param} \leftarrow \mathsf{G}(1^\lambda)$
    $(M_0, M_1, S^*, t^*, \sigma) \leftarrow A^{\mathcal{O}_{\mathsf{sk}},\mathcal{O}_{\mathsf{ku}}}(\mathsf{param})$
    $b \leftarrow \{0,1\}$
    $C_{S^*,t^*} \leftarrow \mathsf{E}(S^*, M_b, t^*)$
    $b' \leftarrow A^{\mathcal{O}_{\mathsf{sk}},\mathcal{O}_{\mathsf{ku}}}(\sigma, C^*)$
    Output 1 iff $b = b'$
end experiment
```

Within each phase, $\mathcal{A}$ has access to oracles producing secret keys and key updates, which can be queried in any order, poly(λ) times, as follows:

- *Secret Key Generation* $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$*: Output* $\mathsf{sk}_{P,\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, P, \mathsf{id})$.
- *Key Update Generation* $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$*. Output* $\mathsf{ku}_t \leftarrow \mathsf{KU}(\mathsf{msk}, t, \mathsf{rl})$.

The challenge attribute set S^* and time t^* cannot be chosen to allow trivial decryption of the challenge ciphertext from the outputs of the queries from Phase 1. This means that if $\mathcal{A}$ has queried $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$ and $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$ for $\mathsf{id} \notin \mathsf{rl}$, then it it cannot select S^* and t^* if $P(S^*) = 1$ and $t \geq t^*$. Similarly, in Phase 2, no queries can be made that would allow trivial decryption of the challenge ciphertext.

Definition 15. *A key-policy attribute-based encryption scheme Π has Revocable Storage if for all oracle P.P.T. adversaries A, there exists a negligible function* negl *such that*

$$\Pr[\mathsf{RS\text{-}KP\text{-}ABE}_{A,\Pi}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \ .$$

12.3 $(1-\varepsilon)$-Revocable, Revocable Storage KP-ABE Security

We now modify the above security definition for Revocable Storage KP-ABE in a similar way as for proxy re-encryption. Here we have in mind a scenario where a user has access at time $t^* + r$ to a file created at time t^*, but then the user is revoked at time $t^* + r + 1$. This user may have downloaded and decrypted part of the time-$(t^* + r)$ or previous versions of this file and obtained the symmetric key. However, when this file is updated to time $t^* + r + 1$, we want the adversary now to be unable to decrypt the ciphertext. This is the same goal as for the proxy re-encryption case except that files are updated to a new time instead of being re-encrypted to a new key.

Definition 16 **($(1-\varepsilon)$-{Static, Adaptive}-Revocable, Revocable Storage KP-ABE Security Game).** *We define the new games* $(1{-}\varepsilon)$-$\mathsf{Stat\text{-}Revoke\text{-}RS\text{-}KP\text{-}ABE}_{A,\Pi}(\lambda)$ *and* $(1{-}\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}RS\text{-}KP\text{-}ABE}_{A,b}(\lambda)$ *as a modification of* $\mathsf{RS\text{-}KP\text{-}ABE}_{\mathcal{A},\Pi}(\lambda)$ *given in Defn. 14, as described in Alg. 7.*

Algorithm 7 $(1-\varepsilon)$-{Static, Adaptive}-Revocable Revocable Storage KP-ABE Security

```
experiment $(1-\varepsilon)$-$\{\mathsf{Stat}, \mathsf{Adap}\}$-$\mathsf{Revoke\text{-}RS\text{-}KP\text{-}ABE}_{\mathcal{A},\Pi}(\lambda)$
    $\mathsf{param} \leftarrow \mathsf{G}(1^\lambda)$
    $(M_0, M_1, S^*, t^*, r), \sigma \leftarrow A^{\mathcal{O}_{\mathsf{sk}}, \mathcal{O}_{\mathsf{ku}}}(\mathsf{param})$
    $b \leftarrow \{0, 1\}$
    $C_{S^*,t^*} \leftarrow \mathsf{E}(S^*, M_b, t^*)$
    For $0 < u \leq r + 1$, $C_{S^*,t^*+u} \leftarrow \mathsf{CTU}(C_{S^*,t^*+u-1})$
    $\sigma' \leftarrow A^{\mathcal{O}_{\mathsf{key}}, \mathcal{O}_{\mathsf{rkey}}, \lfloor \mathsf{C}^* \rceil}(\sigma)$
    $b' \leftarrow A^{\mathcal{O}_{\mathsf{key}}, \mathcal{O}_{\mathsf{rkey}}}(\sigma', C_{S^*,t^*+r+1})$
    Output 1 iff $b = b'$
end experiment
```

Note that the experiment provides a new oracle, $\lfloor \mathsf{C}^* \rceil$, *that provides random access* $(1-\varepsilon)N'$ *bits of* $\{C_{S^*,t^*+u}\}_{0 \leq u \leq r}$, *where* $N' = \max_{0 \leq u \leq r} |C_{S^*,t^*+u}|$. *Again, this access is either adaptive or static, and is equivalent to the adversary selecting* bitPos *in Defn. 5. Once the adversary is done querying* $\lfloor \mathsf{C}^* \rceil$, *it is given complete access to* C_{S^*,t^*+r+1}. *We allow the adversary to make queries such that* $\{C_{S^*,t^*+u}\}_{0 \leq u \leq r}$ *can be trivially decrypted. However, the challenge attribute set* S^* *and time* $t^* + r$ *cannot be chosen to allow trivial decryption of* C_{S^*,t^*+r+1}. *This means that if* $\mathcal{A}$ *has queried* $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$ *and* $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$ *for* $\mathsf{id} \notin \mathsf{rl}$, *then it it cannot select* S^* *and* t^* *if* $P(S^*) = 1$ *and* $t \geq t^* + r + 1$.

Definition 17. *A Key-Policy Attribute-Based Encryption scheme* Π *has* $(1{-}\varepsilon)$-*{Static, Adaptive}-Revocable Revocable Storage if for all oracle P.P.T. adversaries A, there exists a negligible function* negl *such that*

$$\Pr[(1-\varepsilon)\text{-}\{\mathsf{Stat}, \mathsf{Adap}\}\text{-}\mathsf{Revoke\text{-}RS\text{-}KP\text{-}ABE}_{\mathcal{A},\Pi}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

and

$$\Pr[\mathsf{RS\text{-}KP\text{-}ABE}_{\mathcal{A},\Pi}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda) \ .$$

12.4 Revocable-Storage KP-ABE Construction

We now give a hybrid construction of a revocable-storage KP-ABE scheme $\Pi_{hyb} = \left(\mathsf{Setup}^{\mathsf{Hyb}}, \mathsf{G}^{\mathsf{Hyb}}, \mathsf{E}^{\mathsf{Hyb}}, \mathsf{KU}^{\mathsf{Hyb}}, \mathsf{D}^{\mathsf{Hyb}}, \mathsf{CTU}^{\mathsf{Hyb}}\right)$, assuming the existence of a revocable-storage KP-ABE scheme $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$, a symmetric-key encryption scheme $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$, and an AONT T. Our scheme is the revocable-storage KP-ABE version of the PRE construction from Section 6.3. It uses the same notation as Section 6.2 and has the following six algorithms:

- $\mathsf{Setup}^{\mathsf{Hyb}}\big(1^\lambda\big) = \mathsf{Setup}\big(1^\lambda\big)$
- $\mathsf{G}^{\mathsf{Hyb}}(\mathsf{msk}, P, \mathsf{id}) = \mathsf{G}(\mathsf{msk}, P, \mathsf{id})$
- $\mathsf{E}^{\mathsf{Hyb}}$, which is described in Alg. 8
- $\mathsf{KU}^{\mathsf{Hyb}}(\mathsf{msk}, \mathsf{rl}, t) = KU(\mathsf{msk}, \mathsf{rl}, t)$
- $\mathsf{D}^{\mathsf{Hyb}}$, which is described in Alg. 9
- $\mathsf{CTU}^{\mathsf{Hyb}}$, which is described in Alg. 10

Algorithm 8 $\mathsf{E}^{\mathsf{Hyb}}$ Encryption Pseudocode

```
procedure $\mathsf{E}^{\mathsf{Hyb}}(S, M, t)$
    $k_0 \leftarrow \mathsf{G}^{\mathsf{Sym}}\big(1^\lambda\big)$
    $C^{pk} \leftarrow \mathsf{E}(S, k_0, t)$
    $C^T \leftarrow T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M)\big)$
    return $C = \big(C^{pk}, [\,], C^T\big)$
end procedure
```

Algorithm 9 $\mathsf{D}^{\mathsf{Hyb}}$ Decryption Pseudocode

```
procedure $\mathsf{D}^{\mathsf{Hyb}}\big(\mathsf{sk}, \mathsf{ku}, C = \big(C^{pk}, \big[C_1^{bks}, \ldots, C_r^{bks}\big], C_r^T = C_{r,1}^T \cdots C_{r,N}^T\big)\big)$
    for $u \leftarrow r, \ldots, 1$ do
        $(s_u, k_u) \leftarrow \mathsf{D}\big(\mathsf{sk}, \mathsf{ku}, C_u^{bks}\big)$
        $\mathsf{ind}_u \leftarrow \mathsf{Ind}(s_u, \ell^*)$
        $\mathsf{str}_u \leftarrow \mathsf{Ctr}(k_u, \ell^*)$
        for $v \leftarrow 1, \ldots, N$ do
            if $v \in \mathsf{ind}_u$ then
                $C_{u-1,v}^T \leftarrow C_{u,v}^T \oplus \mathsf{str}[v]$
            else
                $C_{u-1,v}^T \leftarrow C_{u,v}^T$
            end if
        end for
    end for
    $C^T \leftarrow C_{0,1}^T \cdots C_{0,N}^T$
    $k_0 \leftarrow \mathsf{D}\big(\mathsf{sk}, \mathsf{ku}, C^{pk}\big)$
    return $M \leftarrow \mathsf{D}^{\mathsf{Sym}}\big(k_0, T^{-1}\big(C^T\big)\big)$
end procedure
```

12.5 Correctness

A valid ciphertext for message M under attribute set S at time t has the form $C = \big(C^{pk}, \big[C_1^{bks}, \ldots, C_r^{bks}\big], C_r^T\big)$ where

- C^{pk} is an encryption under attribute set S at time t of symmetric key k_0

Algorithm 10 $\mathsf{CTU}^{\mathsf{Hyb}}$ Ciphertext Update Pseudocode

```
procedure $\mathsf{CTU}^{\mathsf{Hyb}}\big(C_{S,t} = \big(C^{pk}, \big[C_1^{bks}, \ldots, C_r^{bks}\big], C^T = C_1^T \cdots C_N^T\big)\big)$
    $\widetilde{C^{pk}} \leftarrow \mathsf{CTU}(C^{pk})$
    for $u \leftarrow 1, \ldots, r$ do
        $\widetilde{C_u^{bks}} \leftarrow \mathsf{CTU}(C_u^{bks})$
    end for
    Choose $s_{r+1}, k_{r+1}$ uniformly at random
    $C_{r+1}^{bks} \leftarrow \mathsf{E}(S, (s_{r+1}, k_{r+1}), t+1)$
    $\mathsf{ind}_{r+1} \leftarrow \mathsf{Ind}(s_{r+1}, \ell^*)$
    $\mathsf{str}_{r+1} \leftarrow \mathsf{Ctr}(k_{r+1}, \ell^*)$
    for $v \leftarrow 1, \ldots, N$ do
        if $v \in \mathsf{ind}_{r+1}$ then
            $\widetilde{C_v^T} \leftarrow C_v^T \oplus \mathsf{str}_{r+1}[v]$
        else
            $\widetilde{C_v^T} \leftarrow C_v^T$
        end if
    end for
    $\widetilde{C^T} \leftarrow \widetilde{C_1^T} \cdots \widetilde{C_N^T}$
    return $C = \Big(\widetilde{C^{pk}}, \Big[\widetilde{C_1^{bks}}, \ldots, \widetilde{C_{r+1}^{bks}}\Big], \widetilde{C^T}\Big)$
end procedure
```

- Each C_u^{bks} is an encryption under attribute set S at time t of (s_u, k_u)
- C_r^T is $T\big(\mathsf{E}^{\mathsf{Sym}}(k, M)\big)$ with the bits at positions $\mathsf{Ind}(s_1)$ encrypted under symmetric key k_1, then the bits at positions $\mathsf{Ind}(s_2, \ell^*)$ encrypted under symmetric key k_2, etc.

The decryption algorithm takes secret key $\mathsf{sk}_{P,id}$ with $P(S) = 1$ and key update information $\mathsf{ku}_{t'}$ with $t' \geq t$; computes $k_0 = \mathsf{D}\big(\mathsf{sk}_{P,\mathsf{id}}, \mathsf{ku}_{t'}, C^{pk}\big)$ and each $(s_u, k_u) = \mathsf{D}\big(\mathsf{sk}_{P,\mathsf{id}}, \mathsf{ku}_{t'}, C_u^{bks}\big)$; computes C_0^T by decrypting the bits of C_r^T at positions $\mathsf{Ind}(s_r, \ell^*)$ using symmetric key k_r to produce C_{r-1}^T, then decrypting the bits of C_{r-1}^T at positions $\mathsf{Ind}(s_{r-1}, \ell^*)$ using symmetric key k_{r-1} to produce C_{r-2}^T, etc., eventually reaching C_0^T; and finally computes $M = \mathsf{D}^{\mathsf{Sym}}\big(k, T^{-1}\big(C^T\big)\big)$.

12.6 Security Theorems

We have the following security theorems, which are the revocable-storage KP-ABE versions of Theorems 1, 2, and 11, respectively. The proofs are analogous to the proofs from the corresponding PRE theorems presented in Appendix B, Section 6, and Appendix D, respectively. See Appendix F for details.

Theorem 7. *Assume the existence of a* RS-KP-ABE*-secure revocable-storage attribute-based encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$*, a CPA-secure symmetric-key encryption scheme* $\Pi_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$*, and an AONT* T*. Then the construction in Section 12.4 is* IB-PRE-CPA*-secure.*

Theorem 8. *Assume the existence of a* $\mathsf{RS\text{-}KP\text{-}ABE}$*-secure revocable-storage attribute-based encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$*, a symmetric-key encryption scheme* $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$*, and an adaptive* ℓ*-AONT* T*. Suppose that for the construction from Section 12.4,* C^T *comprises at least a fraction* $1-\delta$ *of the total size of each ciphertext. Then for any* $\varepsilon < 1$ *with* $\varepsilon > \delta$ *and any* $\ell^* > \frac{\ell}{\varepsilon-\delta}$*, this construction is* $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}RS\text{-}KP\text{-}ABE}$*-secure.*

Theorem 9. *Assume the existence of a* $\mathsf{RS\text{-}KP\text{-}ABE}$*-secure revocable-storage attribute-based encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$*, a symmetric-key encryption scheme* $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$*, and an* ℓ*-AONT* T*. Suppose that for the construction from Section 12.4,* C^T *comprises at least a fraction* $1-\delta$ *of the total size of each ciphertext. Then for any* $\varepsilon < 1$ *with* $\varepsilon > \delta$ *and any* $\ell^* > \frac{\ell}{\varepsilon-\delta}$*, this construction is* $(1-\varepsilon)$-$\mathsf{Stat\text{-}Revoke\text{-}RS\text{-}KP\text{-}ABE}$*-secure.*

13 Conclusions

We identified a problem with most current proxy re-encryption schemes. These schemes do not consider the issue that when hybrid encryption is used, revoked users may have stored symmetric keys and can still use them to decrypt files that they are no longer supposed to be able to access.

To address this issue, we produced a definition of revocable-PRE security. This requires that a user given a large fraction of a file it has access to, as well as a re-encryption of this file to a key it does not have access to, will not be able to learn any information about the plaintext. This captures the notion that a user may have insufficient bandwidth to download all the files it has access to, but it may be able to download part of every file. If the scheme is revocable-PRE secure, the adversary will be unable to combine its partial knowledge of the original file with the re-encrypted file to produce the original plaintext.

We provided an efficient hybrid PRE scheme that uses a public-key PRE scheme, a symmetric encryption scheme, and an all-or-nothing transform. The AONT makes it possible for the proxy re-encryption algorithm to only touch a small fraction of the total bits of the ciphertext while making the old symmetric key insufficient for decryption. If the underlying primitives are secure and the number of re-encrypted bits is high enough compared to the fraction of the original file that the adversary downloaded, then this scheme is provably secure.

References

1. W. Aiello, S. Lodha, and R. Ostrovsky. Fast digital identity revocation. In *CRYPTO '98*, pages 137–152, 1998.
2. Y. Aono, X. Boyen, L. T. Phong, and L. Wang. Key-private proxy re-encryption under LWE. In *INDOCRYPT 2013*, pages 1–18, 2013.
3. G. Ateniese, K. Benson, and S. Hohenberger. Key-private proxy re-encryption. In *CT-RSA '09*, pages 279–294, 2009.

4. G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, Feb. 2006.
5. E. Barker. SP 800-57. Recommendation for key management, Part 1: General (revision 4). Technical report, NIST, Jan. 2016.
6. M. Bellare, D. Kane, and P. Rogaway. Big-key symmetric encryption: Resisting key exfiltration. In *Advances in Cryptology - CRYPTO 2016*, pages 373–402, 2016.
7. M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology - EUROCRYPT'98*, pages 127–144, 1998.
8. A. Boldyreva, V. Goyal, and V. Kumar. Identity-based encryption with efficient revocation. In *CCS '08*, pages 417–426, 2008.
9. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *SIAM Journal on Computing*, 32(3):586–615, 2003.
10. D. Boneh, K. Lewi, H. W. Montgomery, and A. Raghunathan. Key homomorphic PRFs and their applications. In *CRYPTO 2013*, pages 410–428, 2013.
11. V. Boyko. On the security properties of OAEP as an all-or-nothing transform. In *CRYPTO '99*, pages 503–518, 1999.
12. R. Canetti, Y. Dodis, S. Halevi, E. Kushilevitz, and A. Sahai. Exposure-resilient functions and all-or-nothing transforms. In *EUROCRYPT*, pages 453–469, 2000.
13. R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *CCS '07*, pages 185–194, 2007.
14. G. D. Crescenzo, R. J. Lipton, and S. Walfish. Perfectly secure password protocols in the bounded retrieval model. In *TCC*, pages 225–244, 2006.
15. Y. Dodis, A. Sahai, and A. Smith. On perfect and adaptive security in exposure-resilient cryptography. In *EUROCRYPT '01*, pages 301–324, 2001.
16. S. Dziembowski. Intrusion-resilience via the bounded-storage model. In *Theory of Cryptography, TCC 2006*, pages 207–224, 2006.
17. W. C. Garrison, III, A. Shull, S. Myers, and A. J. Lee. On the practicality of cryptographically enforcing dynamic access control policies in the cloud. *Proceedings of the 37th IEEE Symposium on Security and Privacy*, 2016.
18. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS '06*, pages 89–98, 2006.
19. M. Green and G. Ateniese. Identity-based proxy re-encryption. In *ACNS '07*, pages 288–306, 2007.
20. V. Guruswami and A. Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, Jan 2008.
21. W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
22. A. Ivan and Y. Dodis. Proxy cryptography revisited. In *NDSS 2003*. The Internet Society, 2003.
23. M. Jakobsson. On quorum controlled asymmetric proxy re-encryption. In *Public Key Cryptography*, pages 112–121, 1999.
24. J. Katz and Y. Lindell. *Introduction to Modern Cryptography.* Chapman & Hall/CRC, 2007.
25. X. Liang, Z. Cao, H. Lin, and J. Shao. Attribute based proxy re-encryption with delegating capabilities. In *ASIACCS '09*, pages 276–286, 2009.
26. B. Libert and D. Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. In *PKC'08*, pages 360–379, 2008.

27. M. Mambo and E. Okamoto. Proxy cryptosystems: Delegation of the power to decrypt ciphertexts. In *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 1997.
28. S. Micali. Efficient certificate revocation. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.
29. M. Naor and K. Nissim. Certificate revocation and certificate update. In *USENIX Security Symposium*, SSYM'98, pages 17–17, 1998.
30. Open Web Application Security Project. Cryptographic storage cheat sheet, Aug. 2016.
31. Payment Card Industry Security Standards Council. Payment card industry (PCI) data security standard, v3.2, Apr. 2016.
32. L. T. Phong, L. Wang, Y. Aono, M. H. Nguyen, and X. Boyen. Proxy re-encryption schemes with key privacy from LWE. Cryptology ePrint Archive, Report 2016/327, 2016. `http://eprint.iacr.org/2016/327`.
33. R. L. Rivest. All-or-nothing encryption and the package transform. In *In Fast Software Encryption, LNCS*, pages 210–218. Springer-Verlag, 1997.
34. RSA Laboratories. PKCS #1 v2.2: RSA cryptography standard. Technical report, EMC Corporation, Oct. 2012.
35. A. Sahai, H. Seyalioglu, and B. Waters. Dynamic credentials and ciphertext delegation for attribute-based encryption. In *CRYPTO*, pages 199–217, 2012.
36. A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT'05*, pages 457–473, 2005.
37. J. Shao and Z. Cao. Multi-use unidirectional identity-based proxy re-encryption from hierarchical identity-based encryption. *Inf. Sci.*, 206:83–95, Nov. 2012.
38. A. Syalim, T. Nishide, and K. Sakurai. Realizing proxy re-encryption in the symmetric world. In *Informatics Engineering and Information Science*, pages 259–274, 2011.
39. F. Wang, J. Mickens, N. Zeldovich, and V. Vaikuntanathan. Sieve: Cryptographically enforced access control for user data in untrusted clouds. In *NSDI 16*, pages 611–626, 2016.
40. H. Wang, Z. Cao, and L. Wang. Multi-use and unidirectional identity-based proxy re-encryption schemes. *Inf. Sci.*, 180(20):4042–4059, Oct. 2010.
41. D. Watanabe and M. Yoshino. Key update mechanism for network storage of encrypted data. In *CloudCom 2013*, pages 493–498, 2013.

A Symmetric-Key Encryption

Definition 18 (Symmetric-Key Encryption)). *A symmetric-key encryption scheme consists of three probabilistic polynomial time algorithms:*

- $\mathsf{G}^{\mathsf{Sym}}(1^\lambda) \to k$. *Key generation takes the security parameter and generates a secret key.*
- $\mathsf{E}^{\mathsf{Sym}}(k, M) \to C$. *Encryption takes a secret key and a message and generates a ciphertext.*
- $\mathsf{D}^{\mathsf{Sym}}(\mathsf{sk}, C) \to M$. *Decryption takes a secret key and a ciphertext, and returns the underlying message or symbol $\perp$ that represents an invalid ciphertext.*

The correctness requirement is that given $k \leftarrow \mathsf{G}^{\mathsf{Sym}}(1^\lambda)$ and $M \in \{0,1\}$, $\mathsf{D}^{\mathsf{Sym}}\big(k, \mathsf{E}^{\mathsf{Sym}}(k, M)\big) = M$.

A.1 CPA-security for symmetric-key encryption

The basic security notion for symmetric-key encryption is indistinguishability under chosen-plaintext attack (CPA-security):

Definition 19 (Symmetric-Key Encryption CPA-Security Game [24]). *The symmetric-key encryption* $\mathsf{IND\text{-}CPA}$ *game is defined in Alg. 11.*

Algorithm 11 Symmetric-Key Encryption CPA-Security

```
experiment $\mathsf{IND\text{-}CPA}_{\mathcal{A},\Pi_{sym}}(\lambda)$
    $b \leftarrow \{0,1\}$
    $k \leftarrow \mathsf{G}^{\mathsf{Sym}}(\lambda)$
    $(M_0, M_1, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{enc}}}(\lambda)$
    $C \leftarrow \mathsf{E}^{\mathsf{Sym}}(k, M_b)$
    $b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{enc}}}(\sigma, C)$
    Output 1 iff $b = b'$
end experiment
```

We refer to (M_0, M_1) *as the challenge query and* C *as the challenge ciphertext.*

Definition 20. *A symmetric-key encryption scheme* Π_{sym} *is CPA-secure if for all oracle P.P.T. adversaries A, there exists a negligible function* negl *s.t.:*

$$\Pr[\mathsf{IND\text{-}CPA}_{A,\Pi_{sym}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{negl}(\lambda)$$

B PRE-CPA-Security of Hybrid Construction

Theorem 10. *Assume the existence of a* $\mathsf{PRE\text{-}CPA}$*-secure public-key proxy re-encryption scheme* $\Pi = (\mathsf{G}, \mathsf{RG}, \mathsf{E}, \mathsf{RE}, \mathsf{D})$*, a CPA-secure symmetric-key encryption scheme* $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$*, and an AONT* T*. Then the construction of* Π_{hyb} *in Section 6.3 is* $\mathsf{PRE\text{-}CPA}$*-secure.*

We show that by the security of the proxy re-encryption scheme, the real game Game_{real} is computationally indistinguishable from a hybrid game Game_{hyb} where the challenge ciphertext is

$$C^* = \left(\mathsf{E}(pk_{i^*}, 0), [\,], T\left(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\right)\right) \ .$$

Then we show that by the security of the symmetric-key encryption scheme, in Game_{hyb} the adversary's success rate is less than $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl.

Lemma 7. *Suppose that an adversary* $\mathcal{A}_1$ *for the* $\mathsf{PRE\text{-}CPA}$ *security game has probability of success* p_0 *in* Game_{real} *and probability of success* p_1 *in* Game_{hyb}*. If the underlying proxy re-encryption scheme* Π *is* $\mathsf{PRE\text{-}CPA}$*-secure, then* $p_1 - p_0$ *is negligible.*

Proof. Given an adversary $\mathcal{A}_1$ as described for the Π_{hyb} construction, we show how to construct an adversary $\mathcal{B}_1$ for the original scheme Π that plays the $\mathsf{PRE\text{-}CPA}$ security game. As is typical, $\mathcal{B}_1$ simulates $\mathcal{A}_1$ and its version of the $\mathsf{PRE\text{-}CPA}$ game. First, $\mathcal{B}_1$ gives its simulation of $\mathcal{A}_1$ the public parameters it receives. $\mathcal{B}_1$ responds to queries from $\mathcal{A}_1$ as follows:

- Uncorrupted Key Generation query $\mathcal{O}_{\mathsf{ukey}}$: $\mathcal{B}_1$ makes an uncorrupted key generation query to its oracle and receives pk_i in response, which is sent to $\mathcal{A}_1$.
- Corrupted Key Generation query $\mathcal{O}_{\mathsf{ckey}}$: $\mathcal{B}_1$ makes a corrupted key generation query to its oracle and receives $(\mathsf{pk}_i, \mathsf{sk}_i)$ in response, which is sent to $\mathcal{A}_1$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(i,j)$: $\mathcal{B}_1$ queries (i,j) to its re-encryption key generation oracle, receiving $\mathsf{rk}'_{i\to j}$ in response. Note that $\mathcal{B}_1$ will have pk_i from a previous key generation query. $\mathcal{B}_1$ sends $\mathsf{rk}_{i\to j} = \big(\mathsf{pk}_j, \mathsf{rk}'_{i\to j}\big)$ to $\mathcal{A}_1$.
- Re-Encryption query $\mathcal{O}_{\mathsf{renc}}(i,j,C)$: $\mathcal{B}_1$ queries (i,j) to its re-encryption key generation oracle, and receives $\mathsf{rk}'_{i\to j}$. $\mathcal{B}_1$ then runs $\mathsf{RE}^{\mathsf{Hyb}}\big((\mathsf{pk}_j, \mathsf{rk}'_{i\to j}), C\big)$ from Algorithm 4 and sends the result to $\mathcal{A}_1$.
- Challenge query $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, i^*)$: $\mathcal{B}_1$ generates a random bit b and key $k_0 \leftarrow \mathsf{G}^{\mathsf{Sym}}$. $\mathcal{B}_1$ then makes its own challenge query $(k_0, 0, i^*)$, receiving ciphertext C' in response. $\mathcal{B}_1$ gives $\mathcal{A}_1$ the following challenge ciphertext:

$$C^* = \big(C', [\,], T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)\big) \ .$$

- Guess b': If $b = b'$ then $\mathcal{B}_1$ guesses that C' is an encryption of k_0, otherwise $\mathcal{B}_1$ guesses that C' is an encryption of 0.

Observe that if C' is an encryption of k_0, then $\mathcal{A}_1$ is executing in a perfect simulation of Game_{real}, and thus has probability of success p_0; hence $\mathcal{B}_1$ is correct with probability p_0. If C' is an encryption of 0 then $\mathcal{A}_1$ is simulated in the Game_{hyb} and has probability of success p_1; thus $\mathcal{B}_1$ is correct with probability $1 - p_1$. Therefore $\mathcal{B}_1$'s probability of success is $\frac{1}{2}(p_0 + 1 - p_1) = \frac{1}{2} + \frac{1}{2}(p_0 - p_1)$. By the $\mathsf{PRE\text{-}CPA}$ security of $\Pi = (\mathsf{G}, \mathsf{RG}, \mathsf{E}, \mathsf{RE}, \mathsf{D})$, $\frac{1}{2}(p_0 - p_1)$ is negligible, and so $p_0 - p_1$ is negligible. □

Lemma 8. *Suppose that an adversary $\mathcal{A}_2$ for the* $\mathsf{PRE\text{-}CPA}$ *security game has probability of success p_1 in* Game_{hyb}. *If the underlying symmetric-key encryption scheme Π_{sym} is CPA-secure, then $p_1 < \frac{1}{2} + \mathsf{negl}$ for some negligible function* negl.

Proof. We construct an adversary $\mathcal{B}_2$ for Π_{sym} that plays the CPA security game by simulating adversary $\mathcal{A}_2$ in the $\mathsf{PRE\text{-}CPA}$ game. $\mathcal{B}_2$ instantiates the PRE itself and sends $\mathcal{A}_2$ the public parameters. $\mathcal{B}_2$ responds to queries from the simulation of $\mathcal{A}_2$ as follows:

- Uncorrupted Key Generation query $\mathcal{O}_{\mathsf{ukey}}$: $\mathcal{B}_2$ creates a key pair $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{G}$ and sends pk_i to $\mathcal{A}_2$.

- Corrupted Key Generation query $\mathcal{O}_{\mathsf{ckey}}$: $\mathcal{B}_2$ creates a key pair $(\mathsf{pk}_i, \mathsf{sk}_i) \leftarrow \mathsf{G}$ and sends $(\mathsf{pk}_i, \mathsf{sk}_i)$ to $\mathcal{A}_2$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(i, j)$: If $i = j$ or if i is uncorrupted and j is corrupted then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{rk}'_{i\to j} \leftarrow \mathsf{RG}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_j, \mathsf{sk}_j)$. $\mathcal{B}_2$ sends $\mathsf{rk}_{i\to j} = \big(\mathsf{pk}_j, \mathsf{rk}'_{i\to j}\big)$ to $\mathcal{A}_2$.
- Re-Encryption query $\mathcal{O}_{\mathsf{renc}}(i, j, C)$: If $i = j$ or if i is uncorrupted and j is corrupted then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{rk}'_{i\to j} \leftarrow \mathsf{RG}(\mathsf{pk}_i, \mathsf{sk}_i, \mathsf{pk}_j, \mathsf{sk}_j)$. $\mathcal{B}_2$ then runs $\mathsf{RE}^{\mathsf{Hyb}}\big(\big(\mathsf{pk}_j, \mathsf{rk}'_{i\to j}\big), C\big)$ from Algorithm 4 and sends the result to $\mathcal{A}_2$.
- Challenge query $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, i^*)$: If i^* is corrupted then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ makes challenge query (M_0, M_1), receiving ciphertext C' in response. $\mathcal{B}_2$ gives $\mathcal{A}_2$ challenge ciphertext $C^* = (\mathsf{E}(pk_{i^*}, 0), [\,], T(C'))$.
- Guess b': $\mathcal{B}_2$ receives b' as $\mathcal{A}_2$'s guess and uses the same bit b' for its guess as well.

Since $\mathcal{A}_2$ is in a perfect simulation of Game_{hyb}, its probability of success is p_1. $\mathcal{B}_2$ is correct when $\mathcal{A}_2$ is correct, so its probability of success is also p_1. By the CPA security of $\Pi_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$, $p_1 < \frac{1}{2} + \mathsf{negl}$ for some negligible function negl. □

Now we use these lemmas to prove Theorem 1.

Proof. By Lemmas 7, we see that the adversary's probability of success in Game_{real} can only be negligibly greater than its probability of success in Game_{hyb}. By Lemma 8, the adversary's success in Game_{hyb} can be at most $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl. Thus the adversary's success in Game_{real} can be at most $\frac{1}{2} + \mathsf{negl}'$ for some negligible function negl'. □

C Probabilistic Technical Lemmas

. This appendix contains several useful facts from probability theory. The proofs use standard techniques.

Lemma 9. *Suppose that there are N possible balls. In the first stage, a fraction ε' are selected (by any method). In the second stage, exactly ℓ^* balls are selected uniformly at random. Let ℓ' be the number of balls selected in both stages. Then for any $t > 0$,*

$$\Pr[\ell' \leq \ell^*(\varepsilon' - t)] \leq e^{-2\ell^* t^2}$$

Proof. This bound is based on Hoeffding's inequality [21], which applies for sampling without replacement. Let $x_1, \ldots, x_N$ correspond to the balls, where $x_i = 0$ if the ith ball is selected in the first stage and $x_i = 1$ otherwise. Then $\frac{1}{N}\sum_{i=1}^{N} x_i = 1 - \varepsilon'$. Let $\{X_1, \ldots, X_{\ell^*}\}$ be a subset of $\{x_1, \ldots, x_N\}$ of size ℓ^* chosen uniformly at random, representing the ℓ^* balls selected in the second stage. Let $S_{\ell^*} = \sum_{i=1}^{\ell^*} X_i$ be the number of balls selected in the second stage that were not selected in the first. Then $\ell' = \ell^* - S_{\ell^*}$ is the number of balls selected in both stages.

By Theorem 1 of [21],

$$\Pr[S_{\ell^*} - \ell^*(1-\varepsilon') \geq \ell^* t] \leq e^{-2\ell^* t^2}$$

To get a bound on ℓ', note that

$$S_{\ell^*} - \ell^*(1-\varepsilon') \geq \ell^* t \iff \ell^* - \ell' - \ell^*(1-\varepsilon') \geq \ell^* t \iff \ell' \leq \ell^*(\varepsilon' - t) \ .$$

Thus we have $\Pr[\ell' \leq \ell^*(\varepsilon' - t)] \leq e^{-2\ell^* t^2}$. □

Lemma 10. *Let A_0, B_0 and A_1, B_1 be events in separate probabilistic experiments, where the probability of $B_b \leq \varepsilon$. Then,* $\Pr[A_0] - \Pr[A_1] \leq \Pr[A_0|\overline{B_0}] - \Pr[A_1|\overline{B_1}] + 2\varepsilon$.

Proof.

$$\begin{aligned}
\Pr[A_0] - \Pr[A_1] &= (\Pr[A_0|\overline{B_0}]\Pr[\overline{B_0}] + \Pr[A_0|B_0]\Pr[B_0]) \\
&\quad - (\Pr[A_1|\overline{B_1}]\Pr[\overline{B_1}] + \Pr[A_1|B_1]\Pr[B_1]) \\
&= (\Pr[A_0|\overline{B_0}](1-\Pr[B_0]) + \Pr[A_0|B_0]\Pr[B_0]) \\
&\quad - (\Pr[A_1|\overline{B_1}](1-\Pr[B_1]) + \Pr[A_1|B_1]\Pr[B_1]) \\
&= (\Pr[A_0|\overline{B_0}] - \Pr[A_1|\overline{B_1}] \\
&\quad - \Pr[A_0|\overline{B_0}]\Pr[B_0]) + \Pr[A_1|\overline{B_1}]\Pr[B_1] \\
&\quad + \Pr[A_0|B_0]\Pr[B_0] - \Pr[A_1|B_1]\Pr[B_1] \\
&\leq (\Pr[A_0|\overline{B_0}] - \Pr[A_1|\overline{B_1}]) + \Pr[A_1|\overline{B_1}]\Pr[B_1] + \Pr[A_0|B_0]\Pr[B_0] \\
&< \Pr[A_0|\overline{B_0}] - \Pr[A_1|\overline{B_1}] + \Pr[B_1] + \Pr[B_0] \\
&< \Pr[A_0|\overline{B_0}] - \Pr[A_1|\overline{B_1}] + 2\varepsilon.
\end{aligned}$$

Lemma 11. *Let A_0, B_0 and A_1, B_1 be events in separate probabilistic experiments, where the probability of $B_b \leq \varepsilon$. Then,* $\Pr[A_0|\overline{B_0}] - \Pr[A_1|\overline{B_1}] \leq \Pr[A_0] - \Pr[A_1] + 2\varepsilon$.

Proof. Is an immediate result of the following two claims.

Claim.

$$\Pr[A|B] \leq \Pr[A] + \Pr[\overline{B}]]$$

Proof.

$$\begin{aligned}
\Pr[A] &= \Pr[A|B]\Pr[B] + \Pr[A|\overline{B}]\Pr[\overline{B}] \\
&\geq \Pr[A|B]\Pr[B] \\
&= \Pr[A|B](1-\Pr[\overline{B}]) \\
&= \Pr[A|B] - \Pr[A|B]\Pr[\overline{B}]) \\
&\geq \Pr[A|B] - \Pr[\overline{B}])
\end{aligned}$$

Claim.

$$\Pr[A|B] \geq \Pr[A] - \Pr[\overline{B}]$$

Proof. The law of total probability gives

$$\begin{aligned}\Pr[A|B] &\geq \Pr[A|B]\Pr[B] \\ &= \Pr[A] - \Pr[A|\overline{B}]\Pr[\overline{B}] \\ &\geq \Pr[A] - \Pr[\overline{B}]\end{aligned}$$

D Proof of Static-Revocable-PRE-CPA Security

Theorem 11. *Assume the existence of a* PRE-CPA*-secure public-key proxy re-encryption scheme* $\mathit{\Pi} = (\mathsf{G}, \mathsf{RG}, \mathsf{E}, \mathsf{RE}, \mathsf{D})$ *with re-encryption history independence, a symmetric-key encryption scheme* $\mathit{\Pi}_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$*, and an* ℓ*-AONT* T*. Suppose that for the construction from Section 6.3,* C^T *comprises at least a fraction* $1-\delta$ *of the total size of each ciphertext. Then for any* $\varepsilon < 1$ *with* $\varepsilon > \delta$ *and any* $\ell^* > \frac{\ell}{\varepsilon - \delta}$*, this construction is* $(1-\varepsilon)$-Stat-Revoke-PRE-CPA-*secure.*

Proof. This proof is identical to the proof of Theorem 2, with the exception of how $\mathcal{B}_2$ responds to the challenge queries. In the proofs of Lemmas 2, 3, and 4, the adversary sends bitPos with its challenge query instead of requesting the bits of $\{C_u^*\}_{0\leq u\leq r}$ adaptively. In the proof of Lemmma 6 for the static case, it works as follows: $\mathcal{B}_2$ receives challenge query $(M_0, M_1, [i_0^*, \ldots, i_r^*], j^*, \mathsf{bitPos})$. Let bitPos_T be the set of all bit positions of C_0^T that correspond to the vth bit of C_u^* for some (u,v) in bitPos. $\mathcal{B}_2$ chooses random $L \in \left\{ {N \atop \ell^*} \right\}$. If $|L \cap ([N] \setminus \mathsf{bitPos}_T)| < \ell$ then B_2 aborts the simulation. Otherwise, $\mathcal{B}_2$ generates random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$ and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_2$ then constructs $C_0^{pk}, \ldots, C_{r+1}^{pk}$ as well as $C_{u,u}^{bks}, \ldots, C_{u,r+1}^{bks}$ for $u \in \{1, \ldots, r\}$, following the procedure in Lemma 3. $\mathcal{B}_2$ makes AONT challenge query

$$\left(\mathsf{E}^{\mathsf{Sym}}(k_0, M_0), \mathsf{E}^{\mathsf{Sym}}(k_0, M_1), L \cap ([N] \setminus \mathsf{bitPos}_T)\right) ,$$

receiving

$$\left[C^T\right]_{L\cap([N]\setminus\mathsf{bitPos}_T)} = \left[T\left(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\right)\right]_{L\cap([N]\setminus\mathsf{bitPos}_T)}$$

in response. $\mathcal{B}_2$ produces $C_0^T \in \{0,1\}^N$ by taking $\left[C^T\right]_{L\cap([N]\setminus\mathsf{bitPos}_T)}$ and filling in the bits from $L \cap ([N] \setminus \mathsf{bitPos}_T)$ arbitrarily (these bits will not affect the challenge ciphertexts), and then it computes

$$\{C_u^* = \left(C_u^{pk}, \left[C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}\right], C_u^T\right)\}_{0\leq u\leq r}$$

for

$$C_u^T = \left[\cdots\left[\left[C_0^T\right]_{\mathsf{Ind}(s_1,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\right]\cdots\right]_{\mathsf{Ind}(s_u,\ell^*),\mathsf{Ctr}(k_u,\ell^*)} .$$

$\mathcal{B}_2$ constructs $\widetilde{C_r^T} \in \{0,1\}^N$ by setting the bits not in L as their corresponding values from C_r^T and choosing the remaining bits randomly. It gives $\mathcal{A}_4$ the requested $(1-\varepsilon)|C_r^*|$ bits of $\{C_u^*\}_{0 \le u \le r}$ bits as determined by bitPos as well as

$$C^{**} = \Big(C_{r+1}^{pk}, \big[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, \mathsf{E}\big(\mathsf{pk}_{j^*}, (0,0)\big)\big], \widetilde{C_r^T}\Big) \ .$$

When considering the probability that $\mathcal{B}_2$ aborts when responding to $\mathcal{A}_4$'s challenge query, here we let $\ell' = |L \cap ([N] \setminus \mathsf{bitPos}_T)|$, which is the number of bit positions of C_0^T that are in L but not in bitPos_T. Again, $\mathcal{B}_2$ aborts if and only if $\ell' < \ell$. The upper bound on ℓ' is still the same as for the adaptive case by the exact same argument. Thus the probability that $\mathcal{B}_2$ aborts is again negligible as long as $\ell^* > \frac{\ell}{\varepsilon - \delta}$. □

E Proofs for Identity-Based Proxy Re-Encryption

E.1 IB-PRE-CPA-Security of Hybrid Construction

Theorem 12. *Assume the existence of a* IB-PRE-CPA*-secure identity-based proxy re-encryption scheme* $\mathit{\Pi} = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$*, a CPA-secure symmetric-key encryption scheme* $\mathit{\Pi}_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$*, and an AONT* T*. Then the construction of* $\mathit{\Pi}_{hyb}$ *in Section 11.4 is* IB-PRE-CPA*-secure.*

We show that by the security of the proxy re-encryption scheme, the real game Game_{real} is computationally indistinguishable from a hybrid game Game_{hyb} where the challenge ciphertext is

$$C^* = \big(\mathsf{E}(\mathsf{id}^*, 0), [\,], T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)\big) \ .$$

Then we show that by the security of the symmetric-key encryption scheme, in Game_{hyb} the adversary's success rate is less than $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl.

Lemma 12. *Suppose that an adversary* $\mathcal{A}_1$ *for the* IB-PRE-CPA *security game has probability of success* p_0 *in* Game_{real} *and probability of success* p_1 *in* Game_{hyb}*. If the underlying proxy re-encryption scheme* $\mathit{\Pi}$ *is* IB-PRE-CPA*-secure, then* $p_1 - p_0$ *is negligible.*

Proof. Given an adversary $\mathcal{A}_1$ as described for the $\mathit{\Pi}_{hyb}$ construction, we show how to construct an adversary $\mathcal{B}_1$ for the original scheme $\mathit{\Pi}$ that plays the IB-PRE-CPA security game. As is typical, $\mathcal{B}_1$ simulates $\mathcal{A}_1$ and its version of the IB-PRE-CPA game. First, $\mathcal{B}_1$ gives its simulation of $\mathcal{A}_1$ the public parameters it receives. $\mathcal{B}_1$ responds to queries from $\mathcal{A}_1$ as follows:

- Key Generation query $\mathcal{O}_{\mathsf{key}}(\mathsf{id})$: $\mathcal{B}_1$ queries id to its key generation oracle and receives $\mathsf{sk}_{\mathsf{id}}$ in response, which is sent to $\mathcal{A}_1$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(\mathsf{id}_i, \mathsf{id}_j)$: $\mathcal{B}_1$ queries $(\mathsf{id}_i, \mathsf{id}_j)$ to its re-encryption key generation oracle and receives $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j}$ in response, which is sent to $\mathcal{A}_1$.

- Challenge query $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, \mathsf{id}^*)$: $\mathcal{B}_1$ generates a random bit b and key $k_0 \leftarrow \mathsf{G}^{\mathsf{Sym}}$. $\mathcal{B}_1$ then makes its own challenge query $(k_0, 0, \mathsf{id}^*)$, receiving ciphertext C' in response. $\mathcal{B}_1$ gives $\mathcal{A}_1$ the following challenge ciphertext:

$$C^* = \big(C', [\,], T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)\big) \ .$$

- Guess b': If $b = b'$ then $\mathcal{B}_1$ guesses that C' is an encryption of k_0, otherwise $\mathcal{B}_1$ guesses that C' is an encryption of 0.

Observe that if C' is an encryption of k_0, then $\mathcal{A}_1$ is executing in a perfect simulation of Game_{real}, and thus has probability of success p_0; hence $\mathcal{B}_1$ is correct with probability p_0. If C' is an encryption of 0 then $\mathcal{A}_1$ is simulated in the Game_{hyb} and has probability of success p_1; thus $\mathcal{B}_1$ is correct with probability $1-p_1$. Therefore $\mathcal{B}_1$'s probability of success is $\frac{1}{2}(p_0+1-p_1) = \frac{1}{2}+\frac{1}{2}(p_0-p_1)$. By the IB-PRE-CPA security of $\mathit{\Pi} = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$, $\frac{1}{2}(p_0-p_1)$ is negligible, and so $p_0 - p_1$ is negligible. □

Lemma 13. *Suppose that an adversary $\mathcal{A}_2$ for the* PRE-CPA *security game has probability of success p_1 in* Game_{hyb}*. If the underlying symmetric-key encryption scheme $\mathit{\Pi}_{sym}$ is CPA-secure, then $p_1 < \frac{1}{2} + \mathsf{negl}$ for some negligible function* negl*.*

Proof. We construct an adversary $\mathcal{B}_2$ for $\mathit{\Pi}_{sym}$ that plays the CPA security game by simulating adversary $\mathcal{A}_2$ in the PRE-CPA game. $\mathcal{B}_2$ instantiates the IB-PRE scheme itself by computing $(\mathsf{param}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends param to $\mathcal{A}_2$. $\mathcal{B}_2$ responds to queries from the simulation of $\mathcal{A}_2$ as follows:

- Key Generation query $\mathcal{O}_{\mathsf{key}}(\mathsf{id})$: If responding to the query would allow $\mathcal{A}_2$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, \mathsf{id})$ and sends $\mathsf{sk}_{\mathsf{id}}$ to $\mathcal{A}_2$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(\mathsf{id}_i, \mathsf{id}_j)$: If $\mathsf{id}_i = \mathsf{id}_j$ or if responding to the query would allow $\mathcal{A}_2$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j} \leftarrow \mathsf{RG}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_j)$ and sends $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j}$ to $\mathcal{A}_2$.
- Challenge query $\mathcal{O}_{\mathsf{chal}}(M_0, M_1, \mathsf{id}^*)$: If $\mathcal{A}_2$ can trivially decrypt ciphertexts encrypted under id^* then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ makes challenge query (M_0, M_1), receiving ciphertext C' in response. $\mathcal{B}_2$ gives $\mathcal{A}_2$ challenge ciphertext $C^* = (\mathsf{E}(\mathsf{id}^*, 0), [\,], T(C'))$.
- Guess b': $\mathcal{B}_2$ receives b' as $\mathcal{A}_2$'s guess and uses the same bit b' for its guess as well.

Since $\mathcal{A}_2$ is in a perfect simulation of Game_{hyb}, its probability of success is p_1. $\mathcal{B}_2$ is correct when $\mathcal{A}_2$ is correct, so its probability of success is also p_1. By the CPA security of $\mathit{\Pi}_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$, $p_1 < \frac{1}{2} + \mathsf{negl}$ for some negligible function negl. □

Now we use these lemmas to prove Theorem 12.

Proof. By Lemmas 12, we see that the adversary's probability of success in Game_{real} can only be negligibly greater than its probability of success in Game_{hyb}. By Lemma 13, the adversary's success in Game_{hyb} can be at most $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl. Thus the adversary's success in Game_{real} can be at most $\frac{1}{2} + \mathsf{negl}'$ for some negligible function negl'. □

E.2 Adaptive-Revocable-IB-PRE-CPA Security

Theorem 13. *Assume the existence of a* IB-PRE-CPA*-secure identity-based proxy re-encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$ *with re-encryption history independence, a symmetric-key encryption scheme* $\Pi_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$*, and an adaptive* ℓ*-AONT* T*. Suppose that for the construction from Section 11.4,* C^T *comprises at least a fraction* $1 - \delta$ *of the total size of each ciphertext. Then for any* $\varepsilon < 1$ *with* $\varepsilon > \delta$ *and any* $\ell^* > \frac{\ell}{\varepsilon - \delta}$*, this construction is* $(1 - \varepsilon)$-Adap-Revoke-IB-PRE-CPA*-secure.*

We show the computational indistinguishability of a series of games. Each game is the same as the real game except in regards to C^{**}. In each game, the challenge query is $(M_0, M_1, [\mathsf{id}_0^*, \ldots, \mathsf{id}_r^*], \mathsf{id}^{**})$. We highlight in bold the portions of C^{**} that differ from the previous games, in the descriptions of the respective games that follow:

Game_0*:* This is the real game, where:

$$C^{**} = \Big(\mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_r^* \to \mathsf{id}^{**}}, C^{pk}\big), \big[C_1^{bks}, \ldots, C_r^{bks}, \mathsf{E}(\mathsf{id}^{**}, (s, k_1))\big], \big[C^T\big]_{\mathsf{Ind}(s,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\Big) .$$

Game_1*:* This is identical to Game_0 except that we replace $\mathsf{E}(\mathsf{id}^{**}, (s, k_1))$ with $\mathsf{E}(\mathsf{id}^{**}, (0, 0))$, resulting in:

$$C^{**} = \Big(\mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_r^* \to \mathsf{id}^{**}}, C^{pk}\big), \big[C_1^{bks}, \ldots, C_r^{bks}, \mathbf{E(id^{**}, (0, 0))}\big], \big[C^T\big]_{\mathsf{Ind}(s,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\Big) .$$

Game_2*:* This is identical to Game_1 except that we replace the pseudorandom $\mathsf{Ind}(s, \ell^*)$ with truly random $\mathsf{rInd}(\ell^*)$, resulting in:

$$C^{**} = \Big(\mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_r^* \to \mathsf{id}^{**}}, C^{pk}\big), \big[C_1^{bks}, \ldots, C_r^{bks}, \mathsf{E}(\mathsf{id}^{**}, (0, 0))\big], \big[C^T\big]_{\mathbf{rInd}(\boldsymbol{\ell^*}),\mathsf{Ctr}(k_1,\ell^*)}\Big) .$$

Game_3*:* This is identical to Game_2 except that we replace the keystream $\mathsf{Ctr}(k_1, \ell^*)$ we get from counter mode encryption with a random string $\mathsf{rStr}(\ell^*)$, resulting in:

$$C^{**} = \Big(\mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_r^* \to \mathsf{id}^{**}}, C^{pk}\big), \big[C_1^{bks}, \ldots, C_r^{bks}, \mathsf{E}(\mathsf{id}^{**}, (0, 0))\big], \big[C^T\big]_{\mathsf{rInd}(\ell^*),\mathbf{rStr}(\boldsymbol{\ell^*})}\Big) .$$

We now provide a series of lemmas that show that any adverary's probabilities of success in two successive games are negligibly close. These are presented in Lemmas 14, 15, and 16. Finally, we show in Lemma 17 that any adversary's chance of success in the final game is negligibly close to 1/2.

Lemma 14. *Suppose that an adversary $\mathcal{A}_1$ for the* $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}IB\text{-}PRE\text{-}CPA}$ *security game has probability of success p_0 in* Game_0 *and probability of success p_1 in* Game_1*. If the underlying proxy re-encryption scheme Π is* $\mathsf{IB\text{-}PRE\text{-}CPA}$-*secure, then $p_1 - p_0$ is negligible.*

Proof. We construct an adversary $\mathcal{B}_1$ that plays the $\mathsf{IB\text{-}PRE\text{-}CPA}$ security game by simulating adversary $\mathcal{A}_1$. $\mathcal{B}_1$ gives $\mathcal{A}_1$ the public parameters it receives. $\mathcal{B}_1$ responds to queries from $\mathcal{A}_1$ as follows:

- Key Generation query $\mathcal{O}_{\mathsf{key}}(\mathsf{id})$: $\mathcal{B}_1$ queries id to its key generation oracle and receives $\mathsf{sk}_{\mathsf{id}}$ in response, which is sent to $\mathcal{A}_1$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(\mathsf{id}_i, \mathsf{id}_j)$: $\mathcal{B}_1$ queries $(\mathsf{id}_i, \mathsf{id}_j)$ to its re-encryption key generation oracle and receives $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j}$ in response, which is sent to $\mathcal{A}_1$.
- Challenge query $(M_0, M_1, [\mathsf{id}^*_0, \ldots, \mathsf{id}^*_r], \mathsf{id}^{**})$: $\mathcal{B}_1$ generates random bit b, random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_1$ also selects r fresh random identities $\mathsf{id}'_0, \ldots, \mathsf{id}'_{r-1}$. If these are later used in queries made by $\mathcal{A}_1$, then this may lead to $\mathcal{B}_1$ making a prohibited query in response to a valid query from $\mathcal{A}_1$, forcing $\mathcal{B}_1$ to abort. However, if I is the length of each identity, then the probability that a particular identity chosen by $\mathcal{A}_1$ is one of $\mathsf{id}'_0, \ldots, \mathsf{id}'_{r-1}$ is at most $q2^{-I}$. If $\mathcal{A}_1$ makes at most q queries, then since each query involves at most two identities, the probability that $\mathcal{B}_1$ aborts is at most $2rq2^{-I} = rq2^{-I+1}$. $\mathcal{B}_1$ creates $C^{pk}_0 = \mathsf{E}(\mathsf{id}^*_0, k_0)$ and $\hat{C}^{pk}_0 = \mathsf{E}\big(\mathsf{id}', k_0\big)$, and then queries

$$
\begin{aligned}
\mathsf{rk}_{\mathsf{id}'_0 \to \mathsf{id}^*_1} &= \mathcal{O}_{\mathsf{rkey}}\big(\mathsf{id}'_0, \mathsf{id}^*_1\big), &\qquad \mathsf{rk}_{\mathsf{id}'_0 \to \mathsf{id}'_1} &= \mathcal{O}_{\mathsf{rkey}}\big(\mathsf{id}'_0, \mathsf{id}'_1\big),\\
\mathsf{rk}_{\mathsf{id}'_1 \to \mathsf{id}^*_2} &= \mathcal{O}_{\mathsf{rkey}}\big(\mathsf{id}'_1, \mathsf{id}^*_2\big), & \mathsf{rk}_{\mathsf{id}'_1 \to \mathsf{id}'_2} &= \mathcal{O}_{\mathsf{rkey}}\big(\mathsf{id}'_1, \mathsf{id}'_2\big), \ \ldots\\
\mathsf{rk}_{\mathsf{id}'_{r-1} \to \mathsf{id}^*_r} &= \mathcal{O}_{\mathsf{rkey}}\big(\mathsf{id}'_{r-1}, \mathsf{id}^*_r\big), & \mathsf{rk}_{\mathsf{id}'_{r-1} \to \mathsf{id}'_r} &= \mathcal{O}_{\mathsf{rkey}}\big(\mathsf{id}'_{r-1}, \mathsf{id}'_r\big),\\
\mathsf{rk}_{\mathsf{id}'_r \to \mathsf{id}^{**}} &= \mathcal{O}_{\mathsf{rkey}}\big(\mathsf{id}'_r, \mathsf{id}^{**}\big)
\end{aligned}
$$

and computes

$$
\begin{aligned}
C^{pk}_1 &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_0 \to \mathsf{id}^*_1}, \hat{C}^{pk}_0\Big), &\qquad \hat{C}^{pk}_1 &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_0 \to \mathsf{id}'_1}, \hat{C}^{pk}_0\Big),\\
C^{pk}_2 &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_1 \to \mathsf{id}^*_2}, \hat{C}^{pk}_1\Big), & \hat{C}^{pk}_2 &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_1 \to \mathsf{id}'_2}, \hat{C}^{pk}_1\Big), \ \ldots\\
C^{pk}_r &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_{r-1} \to \mathsf{id}^*_r}, \hat{C}^{pk}_{r-1}\Big), & \hat{C}^{pk}_r &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_{r-1} \to \mathsf{id}'_r}, \hat{C}^{pk}_{r-1}\Big),\\
C^{pk}_{r+1} &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_r \to \mathsf{id}^{**}}, \hat{C}^{pk}_r\Big)
\end{aligned}
$$

Note that each C^{pk}_u is produced by encrypting k_0 under id'_0 and then re-encrypting it through $[\mathsf{id}'_1, \mathsf{id}'_{u-1}, \mathsf{id}^*_u]$. By re-encryption history independence, this is indistinguishable from a ciphertext produced by encrypting k_0 under i^*_0 and then re-encrypting it through $[\mathsf{id}^*_1, \mathsf{id}^*_{u-1}, \mathsf{id}^*_u]$.

For each $u \in \{1,\ldots,r\}$, $\mathcal{B}_1$ creates $C^{bks}_{u,u} = \mathsf{E}(\mathsf{id}^*_u,(s_u,k_u))$ and $\hat{C}^{bks}_{u,u} = \mathsf{E}\big(\mathsf{id}'_u,(s_u,k_u)\big)$, and then queries

$$\begin{aligned}
C^{bks}_{u,u+1} &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_u \to \mathsf{id}^*_{u+1}}, \hat{C}^{bks}_{u,u}\Big), & \hat{C}^{bks}_{u,u+1} &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_u \to \mathsf{id}'_{u+1}}, \hat{C}^{bks}_{u,u}\Big),\\
C^{bks}_{u,u+2} &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_{u+1} \to \mathsf{id}^*_{u+2}}, \hat{C}^{bks}_{u,u+1}\Big), & \hat{C}^{bks}_{u,u+2} &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_{u+1} \to \mathsf{id}'_{u+2}}, \hat{C}^{bks}_{u,u+1}\Big), \ \ldots\\
C^{bks}_{u,r} &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_{r-1} \to \mathsf{id}^*_{r}}, \hat{C}^{bks}_{u,r-1}\Big), & \hat{C}^{bks}_{u,r} &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_{r-1} \to \mathsf{id}'_{r}}, \hat{C}^{bks}_{u,r-1}\Big),\\
C^{bks}_{u,r+1} &= \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}'_{r} \to \mathsf{id}^*_{r+1}}, \hat{C}^{bks}_{u,r}\Big)
\end{aligned}$$

Note that each $C^{bks}_{u,v}$ is produced by encrypting (s_u,k_u) under id'_u and then re-encrypting it through $[\mathsf{id}'_u, \mathsf{id}'_{v-1}, \mathsf{id}^*_v]$. By re-encryption history independence, this is indistinguishable from a ciphertext produced by encrypting (s_u,k_u) under id^*_u and then re-encrypting it through $[\mathsf{id}^*_u, \mathsf{id}^*_{v-1}, \mathsf{id}^*_v]$.

Then $\mathcal{B}_1$ creates $C^T_0 = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0,M_b)\big)$ and $C^*_0 = \Big(C^{pk}_0, [\,], C^T_0\Big)$. For each $u \in \{1,\ldots,r\}$, $\mathcal{B}_1$ creates

$$C^T_u = \big[C^T_{u-1}\big]_{\mathsf{Ind}(s_u,\ell^*),\mathsf{Ctr}(k_u,\ell^*)}$$

and

$$C^*_u = \big(C^{pk}_u, \big[C^{bks}_{1,u},\ldots,C^{bks}_{u,u}\big], C^T_u\big) \ .$$

$\mathcal{B}_1$ makes challenge query $((s_{r+1},k_{r+1}),(0,0),j^*)$, receiving ciphertext C' in response. It gives $\mathcal{A}_1$ the $(1-\varepsilon)|C^*_r|$ bits that it adaptively requests of $\{C^*_u\}_{0\le u\le r}$ as well as

$$C^{**} = \Big(C^{pk}_{r+1}, \big[C^{bks}_{1,r+1},\ldots,C^{bks}_{r,r+1},C'\big], \big[C^T_r\big]_{\mathsf{Ind}(s_{r+1},\ell^*),\mathsf{Ctr}(k_{r+1},\ell^*)}\Big) \ .$$

- Guess b': If $b = b'$ then $\mathcal{B}_1$ guesses that C' is an encryption of (s_{r+1},k_{r+1}), otherwise $\mathcal{B}_1$ guesses that C' is an encryption of $(0,0)$.

If C' is an encryption of (s_{r+1},k_{r+1}), then $\mathcal{A}_1$ is in Game_0 (the real game) and has probability of success p_0; thus $\mathcal{B}_1$ is correct with probability p_0. If C' is an encryption of $(0,0)$ then $\mathcal{A}_1$ is in Game_1 and has probability of success p_1; thus $\mathcal{B}_1$ is correct with probability $1-p_1$. Since $\mathcal{B}_1$ aborts with probability at most $rq2^{-I+1}$, its probability of success is therefore at least $\frac{1}{2}(p_0+1-p_1) - rq2^{-I+1} = \frac{1}{2}+\frac{1}{2}(p_0-p_1) - rq2^{-I+1}$. By the $\mathsf{IB\text{-}PRE\text{-}CPA}$ security of $(\mathsf{G},\mathsf{RG},\mathsf{E},\mathsf{RE},\mathsf{D})$, $\frac{1}{2}(p_0 - p_1) - rq2^{-I+1}$ is negligible, and since $rq2^{-I+1}$ is negligible, so is $p_0 - p_1$. □

Lemma 15. *Suppose that an adversary $\mathcal{A}_2$ for the $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}IB\text{-}PRE\text{-}CPA}$ security game has probability of success p_1 in Game_1 and probability of success p_2 in Game_2. If $\mathsf{Ind}(s,\ell^*)$ with random seed s is pseudorandom (indistinguishable from $\mathsf{rInd}(\ell^*)$), then $p_2 - p_1$ is negligible.*

Proof. We construct a distinguisher $\mathcal{D}_1$ that receives a set of indices ind—either $\mathsf{ind} = \mathsf{Ind}(s,\ell^*)$ with random seed s or $\mathsf{ind} = \mathsf{rInd}(\ell^*)$. $\mathcal{D}_1$ simulates adversary $\mathcal{A}_2$. $\mathcal{D}_1$ instantiates the IB-PRE scheme itself and sends $\mathcal{A}_2$ the public parameters. $\mathcal{D}_1$ responds to queries from $\mathcal{A}_2$ as follows:

- Key Generation query $\mathcal{O}_{\mathsf{key}}(\mathsf{id})$: If responding to the query would allow $\mathcal{A}_2$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{D}_1$ sends $\perp$; otherwise, $\mathcal{D}_1$ computes $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, \mathsf{id})$ and sends $\mathsf{sk}_{\mathsf{id}}$ to $\mathcal{A}_2$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(\mathsf{id}_i, \mathsf{id}_j)$: If $\mathsf{id}_i = \mathsf{id}_j$ or if responding to the query would allow $\mathcal{A}_2$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{D}_1$ sends $\perp$; otherwise, $\mathcal{D}_1$ computes $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j} \leftarrow \mathsf{RG}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_j)$ and sends $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j}$ to $\mathcal{A}_2$.
- Challenge query $(M_0, M_1, [\mathsf{id}_0^*, \ldots, \mathsf{id}_r^*], \mathsf{id}^{**})$: If $\mathcal{A}_2$ can trivially decrypt ciphertexts encrypted under id^{**} then $\mathcal{D}_1$ sends $\perp$. Otherwise, $\mathcal{D}_1$ generates random bit b, random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{D}_1$ creates $C_0^{pk} = \mathsf{E}(\mathsf{id}_0^*, k_0)$ and

$$C_1^{pk} = \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}_0^* \to \mathsf{id}_1^*}, C_0^{pk}\Big), C_2^{pk} = \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}_1^* \to \mathsf{id}_2^*}, C_1^{pk}\Big),$$
$$\ldots, C_r^{pk} = \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}_{r-1}^* \to \mathsf{id}_r^*}, C_{r-1}^{pk}\Big), C_{r+1}^{pk} = \mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_r^* \to \mathsf{id}^{**}}, C_r^{pk}\big)$$

For each $u \in \{1, \ldots, r\}$, $\mathcal{D}_1$ creates $C_{u,u}^{bks} = \mathsf{E}(\mathsf{id}_u^*, (s_u, k_u))$ and computes

$$C_{u,u+1}^{bks} = \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}_u^* \to \mathsf{id}_{u+1}^*}, C_{u,u}^{bks}\Big), C_{u,u+2}^{bks} = \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}_{u+1}^* \to \mathsf{id}_{u+2}^*}, C_{u,u+1}^{bks}\Big),$$
$$\ldots, C_{u,r}^{bks} = \mathsf{RE}\Big(\mathsf{rk}_{\mathsf{id}_{r-1}^* \to \mathsf{id}_r^*}, C_{u,r-1}^{bks}\Big), C_{u,r+1}^{bks} = \mathsf{RE}\big(\mathsf{rk}_{\mathsf{id}_r^* \to \mathsf{id}^{**}}, C_{u,r}^{bks}\big)$$

Then $\mathcal{D}_1$ creates $C_0^T = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)$ and $C_0^* = \Big(C_0^{pk}, [\,], C_0^T\Big)$. For each $u \in \{1, \ldots, r\}$, $\mathcal{D}_1$ creates

$$C_u^T = \big[C_{u-1}^T\big]_{\mathsf{Ind}(s_u, \ell^*), \mathsf{Ctr}(k_u, \ell^*)}$$

and

$$C_u^* = \big(C_u^{pk}, \big[C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}\big], C_u^T\big) \ .$$

$\mathcal{D}_1$ gives $\mathcal{A}_2$ the $(1-\varepsilon)|C_r^*|$ bits that it adaptively requests of $\{C_u^*\}_{0 \le u \le r}$ as well as

$$C^{**} = \Big(C_{r+1}^{pk}, \big[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, \mathsf{E}(\mathsf{id}^{**}, (0,0))\big], \big[C_r^T\big]_{\mathsf{ind}, \mathsf{Ctr}(k_{r+1}, \ell^*)}\Big) \ .$$

- Guess b': If $b = b'$ then $\mathcal{D}_1$ guesses that $\mathsf{ind} = \mathsf{Ind}(s, \ell^*)$, otherwise $\mathcal{D}_1$ guesses that $\mathsf{ind} = \mathsf{rInd}(\ell^*)$.

If $\mathsf{ind} = \mathsf{Ind}(s, \ell^*)$, then $\mathcal{A}_2$ is in Game_1 and has probability of success p_1; thus $\mathcal{D}_1$ is correct with probability p_1. If $\mathsf{ind} = \mathsf{rInd}(\ell^*)$, then $\mathcal{A}_2$ is in Game_2 and has probability of success p_2; thus $\mathcal{D}_1$ is correct with probability $1 - p_2$. Therefore $\mathcal{D}_1$'s probability of success is $\frac{1}{2}(p_1 + 1 - p_2) = \frac{1}{2} + \frac{1}{2}(p_1 - p_2)$. By the pseudorandomness of $\mathsf{Ind}(s, \ell^*)$, $\frac{1}{2}(p_1 - p_2)$ is negligible, and so $p_1 - p_2$ is negligible. □

Lemma 16. *Suppose that an adversary $\mathcal{A}_3$ for the $(1-\varepsilon)$-Adap-Revoke-IB-PRE-CPA security game has probability of success p_2 in Game_2 and probability of success p_3 in Game_3. If $\mathsf{Ctr}(k_1, \ell^*)$ with random key k_1 is pseudorandom (indistinguishable from $\mathsf{rStr}(\ell^*)$), then $p_3 - p_2$ is negligible.*

Proof. We construct a distinguisher $\mathcal{D}_2$ that receives a bitstream str—either $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$ with random key k_1 or $\mathsf{str} = \mathsf{rStr}(\ell^*)$. $\mathcal{D}_2$ simulates adversary $\mathcal{A}_3$. $\mathcal{D}_2$ instantiates the IB-PRE scheme itself and sends $\mathcal{A}_3$ the public parameters. $\mathcal{D}_2$ responds to queries from $\mathcal{A}_3$ as follows:

- Key Generation query $\mathcal{O}_{\mathsf{key}}(\mathsf{id})$: If responding to the query would allow $\mathcal{A}_3$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{D}_2$ sends $\perp$; otherwise, $\mathcal{D}_2$ computes $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, \mathsf{id})$ and sends $\mathsf{sk}_{\mathsf{id}}$ to $\mathcal{A}_3$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(\mathsf{id}_i, \mathsf{id}_j)$: If $\mathsf{id}_i = \mathsf{id}_j$ or if responding to the query would allow $\mathcal{A}_3$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{D}_2$ sends $\perp$; otherwise, $\mathcal{D}_2$ computes $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j} \leftarrow \mathsf{RG}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_j)$ and sends $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j}$ to $\mathcal{A}_3$.
- Challenge query $(M_0, M_1, [\mathsf{id}_0^*, \ldots, \mathsf{id}_r^*], \mathsf{id}^{**})$: If $\mathcal{A}_3$ can trivially decrypt ciphertexts encrypted under id^{**} then $\mathcal{D}_2$ sends $\perp$. Otherwise, $\mathcal{D}_2$ generates random bit b, random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{D}_2$ then constructs $C_0^*, \ldots, C_r^*, C_{r+1}^{pk}, \left[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}\right]$, and C_r^T following the procedure in Lemma 15. $\mathcal{D}_2$ gives $\mathcal{A}_3$ the $(1-\varepsilon)|C_r^*|$ bits that it adaptively requests of $\{C_u^*\}_{0 \le u \le r}$ as well as
$$C^{**} = \left(C_{r+1}^{pk}, \left[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, \mathsf{E}(\mathsf{id}^{**}, (0,0))\right], \left[C_r^T\right]_{\mathsf{rInd}(\ell^*),\mathsf{str}}\right) .$$
- Guess b': If $b = b'$ then $\mathcal{D}_2$ guesses that $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$, otherwise $\mathcal{D}_2$ guesses that $\mathsf{str} = \mathsf{rStr}(\ell^*)$.

If $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$, then $\mathcal{A}_2$ is in Game_2 and has probability of success p_2; thus $\mathcal{D}_2$ is correct with probability p_2. If $\mathsf{str} = \mathsf{rStr}(\ell^*)$, then $\mathcal{A}_2$ is in Game_3 and has probability of success p_3; thus $\mathcal{D}_2$ is correct with probability $1 - p_3$. Therefore $\mathcal{D}_2$'s probability of success is $\frac{1}{2}(p_2 + 1 - p_3) = \frac{1}{2} + \frac{1}{2}(p_2 - p_3)$. By the pseudorandomness of $\mathsf{Ctr}(k_1, \ell^*)$, $\frac{1}{2}(p_2 - p_3)$ is negligible, and so $p_2 - p_3$ is negligible. □

Lemma 17. *Suppose that an adversary $\mathcal{A}_4$ for the $(1-\varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}IB\text{-}PRE\text{-}CPA}$ security game has probability of success p_3 in Game_3. Suppose also that the underlying AONT T is a computationally-secure adaptive ℓ-AONT and C^T comprises at least a fraction $1-\delta$ of the total size of each ciphertext. If $\varepsilon > \delta$ and $\ell^* > \frac{\ell}{\varepsilon - \delta}$, then $p_3 < \frac{1}{2} + \mathsf{negl}$ for some negligible function negl.*

Proof. We construct an adversary $\mathcal{B}_2$ that plays the ℓ-AONT security game by simulating adversary $\mathcal{A}_4$. $\mathcal{B}_2$ instantiates the IB-PRE scheme itself and sends $\mathcal{A}_4$ the public parameters. $\mathcal{B}_2$ responds to queries from $\mathcal{A}_4$ as follows:

- Key Generation query $\mathcal{O}_{\mathsf{key}}(\mathsf{id})$: If responding to the query would allow $\mathcal{A}_4$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{sk}_{\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, \mathsf{id})$ and sends $\mathsf{sk}_{\mathsf{id}}$ to $\mathcal{A}_4$.
- Re-Encryption Key Generation query $\mathcal{O}_{\mathsf{rkey}}(\mathsf{id}_i, \mathsf{id}_j)$: If $\mathsf{id}_i = \mathsf{id}_j$ or if responding to the query would allow $\mathcal{A}_4$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j} \leftarrow \mathsf{RG}(\mathsf{sk}_{\mathsf{id}_i}, \mathsf{id}_i, \mathsf{id}_j)$ and sends $\mathsf{rk}_{\mathsf{id}_i \to \mathsf{id}_j}$ to $\mathcal{A}_4$.

- Challenge query $(M_0, M_1, [\mathsf{id}_0^*, \ldots, \mathsf{id}_r^*], \mathsf{id}^{**})$: If $\mathcal{A}_4$ can trivially decrypt ciphertexts encrypted under id^{**} then $\mathcal{B}_2$ sends $\perp$. Otherwise, $\mathcal{B}_2$ generates keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_2$ then constructs $C_0^{pk}, \ldots, C_{r+1}^{pk}$ as well as $C_{u,u}^{bks}, \ldots, C_{u,r+1}^{bks}$ for $u \in \{1, \ldots, r\}$ following the procedure in Lemma 15. $\mathcal{B}_2$ makes AONT challenge query $\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_0), \mathsf{E}^{\mathsf{Sym}}(k_0, M_1)\big)$, receiving oracle access to any $N-\ell$ bits of AONT output $C_0^T = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b\big)$ in response. $\mathcal{B}_2$ responds to $\mathcal{A}_4$ by giving it oracle access to
$$\{C_u^* = \big(C_u^{pk}, \big[C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}\big], C_u^T\big)\}_{0 \leq u \leq r}$$
for
$$C_u^T = \Big[\cdots \Big[\big[C_0^T\big]_{\mathsf{Ind}(s_1,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\Big] \cdots\Big]_{\mathsf{Ind}(s_u,\ell^*),\mathsf{Ctr}(k_u,\ell^*)},$$
with a limit of $(1-\varepsilon)|C_r^*|$ total bits queried. Whenever $\mathcal{A}_4$ queries a bit of C_u^T, $\mathcal{B}_2$ queries the corresponding bit of C_0^T. For each $u' \in \{1, \ldots, u\}$, if the bit is in $\mathsf{Ind}(s_{u'}, \ell^*)$, $\mathcal{B}_2$ XORs it with the corresponding bit of $\mathsf{Ctr}(k_{u'})$. $\mathcal{B}_2$ returns the resulting bit to $\mathcal{A}_4$. When $\mathcal{A}_4$ finsihes its queries, $\mathcal{B}_2$ chooses random $L \in \left\{ {N \atop \ell^*} \right\}$ and queries any bits of $\big[C_0^T\big]_L$ that it hadn't previously queried. If this requires more than $N - \ell$ bits of C_0^T then $\mathcal{B}_2$ aborts the simulation. Otherwise, it produces $\widetilde{C_r^T} \in \{0,1\}^N$ by setting the bits not in L as their corresponding values from C_r^T and choosing the bits in L randomly. $\mathcal{B}_2$ then gives $\mathcal{A}_4$
$$C^{**} = \Big(C_{r+1}^{pk}, \big[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, \mathsf{E}(\mathsf{id}^{**}, (0,0))\big], \widetilde{C_r^T}\Big)$$
- Guess b': $\mathcal{B}_2$ receives b' as $\mathcal{A}_4$'s guess and uses the same bit b' for its guess as well.

We now look at the probability that $\mathcal{B}_2$ aborts when responding to $\mathcal{A}_4$'s challenge query. Let ε' be the fraction of C^T that $\mathcal{A}_4$ does not query from C^*; note that $\varepsilon' \geq \varepsilon - \delta$. Let ℓ' be the number of bits of C^T that are in L but $\mathcal{A}_4$ does not query from its oracle. $\mathcal{B}_2$ aborts if and only if $\ell' < \ell$. To get any upper bound on ℓ', we apply Lemma 5. We let balls in the lemma correspond to bits here. Balls selected in the first stage correspond to bits of C^T that were not queried, and balls selected in the second stage correspond to bits in L. Then ℓ' here is equivalent to ℓ' in the lemma. Thus by Lemma 5, for any $t > 0$, $\Pr[\ell' \leq \ell^*(\varepsilon' - t)] \leq e^{-2\ell^* t^2}$.

For any fixed t, this probability is negligible for $\ell^* = \omega(\log(\lambda))$. Thus as long as $\ell < \ell^* \cdot \varepsilon'$, the probability that $\mathcal{B}_2$ aborts is negligible. Since $\varepsilon' \geq \varepsilon - \delta$, if $\ell^* > \frac{\ell}{\varepsilon - \delta}$ then this will hold.

If $\mathcal{B}_2$ does not abort then its probability of success is identical to $\mathcal{A}_4$'s probability of success. Since L was chosen randomly and any string XORed with a random string is a random string, $\widetilde{C_r^T} = \big[C_r^T\big]_{\mathsf{rInd}(\ell^*),\mathsf{rStr}(\ell^*)}$.

Thus $\mathcal{A}_4's$ view here is the same as in Game_3, so its probability of success is p_3. Hence $\mathcal{B}_2$'s probability of success, when accounting for the probability that it aborts, is $p_3 - \mathsf{negl}'$ for some negligible function negl'. By the ℓ-AONT security of T, $p_3 - \mathsf{negl}' < \frac{1}{2} + \mathsf{negl}''$ for some negligible function negl''. Therefore $p_3 < \frac{1}{2} + \mathsf{negl}$ for $\mathsf{negl} = \mathsf{negl}' + \mathsf{negl}''$. □

Now we use these lemmas to prove Theorem 13.

Proof. By Lemmas 14, 15, and 16, we see that the adversary's probability of success in Game_0 can only be negligibly greater than its probability of success in Game_3. By Lemma 17, the adversary's success in Game_3 can be at most $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl. Thus the adversary's success in Game_0, which is the real case, can be at most $\frac{1}{2} + \mathsf{negl}'$ for some negligible function negl'. Combined with Theorem 12, this proves that is $(1 - \varepsilon)$-$\mathsf{Adap\text{-}Revoke\text{-}IB\text{-}PRE\text{-}CPA}$-secure. □

E.3 Static-Revocable-IB-PRE-CPA Security

Theorem 14. *Assume the existence of a* $\mathsf{IB\text{-}PRE\text{-}CPA}$*-secure identity-based proxy re-encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{D}, \mathsf{RG}, \mathsf{RE})$ *with re-encryption history independence, a symmetric-key encryption scheme* $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$*, and an* ℓ*-AONT* T*. Suppose that for the construction from Section 11.4,* C^T *comprises at least a fraction* $1-\delta$ *of the total size of each ciphertext. Then for any* $\varepsilon < 1$ *with* $\varepsilon > \delta$ *and any* $\ell^* > \frac{\ell}{\varepsilon - \delta}$*, this construction is* $(1-\varepsilon)$-$\mathsf{Stat\text{-}Revoke\text{-}IB\text{-}PRE\text{-}CPA}$-*secure.*

Proof. This proof is identical to the proof of Theorem 13, with the exception of how $\mathcal{B}_2$ responds to the challenge queries. In the proofs of Lemmas 14, 15, and 16, the adversary requests the bits of $\{C_u^*\}_{0 \le u \le r}$ to receive all at once instead of adaptively. In the proof of Lemmma 17 for the static case, it works as follows: $\mathcal{B}_2$ receives challenge query $(M_0, M_1, [\mathsf{id}_0^*, \ldots, \mathsf{id}_r^*], \mathsf{id}^{**})$. Let bitPos_T be the set of bit positions requested that correspond to bits of C^T. $\mathcal{B}_2$ chooses random $L \in \left\{ {N \atop \ell^*} \right\}$. If $|L \cap ([N] \setminus \mathsf{bitPos}_T)| < \ell$ then B_2 aborts the simulation. Otherwise, $\mathcal{B}_2$ generates random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$ and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_2$ then constructs $C_0^{pk}, \ldots, C_{r+1}^{pk}$ as well as $C_{u,u}^{bks}, \ldots, C_{u,r+1}^{bks}$ for $u \in \{1, \ldots, r\}$, following the procedure in Lemma 15. $\mathcal{B}_2$ makes AONT challenge query

$$\left(\mathsf{E}^{\mathsf{Sym}}(k_0, M_0), \mathsf{E}^{\mathsf{Sym}}(k_0, M_1), L \cap ([N] \setminus \mathsf{bitPos}_T)\right) \ ,$$

receiving

$$\left[C^T\right]_{L \cap ([N] \setminus \mathsf{bitPos}_T)} = \left[T\left(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\right)\right]_{L \cap ([N] \setminus \mathsf{bitPos}_T)}$$

in response. $\mathcal{B}_2$ produces $C_0^T \in \{0,1\}^N$ by taking $\left[C^T\right]_{L \cap ([N] \setminus \mathsf{bitPos}_T)}$ and filling in the bits from $L \cap ([N] \setminus \mathsf{bitPos}_T)$ arbitrarily (these bits will not affect the challenge ciphertexts), and then it computes

$$\{C_u^* = \left(C_u^{pk}, \left[C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}\right], C_u^T\right)\}_{0 \le u \le r}$$

for

$$C_u^T = \left[\cdots \left[\left[C_0^T\right]_{\mathsf{Ind}(s_1, \ell^*), \mathsf{Ctr}(k_1, \ell^*)}\right] \cdots\right]_{\mathsf{Ind}(s_u, \ell^*), \mathsf{Ctr}(k_u, \ell^*)} \ .$$

$\mathcal{B}_2$ constructs $\widetilde{C_r^T} \in \{0,1\}^N$ by setting the bits not in L as their corresponding values from C_r^T and choosing the remaining bits randomly. It gives $\mathcal{A}_4$ the requested $(1-\varepsilon)|C_r^*|$ bits of $\{C_u^*\}_{0\le u\le r}$ as well as

$$C^{**} = \left(C_{r+1}^{pk}, \left[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, \mathsf{E}(\mathsf{id}^{**}, (0,0))\right], \widetilde{C_r^T}\right) \ .$$

When considering the probability that $\mathcal{B}_2$ aborts when responding to $\mathcal{A}_4$'s challenge query, here we let $\ell' = |L \cap ([N] \setminus \mathsf{bitPos}_T)|$, which is the number of bit positions of C^T that are in L but not in bitPos_T. Again, $\mathcal{B}_2$ aborts if and only if $\ell' < \ell$. The upper bound on ℓ' is still the same as for the adaptive case by the exact same argument. Thus the probability that $\mathcal{B}_2$ aborts is again negligible as long as $\ell^* > \frac{\ell}{\varepsilon-\delta}$. □

F Proofs for Revocable-Storage Attribute-Based Encryption

F.1 RS-KP-ABE-Security of Hybrid Construction

Theorem 15. *Assume the existence of a* RS-KP-ABE*-secure revocable-storage attribute-based encryption* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$*, a CPA-secure symmetric-key encryption scheme* $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$*, and an AONT* T*. Then the construction of* Π_{hyb} *in Section 12.4 is* RS-KP-ABE*-secure.*

We show that by the security of the proxy re-encryption scheme, the real game Game_{real} is computationally indistinguishable from a hybrid game Game_{hyb} where the challenge ciphertext is

$$C^* = \left(\mathsf{E}(S, 0, t), [\,], T\left(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\right)\right) \ .$$

Then we show that by the security of the symmetric-key encryption scheme, in Game_{hyb} the adversary's success rate is less than $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl.

Lemma 18. *Suppose that an adversary* $\mathcal{A}_1$ *for the* RS-KP-ABE *security game has probability of success* p_0 *in* Game_{real} *and probability of success* p_1 *in* Game_{hyb}*. If the underlying proxy re-encryption scheme* Π *is* RS-KP-ABE*-secure, then* $p_1 - p_0$ *is negligible.*

Proof. Given an adversary $\mathcal{A}_1$ as described for the Π_{hyb} construction, we show how to construct an adversary $\mathcal{B}_1$ for the original scheme Π that plays the RS-KP-ABE security game. As is typical, $\mathcal{B}_1$ simulates $\mathcal{A}_1$ and its version of the RS-KP-ABE security game. First, $\mathcal{B}_1$ gives its simulation of $\mathcal{A}_1$ the public parameters it receives. $\mathcal{B}_1$ responds to queries from $\mathcal{A}_1$ as follows:

- Secret Key Generation query $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$: $\mathcal{B}_1$ queries (P, id) to its secret key generation oracle and receives $\mathsf{sk}_{P,\mathsf{id}}$ in response, which is sent to $\mathcal{A}_1$.

- Key Update Generation query $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$: $\mathcal{B}_1$ queries (t, rl) to its key update generation oracle and receives ku_t in response, which is sent to $\mathcal{A}_1$.
- Challenge query (M_0, M_1, S^*, t^*): $\mathcal{B}_1$ generates a random bit b and key $k_0 \leftarrow \mathsf{G}^{\mathsf{Sym}}$. $\mathcal{B}_1$ then makes its own challenge query $(k_0, 0, S^*, t^*)$, receiving ciphertext C' in response. $\mathcal{B}_1$ gives $\mathcal{A}_1$ the following challenge ciphertext:

$$C^* = \big(C', [\,], T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)\big) \ .$$

- Guess b': If $b = b'$ then $\mathcal{B}_1$ guesses that C' is an encryption of k_0, otherwise $\mathcal{B}_1$ guesses that C' is an encryption of 0.

Observe that if C' is an encryption of k_0, then $\mathcal{A}_1$ is executing in a perfect simulation of Game_{real}, and thus has probability of success p_0; hence $\mathcal{B}_1$ is correct with probability p_0. If C' is an encryption of 0 then $\mathcal{A}_1$ is simulated in the Game_{hyb} and has probability of success p_1; thus $\mathcal{B}_1$ is correct with probability $1 - p_1$. Therefore $\mathcal{B}_1$'s probability of success is $\frac{1}{2}(p_0 + 1 - p_1) = \frac{1}{2} + \frac{1}{2}(p_0 - p_1)$. By the $\mathsf{RS\text{-}KP\text{-}ABE}$ security of $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$, $\frac{1}{2}(p_0 - p_1)$ is negligible, and so $p_0 - p_1$ is negligible. □

Lemma 19. *Suppose that an adversary $\mathcal{A}_2$ for the* $\mathsf{RS\text{-}KP\text{-}ABE}$ *security game has probability of success p_1 in* Game_{hyb}. *If the underlying symmetric-key encryption scheme Π_{sym} is CPA-secure, then $p_1 < \frac{1}{2} + \mathsf{negl}$ for some negligible function* negl.

Proof. We construct an adversary $\mathcal{B}_2$ for Π_{sym} that plays the CPA security game by simulating adversary $\mathcal{A}_2$ in the $\mathsf{RS\text{-}KP\text{-}ABE}$ security game. $\mathcal{B}_2$ instantiates the $\mathsf{RS\text{-}KP\text{-}ABE}$ scheme itself by computing $(\mathsf{param}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends param to $\mathcal{A}_2$. $\mathcal{B}_2$ responds to queries from the simulation of $\mathcal{A}_2$ as follows:

- Secret Key Generation query $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$: If responding to the query would allow $\mathcal{A}_2$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{sk}_{P,\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, P, \mathsf{id})$ and sends $\mathsf{sk}_{P,\mathsf{id}}$ to $\mathcal{A}_2$.
- Key Update Generation query $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$: If responding to the query would allow $\mathcal{A}_2$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{ku}_{t,\mathsf{rl}} \leftarrow \mathsf{RG}(\mathsf{msk}, t, \mathsf{rl})$ and sends $\mathsf{ku}_{t,\mathsf{rl}}$ to $\mathcal{A}_2$.
- Challenge query (M_0, M_1, S^*, t^*): If $\mathcal{A}_2$ can trivially decrypt ciphertexts encrypted under $\S^*$ at time $t*$ then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ makes challenge query (M_0, M_1), receiving ciphertext C' in response. $\mathcal{B}_2$ gives $\mathcal{A}_2$ challenge ciphertext $C^* = (\mathsf{E}(S^*, 0, t^*), [\,], T(C'))$.
- Guess b': $\mathcal{B}_2$ receives b' as $\mathcal{A}_2$'s guess and uses the same bit b' for its guess as well.

Since $\mathcal{A}_2$ is in a perfect simulation of Game_{hyb}, its probability of success is p_1. $\mathcal{B}_2$ is correct when $\mathcal{A}_2$ is correct, so its probability of success is also p_1. By the CPA security of $\Pi_{sym} = \big(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\big)$, $p_1 < \frac{1}{2} + \mathsf{negl}$ for some negligible function negl. □

Now we use these lemmas to prove Theorem 15.

Proof. By Lemmas 18, we see that the adversary's probability of success in Game_{real} can only be negligibly greater than its probability of success in Game_{hyb}. By Lemma 19, the adversary's success in Game_{hyb} can be at most $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl. Thus the adversary's success in Game_{real} can be at most $\frac{1}{2} + \mathsf{negl}'$ for some negligible function negl'. □

F.2 Adaptive-Revocable-RS-KP-ABE Security

Theorem 16. *Assume the existence of a* RS-KP-ABE*-secure revocable-storage attribute-based encryption scheme $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$, a symmetric-key encryption scheme $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$, and an adaptive ℓ-AONT T. Suppose that for the construction from Section 12.4, C^T comprises at least a fraction $1-\delta$ of the total size of each ciphertext. Then for any $\varepsilon < 1$ with $\varepsilon > \delta$ and any $\ell^* > \frac{\ell}{\varepsilon - \delta}$, this construction is $(1-\varepsilon)$-*Adap-Revoke-RS-KP-ABE*-secure.*

We show the computational indistinguishability of a series of games. Each game is the same as the real game except in regards to C_{S^*,t^*+r+1}. In each game, the challenge query is $(M_0, M_1, S^*, t^* + r)$. We highlight in bold the portions of C_{S^*,t^*+r+1} that differ from the previous games, in the descriptions of the respective games that follow:

Game_0*:* This is the real game, where:

$$C_{S^*,t^*+r+1} = \Big(\mathsf{CTU}\big(C^{pk}\big), \big[C_1^{bks}, \ldots, C_r^{bks}, \mathsf{E}(S^*, (s, k_1), t^* + r + 1)\big], \big[C^T\big]_{\mathsf{Ind}(s,\ell^*), \mathsf{Ctr}(k_1,\ell^*)}\Big) \ .$$

Game_1*:* This is identical to Game_0 except that we replace $\mathsf{E}(S^*, (s, k_1), t^* + r + 1)$ with $\mathsf{E}(S^*, (0, 0), t^* + r + 1)$, resulting in:

$$C_{S^*,t^*+r+1} = \Big(\mathsf{CTU}\big(C^{pk}\big), \big[C_1^{bks}, \ldots, C_r^{bks}, \boldsymbol{\mathsf{E}(S^*, (0, 0), t^* + r + 1)}\big], \big[C^T\big]_{\mathsf{Ind}(s,\ell^*), \mathsf{Ctr}(k_1,\ell^*)}\Big) \ .$$

Game_2*:* This is identical to Game_1 except that we replace the pseudorandom $\mathsf{Ind}(s, \ell^*)$ with truly random $\mathsf{rInd}(\ell^*)$, resulting in:

$$C_{S^*,t^*+r+1} = \Big(\mathsf{CTU}\big(C^{pk}\big), \big[C_1^{bks}, \ldots, C_r^{bks}, \mathsf{E}(S^*, (0, 0), t^* + r + 1)\big], \big[C^T\big]_{\boldsymbol{\mathsf{rInd}(\ell^*)}, \mathsf{Ctr}(k_1,\ell^*)}\Big) \ .$$

Game_3*:* This is identical to Game_2 except that we replace the keystream $\mathsf{Ctr}(k_1, \ell^*)$ we get from counter mode encryption with a random string $\mathsf{rStr}(\ell^*)$, resulting in:

$$C_{S^*,t^*+r+1} = \Big(\mathsf{CTU}\big(C^{pk}\big), \big[C_1^{bks}, \ldots, C_r^{bks}, \mathsf{E}(S^*, (0, 0), t^* + r + 1)\big], \big[C^T\big]_{\mathsf{rInd}(\ell^*), \boldsymbol{\mathsf{rStr}(\ell^*)}}\Big) \ .$$

We now provide a series of lemmas that show that any adverary's probabilities of success in two successive games are negligibly close. These are presented in Lemmas 20, 21, and 22. Finally, we show in Lemma 23 that any adversary's chance of success in the final game is negligibly close to 1/2.

Lemma 20. *Suppose that an adversary $\mathcal{A}_1$ for the* $(1-\varepsilon)$-Adap-Revoke-RS-KP-ABE *security game has probability of success p_0 in* Game_0 *and probability of success p_1 in* Game_1*. If the underlying proxy re-encryption scheme Π is* RS-KP-ABE*-secure, then $p_1 - p_0$ is negligible.*

Proof. We construct an adversary $\mathcal{B}_1$ that plays the RS-KP-ABE security game by simulating adversary $\mathcal{A}_1$. $\mathcal{B}_1$ gives $\mathcal{A}_1$ the public parameters it receives. $\mathcal{B}_1$ responds to queries from $\mathcal{A}_1$ as follows:

- Secret Key Generation query $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$: $\mathcal{B}_1$ queries (P, id) to its secret key generation oracle and receives $\mathsf{sk}_{P,\mathsf{id}}$ in response, which is sent to $\mathcal{A}_1$.
- Key Update Generation query $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$: $\mathcal{B}_1$ queries (t, rl) to its key update generation oracle and receives ku_t in response, which is sent to $\mathcal{A}_1$.
- Challenge query (M_0, M_1, S^*, t^*, r): $\mathcal{B}_1$ generates random bit b, random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_1$ creates $C_0^{pk} = \mathsf{E}(S^*, k_0, t^*)$, and then computes

$$C_1^{pk} = \mathsf{CTU}\Big(C_0^{pk}\Big), C_2^{pk} = \mathsf{CTU}\Big(C_1^{pk}\Big), \\ \ldots, C_r^{pk} = \mathsf{CTU}\Big(C_{r-1}^{pk}\Big), C_{r+1}^{pk} = \mathsf{CTU}\big(C_r^{pk}\big)$$

For each $u \in \{1, \ldots, r\}$, $\mathcal{B}_1$ creates $C_{u,u}^{bks} = \mathsf{E}(S^*, (s_u, k_u), t^* + u)$ and computes

$$C_{u,u+1}^{bks} = \mathsf{CTU}\big(C_{u,u}^{bks}\big), C_{u,u+2}^{bks} = \mathsf{CTU}\big(C_{u,u+1}^{bks}\big), \\ \ldots, C_{u,r}^{bks} = \mathsf{CTU}\big(C_{u,r-1}^{bks}\big), C_{u,r+1}^{bks} = \mathsf{CTU}\big(C_{u,r}^{bks}\big)$$

Then $\mathcal{B}_1$ creates $C_0^T = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)$ and $C_{S^*,t^*} = \Big(C_0^{pk}, [\,], C_0^T\Big)$. For each $u \in \{1, \ldots, r\}$, $\mathcal{B}_1$ creates

$$C_u^T = \big[C_{u-1}^T\big]_{\mathsf{Ind}(s_u, \ell^*), \mathsf{Ctr}(k_u, \ell^*)}$$

and

$$C_{S^*,t^*+u} = \big(C_u^{pk}, \big[C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}\big], C_u^T\big) \ .$$

$\mathcal{B}_1$ makes challenge query $((s_{r+1}, k_{r+1}), (0, 0), S^*, t^*)$, receiving ciphertext C' in response. It gives $\mathcal{A}_1$ the $(1-\varepsilon)|C_{S^*,t^*+r}|$ bits that it adaptively requests of $\{C_{S^*,t^*+u}\}_{0 \le u \le r}$ as well as

$$C_{S^*,t^*+r+1} = \Big(C_{r+1}^{pk}, \big[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, C'\big], \big[C_r^T\big]_{\mathsf{Ind}(s_{r+1}, \ell^*), \mathsf{Ctr}(k_{r+1}, \ell^*)}\Big) \ .$$

- Guess b': If $b = b'$ then $\mathcal{B}_1$ guesses that C' is an encryption of (s_{r+1}, k_{r+1}), otherwise $\mathcal{B}_1$ guesses that C' is an encryption of $(0, 0)$.

If C' is an encryption of (s_{r+1}, k_{r+1}), then $\mathcal{A}_1$ is in Game_0 (the real game) and has probability of success p_0; thus $\mathcal{B}_1$ is correct with probability p_0. If C'

is an encryption of $(0,0)$ then $\mathcal{A}_1$ is in Game_1 and has probability of success p_1; thus $\mathcal{B}_1$ is correct with probability $1 - p_1$. Therefore $\mathcal{B}_1$'s probability of success is $\frac{1}{2}(p_0 + 1 - p_1) = \frac{1}{2} + \frac{1}{2}(p_0 - p_1)$. By the RS-KP-ABE security of $(\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$, $\frac{1}{2}(p_0 - p_1)$ is negligible, and so $p_0 - p_1$ is negligible.

□

Lemma 21. *Suppose that an adversary $\mathcal{A}_2$ for the* $(1-\varepsilon)$-Adap-Revoke-RS-KP-ABE *security game has probability of success p_1 in* Game_1 *and probability of success p_2 in* Game_2*. If* $\mathsf{Ind}(s, \ell^*)$ *with random seed s is pseudorandom (indistinguishable from* $\mathsf{rInd}(\ell^*)$*), then $p_2 - p_1$ is negligible.*

Proof. We construct a distinguisher $\mathcal{D}_1$ that receives a set of indices ind—either $\mathsf{ind} = \mathsf{Ind}(s, \ell^*)$ with random seed s or $\mathsf{ind} = \mathsf{rInd}(\ell^*)$. $\mathcal{D}_1$ simulates adversary $\mathcal{A}_2$. $\mathcal{D}_1$ instantiates the RS-KP-ABE scheme itself and sends $\mathcal{A}_2$ the public parameters. $\mathcal{D}_1$ responds to queries from $\mathcal{A}_2$ as follows:

- Secret Key Generation query $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$: If responding to the query would allow $\mathcal{A}_2$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{D}_1$ sends $\perp$; otherwise, $\mathcal{D}_1$ computes $\mathsf{sk}_{P,\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, P, \mathsf{id})$ and sends $\mathsf{sk}_{P,\mathsf{id}}$ to $\mathcal{A}_2$.
- Key Update Generation query $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$: If responding to the query would allow $\mathcal{A}_2$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{D}_1$ sends $\perp$; otherwise, $\mathcal{D}_1$ computes $\mathsf{ku}_{t,\mathsf{rl}} \leftarrow \mathsf{RG}(\mathsf{msk}, t, \mathsf{rl})$ and sends $\mathsf{ku}_{t,\mathsf{rl}}$ to $\mathcal{A}_2$.
- Challenge query (M_0, M_1, S^*, t^*, r): If $\mathcal{A}_2$ can trivially decrypt ciphertexts encrypted under $\S^*$ at time $t* + r$ then $\mathcal{D}_1$ sends $\perp$. Otherwise, $\mathcal{D}_1$ generates random bit b, random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{D}_1$ creates $C_0^{pk} = \mathsf{E}(S^*, k_0, t^*)$, and then computes

$$C_1^{pk} = \mathsf{CTU}\Big(C_0^{pk}\Big), C_2^{pk} = \mathsf{CTU}\Big(C_1^{pk}\Big), \ldots, C_r^{pk} = \mathsf{CTU}\Big(C_{r-1}^{pk}\Big), C_{r+1}^{pk} = \mathsf{CTU}\big(C_r^{pk}\big)$$

  For each $u \in \{1, \ldots, r\}$, $\mathcal{D}_1$ creates $C_{u,u}^{bks} = \mathsf{E}(S^*, (s_u, k_u), t^* + u)$ and computes

$$C_{u,u+1}^{bks} = \mathsf{CTU}\big(C_{u,u}^{bks}\big), C_{u,u+2}^{bks} = \mathsf{CTU}\big(C_{u,u+1}^{bks}\big), \ldots, C_{u,r}^{bks} = \mathsf{CTU}\big(C_{u,r-1}^{bks}\big), C_{u,r+1}^{bks} = \mathsf{CTU}\big(C_{u,r}^{bks}\big)$$

  Then $\mathcal{D}_1$ creates $C_0^T = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\big)$ and $C_0^* = \Big(C_0^{pk}, [\,], C_0^T\Big)$. For each $u \in \{1, \ldots, r\}$, $\mathcal{D}_1$ creates

$$C_u^T = \big[C_{u-1}^T\big]_{\mathsf{Ind}(s_u, \ell^*), \mathsf{Ctr}(k_u, \ell^*)}$$

  and

$$C_{S^*, t^* + u} = \big(C_u^{pk}, \big[C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}\big], C_u^T\big) \ .$$

$\mathcal{D}_1$ gives $\mathcal{A}_2$ the $(1-\varepsilon)|C_{S^*,t^*+r}|$ bits that it adaptively requests of $\{C_{S^*,t^*+u}\}_{0\le u\le r}$ as well as

$$C_{S^*,t^*+r+1} = \left(C^{pk}_{r+1}, \left[C^{bks}_{1,r+1}, \ldots, C^{bks}_{r,r+1}, \mathsf{E}(S^*, (0,0), t^* + r + 1)\right], \left[C^{T}_{r}\right]_{\mathsf{ind},\mathsf{Ctr}(k_{r+1},\ell^*)}\right) .$$

- Guess b': If $b = b'$ then $\mathcal{D}_1$ guesses that $\mathsf{ind} = \mathsf{Ind}(s, \ell^*)$, otherwise $\mathcal{D}_1$ guesses that $\mathsf{ind} = \mathsf{rInd}(\ell^*)$.

If $\mathsf{ind} = \mathsf{Ind}(s, \ell^*)$, then $\mathcal{A}_2$ is in Game_1 and has probability of success p_1; thus $\mathcal{D}_1$ is correct with probability p_1. If $\mathsf{ind} = \mathsf{rInd}(\ell^*)$, then $\mathcal{A}_2$ is in Game_2 and has probability of success p_2; thus $\mathcal{D}_1$ is correct with probability $1 - p_2$. Therefore $\mathcal{D}_1$'s probability of success is $\frac{1}{2}(p_1 + 1 - p_2) = \frac{1}{2} + \frac{1}{2}(p_1 - p_2)$. By the pseudorandomness of $\mathsf{Ind}(s, \ell^*)$, $\frac{1}{2}(p_1 - p_2)$ is negligible, and so $p_1 - p_2$ is negligible. □

Lemma 22. *Suppose that an adversary $\mathcal{A}_3$ for the $(1-\varepsilon)$-Adap-Revoke-RS-KP-ABE security game has probability of success p_2 in Game_2 and probability of success p_3 in Game_3. If $\mathsf{Ctr}(k_1, \ell^*)$ with random key k_1 is pseudorandom (indistinguishable from $\mathsf{rStr}(\ell^*)$), then $p_3 - p_2$ is negligible.*

Proof. We construct a distinguisher $\mathcal{D}_2$ that receives a bitstream str—either $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$ with random key k_1 or $\mathsf{str} = \mathsf{rStr}(\ell^*)$. $\mathcal{D}_2$ simulates adversary $\mathcal{A}_3$. $\mathcal{D}_2$ instantiates the RS-KP-ABE scheme itself and sends $\mathcal{A}_3$ the public parameters. $\mathcal{D}_2$ responds to queries from $\mathcal{A}_3$ as follows:

- Secret Key Generation query $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$: If responding to the query would allow $\mathcal{A}_3$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{D}_2$ sends $\perp$; otherwise, $\mathcal{D}_2$ computes $\mathsf{sk}_{P,\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, P, \mathsf{id})$ and sends $\mathsf{sk}_{P,\mathsf{id}}$ to $\mathcal{A}_3$.
- Key Update Generation query $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$: If responding to the query would allow $\mathcal{A}_3$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{D}_2$ sends $\perp$; otherwise, $\mathcal{D}_2$ computes $\mathsf{ku}_{t,\mathsf{rl}} \leftarrow \mathsf{RG}(\mathsf{msk}, t, \mathsf{rl})$ and sends $\mathsf{ku}_{t,\mathsf{rl}}$ to $\mathcal{A}_3$.
- Challenge query (M_0, M_1, S^*, t^*, r): If $\mathcal{A}_3$ can trivially decrypt ciphertexts encrypted under $\S^*$ at time $t*{+}r$ then $\mathcal{D}_2$ sends $\perp$. Otherwise, $\mathcal{D}_2$ generates random bit b, random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{D}_2$ then constructs $C^*_0, \ldots, C^*_r, C^{pk}_{r+1}, \left[C^{bks}_{1,r+1}, \ldots, C^{bks}_{r,r+1}\right]$, and C^T_r following the procedure in Lemma 21. $\mathcal{D}_2$ gives $\mathcal{A}_3$ the $(1-\varepsilon)|C_{S^*,t^*+r}|$ bits that it adaptively requests of $\{C_{S^*,t^*+u}\}_{0\le u\le r}$ as well as

$$C_{S^*,t^*+r+1} = \left(C^{pk}_{r+1}, \left[C^{bks}_{1,r+1}, \ldots, C^{bks}_{r,r+1}, \mathsf{E}(S^*, (0,0), t^* + r + 1)\right], \left[C^{T}_{r}\right]_{\mathsf{ind},\mathsf{Ctr}(k_{r+1},\ell^*)}\right) .$$

- Guess b': If $b = b'$ then $\mathcal{D}_2$ guesses that $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$, otherwise $\mathcal{D}_2$ guesses that $\mathsf{str} = \mathsf{rStr}(\ell^*)$.

If $\mathsf{str} = \mathsf{Ctr}(k_1, \ell^*)$, then $\mathcal{A}_2$ is in Game_2 and has probability of success p_2; thus $\mathcal{D}_2$ is correct with probability p_2. If $\mathsf{str} = \mathsf{rStr}(\ell^*)$, then $\mathcal{A}_2$ is in Game_3

and has probability of success p_3; thus $\mathcal{D}_2$ is correct with probability $1 - p_3$. Therefore $\mathcal{D}_2$'s probability of success is $\frac{1}{2}(p_2 + 1 - p_3) = \frac{1}{2} + \frac{1}{2}(p_2 - p_3)$. By the pseudorandomness of $\mathsf{Ctr}(k_1, \ell^*)$, $\frac{1}{2}(p_2 - p_3)$ is negligible, and so $p_2 - p_3$ is negligible. □

Lemma 23. *Suppose that an adversary $\mathcal{A}_4$ for the* $(1-\varepsilon)$-Adap-Revoke-RS-KP-ABE *security game has probability of success p_3 in* Game_3*. Suppose also that the underlying AONT T is a computationally-secure adaptive ℓ-AONT and C^T comprises at least a fraction $1-\delta$ of the total size of each ciphertext. If $\varepsilon > \delta$ and $\ell^* > \frac{\ell}{\varepsilon-\delta}$, then $p_3 < \frac{1}{2} + \mathsf{negl}$ for some negligible function* negl*.*

Proof. We construct an adversary $\mathcal{B}_2$ that plays the ℓ-AONT security game by simulating adversary $\mathcal{A}_4$. $\mathcal{B}_2$ instantiates the RS-KP-ABE scheme itself and sends $\mathcal{A}_4$ the public parameters. $\mathcal{B}_2$ responds to queries from $\mathcal{A}_4$ as follows:

– Secret Key Generation query $\mathcal{O}_{\mathsf{sk}}(P, \mathsf{id})$: If responding to the query would allow $\mathcal{A}_4$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{sk}_{P,\mathsf{id}} \leftarrow \mathsf{G}(\mathsf{msk}, P, \mathsf{id})$ and sends $\mathsf{sk}_{P,\mathsf{id}}$ to $\mathcal{A}_4$.

– Key Update Generation query $\mathcal{O}_{\mathsf{ku}}(t, \mathsf{rl})$: If responding to the query would allow $\mathcal{A}_4$ to trivially decrypt the challenge ciphertext (in Phase 2), then $\mathcal{B}_2$ sends $\perp$; otherwise, $\mathcal{B}_2$ computes $\mathsf{ku}_{t,\mathsf{rl}} \leftarrow \mathsf{RG}(\mathsf{msk}, t, \mathsf{rl})$ and sends $\mathsf{ku}_{t,\mathsf{rl}}$ to $\mathcal{A}_4$.

– Challenge query (M_0, M_1, S^*, t^*, r): If $\mathcal{A}_4$ can trivially decrypt ciphertexts encrypted under $\S^*$ at time $t * +r$ then $\mathcal{B}_2$ sends $\perp$. Otherwise, $\mathcal{B}_2$ generates keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$, and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_2$ then constructs $C_0^{pk}, \ldots, C_{r+1}^{pk}$ as well as $C_{u,u}^{bks}, \ldots, C_{u,r+1}^{bks}$ for $u \in \{1, \ldots, r\}$ following the procedure in Lemma 21. $\mathcal{B}_2$ makes AONT challenge query $\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_0), \mathsf{E}^{\mathsf{Sym}}(k_0, M_1)\big)$, receiving oracle access to AONT output $C_0^T = T\big(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)$ in response. $\mathcal{B}_2$ responds to $\mathcal{A}_4$ by giving it oracle access to

$$\{C_{S^*,t^*+u} = \big(C_u^{pk}, [C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}], C_u^T\big)\}_{0 \le u \le r}$$

for

$$C_u^T = \Big[\cdots \Big[\big[C_0^T\big]_{\mathsf{Ind}(s_1,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\Big] \cdots\Big]_{\mathsf{Ind}(s_u,\ell^*),\mathsf{Ctr}(k_u,\ell^*)} ,$$

with a limit of $(1-\varepsilon)|C_r^*|$ total bits queried. Whenever $\mathcal{A}_4$ queries a bit of C_u^T, $\mathcal{B}_2$ queries the corresponding bit of C_0^T. For each $u' \in \{1, \ldots, u\}$, if the bit is in $\mathsf{Ind}(s_{u'}, \ell^*)$, $\mathcal{B}_2$ XORs it with the corresponding bit of $\mathsf{Ctr}(k_{u'})$. $\mathcal{B}_2$ returns the resulting bit to $\mathcal{A}_4$. When $\mathcal{A}_4$ finsihes its queries, $\mathcal{B}_2$ chooses random $L \in \left\{ {N \atop \ell^*} \right\}$ and queries any bits of $\big[C_0^T\big]_L$ that it hadn't previously queried. If this requires more than $N - \ell$ bits of C_0^T then $\mathcal{B}_2$ aborts the simulation. Otherwise, it produces $\widetilde{C_r^T} \in \{0,1\}^N$ by setting the bits not in L as their corresponding values from C_r^T and choosing the bits in L randomly. $\mathcal{B}_2$ then gives $\mathcal{A}_4$

$$C_{S^*,t^*+r+1} = \Big(C_{r+1}^{pk}, \big[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, \mathsf{E}(S^*, (0,0), t^* + r + 1)\big], \widetilde{C_r^T}\Big)$$

- Guess b': $\mathcal{B}_2$ receives b' as $\mathcal{A}_4$'s guess and uses the same bit b' for its guess as well.

We now look at the probability that $\mathcal{B}_2$ aborts when responding to $\mathcal{A}_4$'s challenge query. Let ε' be the fraction of C^T that $\mathcal{A}_4$ does not query from C^*; note that $\varepsilon' \geq \varepsilon - \delta$. Let ℓ' be the number of bits of C^T that are in L but $\mathcal{A}_4$ does not query from its oracle. $\mathcal{B}_2$ aborts if and only if $\ell' < \ell$. To get any upper bound on ℓ', we apply Lemma 5. We let balls in the lemma correspond to bits here. Balls selected in the first stage correspond to bits of C^T that were not queried, and balls selected in the second stage correspond to bits in L. Then ℓ' here is equivalent to ℓ' in the lemma. Thus by Lemma 5, for any $t > 0$, $\Pr[\ell' \leq \ell^*(\varepsilon' - t)] \leq e^{-2\ell^* t^2}$.

For any fixed t, this probability is negligible for $\ell^* = \omega(\log(\lambda))$. Thus as long as $\ell < \ell^* \cdot \varepsilon'$, the probability that $\mathcal{B}_2$ aborts is negligible. Since $\varepsilon' \geq \varepsilon - \delta$, if $\ell^* > \frac{\ell}{\varepsilon - \delta}$ then this will hold.

If $\mathcal{B}_2$ does not abort then its probability of success is identical to $\mathcal{A}_4$'s probability of success. Since L was chosen randomly and any string XORed with a random string is a random string, $\widetilde{C_r^T} = \left[C_r^T\right]_{\mathsf{rInd}(\ell^*),\mathsf{rStr}(\ell^*)}$.

Thus $\mathcal{A}_4's$ view here is the same as in Game_3, so its probability of success is p_3. Hence $\mathcal{B}_2$'s probability of success, when accounting for the probability that it aborts, is $p_3 - \mathsf{negl}'$ for some negligible function negl'. By the ℓ-AONT security of T, $p_3 - \mathsf{negl}' < \frac{1}{2} + \mathsf{negl}''$ for some negligible function negl''. Therefore $p_3 < \frac{1}{2} + \mathsf{negl}$ for $\mathsf{negl} = \mathsf{negl}' + \mathsf{negl}''$. □

Now we use these lemmas to prove Theorem 16.

Proof. By Lemmas 20, 21, and 22, we see that the adversary's probability of success in Game_0 can only be negligibly greater than its probability of success in Game_3. By Lemma 23, the adversary's success in Game_3 can be at most $\frac{1}{2} + \mathsf{negl}$ for some negligible function negl. Thus the adversary's success in Game_0, which is the real case, can be at most $\frac{1}{2} + \mathsf{negl}'$ for some negligible function negl'. Combined with Theorem 15, this proves that is $(1-\varepsilon)$-Adap-Revoke-RS-KP-ABE-secure. □

F.3 Static-Revocable-RS-KP-ABE Security

Theorem 17. *Assume the existence of a* RS-KP-ABE-*secure revocable-storage attribute-based encryption scheme* $\Pi = (\mathsf{Setup}, \mathsf{G}, \mathsf{E}, \mathsf{KU}, \mathsf{D}, \mathsf{CTU})$*, a symmetric-key encryption scheme* $\Pi_{sym} = \left(\mathsf{G}^{\mathsf{Sym}}, \mathsf{E}^{\mathsf{Sym}}, \mathsf{D}^{\mathsf{Sym}}\right)$*, and an ℓ-AONT T. Suppose that for the construction from Section 12.4, C^T comprises at least a fraction $1 - \delta$ of the total size of each ciphertext. Then for any $\varepsilon < 1$ with $\varepsilon > \delta$ and any $\ell^* > \frac{\ell}{\varepsilon - \delta}$, this construction is* $(1 - \varepsilon)$-Stat-Revoke-RS-KP-ABE-*secure.*

Proof. This proof is identical to the proof of Theorem 16, with the exception of how $\mathcal{B}_2$ responds to the challenge queries. In the proofs of Lemmas 20, 21, and 22, the adversary requests the bits of $\{C_{S^*,t^*+u}\}_{0 \leq u \leq r}$ to receive all at once instead of adaptively. In the proof of Lemmma 23 for the static case, it works as follows: $\mathcal{B}_2$ receives challenge query $M_0, M_1, S^*, t^*, r)$. Let bitPos_T be the set of bit positions requested that correspond to bits of C^T. $\mathcal{B}_2$ chooses random $L \in \binom{N}{\ell^*}$. If $|L \cap ([N] \setminus \mathsf{bitPos}_T)| < \ell$ then B_2 aborts the simulation. Otherwise, $\mathcal{B}_2$ generates random symmetric keys $k_0, k_1, \ldots, k_{r+1} \leftarrow \mathsf{G}^{\mathsf{Sym}}$ and random seeds $s_1, \ldots, s_{r+1}$. $\mathcal{B}_2$ then constructs $C_0^{pk}, \ldots, C_{r+1}^{pk}$ as well as $C_{u,u}^{bks}, \ldots, C_{u,r+1}^{bks}$ for $u \in \{1, \ldots, r\}$, following the procedure in Lemma 21. $\mathcal{B}_2$ makes AONT challenge query

$$\left(\mathsf{E}^{\mathsf{Sym}}(k_0, M_0), \mathsf{E}^{\mathsf{Sym}}(k_0, M_1), L \cap ([N] \setminus \mathsf{bitPos}_T)\right) ,$$

receiving

$$\left[C^T\right]_{L\cap([N]\setminus\mathsf{bitPos}_T)} = \left[T\left(\mathsf{E}^{\mathsf{Sym}}(k_0, M_b)\right)\right]_{L\cap([N]\setminus\mathsf{bitPos}_T)}$$

in response. $\mathcal{B}_2$ produces $C_0^T \in \{0,1\}^N$ by taking $\left[C^T\right]_{L\cap([N]\setminus\mathsf{bitPos}_T)}$ and filling in the bits from $L \cap ([N] \setminus \mathsf{bitPos}_T)$ arbitrarily (these bits will not affect the challenge ciphertexts), and then it computes

$$\{C_{S^*,t^*+u} = \left(C_u^{pk}, \left[C_{1,u}^{bks}, \ldots, C_{u,u}^{bks}\right], C_u^T\right)\}_{0\leq u\leq r}$$

for

$$C_u^T = \left[\cdots\left[\left[C_0^T\right]_{\mathsf{Ind}(s_1,\ell^*),\mathsf{Ctr}(k_1,\ell^*)}\right]\cdots\right]_{\mathsf{Ind}(s_u,\ell^*),\mathsf{Ctr}(k_u,\ell^*)} .$$

$\mathcal{B}_2$ constructs $\widetilde{C_r^T} \in \{0,1\}^N$ by setting the bits not in L as their corresponding values from C_r^T and choosing the remaining bits randomly. $\mathcal{B}_2$ gives $\mathcal{A}_4$ the requested $(1-\varepsilon)|C_r^*|$ bits of $\{C_{S^*,t^*+u}\}_{0\leq u\leq r}$ as well as

$$C_{S^*,t^*+r+1} = \left(C_{r+1}^{pk}, \left[C_{1,r+1}^{bks}, \ldots, C_{r,r+1}^{bks}, \mathsf{E}(S^*, (0,0), t^*+r+1)\right], \widetilde{C_r^T}\right) .$$

When considering the probability that $\mathcal{B}_2$ aborts when responding to $\mathcal{A}_4$'s challenge query, here we let $\ell' = |L \cap ([N] \setminus \mathsf{bitPos}_T)|$, which is the number of bit positions of C^T that are in L but not in bitPos_T. Again, $\mathcal{B}_2$ aborts if and only if $\ell' < \ell$. The upper bound on ℓ' is still the same as for the adaptive case by the exact same argument. Thus the probability that $\mathcal{B}_2$ aborts is again negligible as long as $\ell^* > \frac{\ell}{\varepsilon-\delta}$. □

G Adaptive AONT Construction

We note that based on the work of Dodis et al. [15], we can construct adaptive AONT based on the existence of Adaptive Exposure Resistant Functions (ERF). This work builds on the work of Canetti et al. [12], who show in the non-adaptive setting how to build AONT from ERFs. Both works strive to prove their results in the standard model. Since our goal is strict efficiency for a practical problem, we note that their constructions can be efficiently lifted to the Random Oracle model. The results provide intuitive constructions based on OAEP similar to those of Boyko [11], and proofs that are much simpler, but at the expense of tightness in the security reduction.

G.1 Random Oracles as Adaptive ℓ-Exposure Resilient Functions

Exposure resilient functions are to a first approximation efficiently computable functions where the output appears random, even if some bits of the randomly chosen input are leaked to the adversary. We give the adaptive definition, as it is used in out constructions later. We refer the reader to Dodis et al. [15] for a more thorough discussion of this and related definitions, and constructions of these funcitons in the standard model.

Definition 21. *A PPT (oracle) computable function $f : \{0,1\}^n \to \{0,1\}^k$ is an adaptive $\ell-ERF$ with ε security, if for any PPT oracle adversary A:*

$$\Pr_{G,r}[A^{[r]_\ell,G}(f(r)) = 1] - \Pr_{G,r,R}[A^{[r]_\ell,G}(R) = 1] \leq \varepsilon,$$

where $[r]_\ell$ is an oracle that allows A to query i and receive the ith bit of r, up to $n-\ell$ queries; G denotes any applicable Random Oracles, and f may be dependent on G.

As noted by Canetti et al. [12], random oracles are ERFs. For completeness, we formalize this in the following claim:

Lemma 24. *Let $F : \{0,1\}^n \to \{0,1\}^k$ be a random oracle. Then $f(x) = F(x)$ is an adaptive $\ell-ERF$, for $\ell < n$ with $q/2^{\ell-1}$ security against an adversary limited to $q < 2^{\ell-1}$ queries to F.*

Proof. Note that for a given x, $f(x) = F(x)$ is a completely random output to A unless it otherwise queries $F(x)$. Therefore, our problem reduces to bounding the probability that in the experiment $\Pr_{G,r}[A^{[r]_\ell,G}(f(r))] = 1$ that A queries $F(r) = f(r)$, given its access to F and $[r]_\ell$ oracles. An adversary that will make q queries optimizes its probability of querying $F(r)$ by learning as much of r as possible from $[r]_\ell$ prior to making any direct queries to F, lest it make any queries that are incompatible with the bits learned about r. The probability of successfully querying r is now $\sum_{i=0}^{q-1} 1/(2^\ell - i) < q/(2^\ell - q) < q/(2^\ell - 2^{\ell-1}) = q/2^{\ell-1}$. □

G.2 Original OAEP as an Adaptive AONT in the Random Oracle Model

Canetti et al. [12] present a construction of a secret-only AONT based on OAEP in the non-adaptive setting. Unfortunately, this construction has the negative side-effect that it doubles the storage and download bandwidth for a given file. While storage is effectively free, the increase in access time if files are stored on disk, and the bandwidth and download effects are undesirable. Here we show that a slight modification of their arguments allows a traditional OAEP construction, as originally suggested by Boyko [11], can be proven adaptively secure.

Lemma 25. *Let $G : \{0,1\}^k \to \{0,1\}^n$, and $H : \{0,1\}^n \to \{0,1\}^k$ be random oracles. Define the probablistic function $f : \{0,1\}^n \to \{0,1\}^k$: $f(x;r) = \langle G(r) \oplus x, H(G(r) \oplus x) \oplus r) \rangle$, where $r \in_R \{0,1\}^k$. Let $\ell \leq k$, then f is an adaptive 2ℓ-AONT, with security $q/2^{\ell-2}$ for an adversary that makes at most $q < 2^{\ell-1}$ adaptive queries to G or H.*

Proof. We first note that an adaptive adversary A that can make ℓ queries to $\langle G(r) \oplus x, H(G(r) \oplus x) \oplus r) \rangle$ will either make a majority of its queries $\ell = \lceil 2\ell/2 \rceil$ to either the bits contained in $G(r) \oplus x$, or $H(G(r) \oplus x) \oplus r$. Therefore, it suffices to prove the result for both cases when ℓ queries are made to either $G(r) \oplus x$ or $H(G(r) \oplus x) \oplus r$, assuming the adversary would be given the entirety of the other oracle, after it had finished making its adaptive queries; more information cannot harm the adversary's success probability.

We consider the two cases seperately. We then take the minimum of the security guarantees to prove the lemma.

Case: ℓ queries to $G(r) \oplus x$. We need to bound:

$$\Pr_{G,H,r}[A^{[G(r)\oplus x_0]_\ell,G,H}(x_0, x_1, H(G(r) \oplus x_0) \oplus r) = 1] - \Pr_{G,H,r}[A^{[G(r)\oplus x_1]_\ell,G,H}(x_0, x_1, H(G(r) \oplus x_1) \oplus r) = 1] \ . \quad (1)$$

With an eye towards using Lemma 24, we define the event that A queries $G(r)$ as BAD, and bound its probability.

Claim. $\Pr[BAD] \leq q/2^\ell$

Proof. For an adversary with q queries to query $G(r)$, it has two strategies to consider: i) try to query H on $(G(r) \oplus x_b)$, as it knows all but ℓ bits of this string, and then use its input $H(G(r) \oplus x_b) \oplus r$ to retrieve r, or ii) brute-force queries to G in hopes of querying $G(r)$ directly.

The adversary optimizes its probability of querying $H(G(r) \oplus x_b)$ by learning as much of $G(r) \oplus x_b$ as possible from $[G(r) \oplus x_b]_\ell$ prior to making any direct queries to G, lest it make any queries that are incompatible with the bits learned about $G(r) \oplus x_b$. Then it will guess the remaining ℓ bits of $G(r) \oplus x_b$, query this to H, and then XOR the response with $H(G(r) \oplus x_0) \oplus r$ to produce its guess of r. This procedure will require one query to H and one to G. Thus the probability of successfully querying $G(r)$ is $\sum_{i=0}^{\lfloor q/2 \rfloor - 1} 1/(2^\ell - i) < (q/2)/(2^\ell - q/2) < (q/2)/(2^\ell - 2^{\ell-1}) = q/2^{\ell-1}$.

If the adversary tries to brute-force queries to G, then it has to guess the correct value of r out of 2^k possibilities. Thus its probability of success is $\sum_{i=0}^{q-1} 1/(2^k - i) < q/(2^k - q) < q/(2^k - 2^{k-1}) = q/2^{k-1}$. Since $k \geq \ell$, $q/2^{k-1} \leq q/2^{\ell-1}$. Thus in either case, $\Pr[BAD] \leq q/2^{\ell-1}$. □

We will use Lemma 10 to achieve our bound for A, so now we look at:

$$\Pr_{G,H,r}[A^{[G(r)\oplus x_0]_\ell,G,H}(x_0, x_1, H(R \oplus x_0) \oplus r) = 1|\overline{BAD}] - \Pr_{G,H,r}[A^{[G(r)\oplus x_1]_\ell,G,H}(x_0, x_1, H(R \oplus x_1) \oplus r) = 1|\overline{BAD}] \ . \quad (2)$$

Conditioned on $\overline{BAD}$, $G(r) = R$ for $R \in_R \{0,1\}^n$, so the above is equal to:

$$\Pr_{G,H,r}[A^{[R \oplus x_0]_\ell, G, H}(x_0, x_1, H(R \oplus x_0) \oplus r) = 1 | \overline{BAD}] - \Pr_{G,H,r}[A^{[R \oplus x_1]_\ell, G, H}(x_0, x_1, H(R \oplus x_1) \oplus r) = 1 | \overline{BAD}] \ , \quad (3)$$

which is equivalent to:

$$\Pr_{G,H,r}[A^{[R_0]_\ell, G, H}(x_0, x_1, H(R_0) \oplus r) = 1 | \overline{BAD}] - \Pr_{G,H,r}[A^{[R_1]_\ell, G, H}(x_0, x_1, H(R_1) \oplus r) = 1 | \overline{BAD}] \ , \quad (4)$$

where $R_b = R \oplus x_b$ ($b \in \{0,1\}$), and is uniformly distributed (given that R is). Since R_0 and R_0 are both uniformly random, $\Pr_{G,H,r}[A^{[R_0]_\ell, G, H}(x_0, x_1, H(R_0) \oplus r) = 1 | \overline{BAD}]$ and $\Pr_{G,H,r}[A^{[R_1]_\ell, G, H}(x_0, x_1, H(R_1) \oplus r) = 1 | \overline{BAD}]$ are equivalent, so

$$\Pr_{G,H,r}[A^{[G(r) \oplus x_0]_\ell, G, H}(x_0, x_1, H(R \oplus x_0) \oplus r) = 1 | \overline{BAD}] - \Pr_{G,H,r}[A^{[G(r) \oplus x_1]_\ell, G, H}(x_0, x_1, H(R \oplus x_1) \oplus r) = 1 | \overline{BAD}] = 0 \ . \quad (5)$$

Applying Lemma 10 we get that:

$$\Pr_{G,H,r}[A^{[G(r) \oplus x_0]_\ell, G, H}(x_0, x_1, H(G(r) \oplus x_0) \oplus r) = 1] - \Pr_{G,H,r}[A^{[G(r) \oplus x_1]_\ell, G, H}(x_0, x_1, H(G(r) \oplus x_1) \oplus r) = 1] \leq 0 + 2\Pr[BAD] = 2q/2^\ell = q/2^{\ell - 1}. \quad (6)$$

Case: ℓ' queries to $H(G(r) \oplus x$ This is a similar, but simpler case than the previous one. We need to bound:

$$\Pr_{G,H,r}[A^{H(G(r) \oplus x_0) \oplus r]_\ell, G, H}(x_0, x_1, G(r) \oplus x_0) = 1] - \Pr_{G,H,r}[A^{[(H(G(r) \oplus x_1) \oplus r]_\ell, G, H}(x_0, x_1, G(r) \oplus x_1)) = 1]. \quad (7)$$

We note that the adversary, given access to $G(r) \oplus x_b$ as part of its input, can query $H(G(r) \oplus x_b)$, and the result is that its dynamic oracle is equivalent to having access to $r = H(G(r) \oplus x_b) \oplus r \oplus H(G(r) \oplus x_b)$. Thus it suffices to bound:

$$\Pr_{G,H,r}[A^{[r]_\ell, G, H}(x_0, x_1, G(r) \oplus x_0) = 1] - \Pr_{G,H,r}[A^{[r]_\ell, G, H}(x_0, x_1, G(r) \oplus x_1)) = 1]. \quad (8)$$

Noting that G as a random oracle is an $\ell-$ERF, and applying Lemma 24 to both sides of the summand, we have that the above is strictly less than:

$$\Pr_{G,H,r,R_0}[A^{[r]_\ell,G,H}(x_0,x_1,R_0 \oplus x_0) = 1] - \\ \Pr_{G,H,r,R_1}[A^{[r]_\ell,G,H}(x_0,x_1,R_1 \oplus x_1)) = 1] + 2q/2^{\ell-1}. \quad (9)$$

Since R_0 and R_1 are uniformly random, $\Pr_{G,H,r,R_0}[A^{[r]_\ell,G,H}(x_0,x_1,R_0 \oplus x_0) = 1]$ and $\Pr_{G,H,r,R_1}[A^{[r]_\ell,G,H}(x_0,x_1,R_1 \oplus x_1)) = 1]$ are clearly equivalent, and thus the above is bounded by $2q/2^{\ell-1} = q/2^{\ell-2}$. □