Innovations in Computer Science 2011

Semantic Security Under Related-Key Attacks and Applications

Benny Applebaum[1] Danny Harnik[2] Yuval Ishai[3]
[1]School of Electrical Engineering, Tel-Aviv University [2]IBM Haifa Research Labs
[3]Department of Computer Science, Technion
benny.applebaum@gmail.com dannyh@il.ibm.com yuvali@cs.technion.ac.il

Abstract: In a related-key attack (RKA) an adversary attempts to break a cryptographic primitive by invoking the primitive with several secret keys which satisfy some known, or even chosen, relation. We initiate a formal study of RKA security for *randomized encryption* schemes. We begin by providing general definitions for semantic security under passive and active RKAs. We then focus on RKAs in which the keys satisfy known linear relations over some Abelian group. We construct simple and efficient schemes which resist such RKAs even when the adversary can choose the linear relation adaptively during the attack.
More concretely, we present two approaches for constructing RKA-secure encryption schemes. The first is based on standard randomized encryption schemes which additionally satisfy a natural "key-homomorphism" property. We instantiate this approach under number-theoretic or lattice-based assumptions such as the Decisional Diffie-Hellman (DDH) assumption and the Learning Noisy Linear Equations assumption. Our second approach is based on RKA-secure pseudorandom generators. This approach can yield either *deterministic, one-time use* schemes with optimal ciphertext size or randomized unlimited use schemes. We instantiate this approach by constructing a simple RKA-secure pseurodandom generator under a variant of the DDH assumption.
Finally, we present several applications of RKA-secure encryption by showing that previous protocols which made a specialized use of random oracles in the form of *operation respecting synthesizers* (Naor and Pinkas, Crypto 1999) or *correlation-robust hash functions* (Ishai et. al., Crypto 2003) can be instantiated with RKA-secure encryption schemes. This includes the Naor-Pinkas protocol for oblivious transfer (OT) with adaptive queries, the IKNP protocol for batch-OT, the optimized garbled circuit construction of Kolesnikov and Schneider (ICALP 2008), and other results in the area of secure computation. Hence, by plugging in our constructions we get instances of these protocols that are provably secure in the standard model under standard assumptions.

Keywords: cryptography, encryption, related-key attacks, correlation-robust hash functions, oblivious transfer.

1 Introduction

Encryption is the most useful and widely known cryptographic primitive. Encryption schemes are being used both as standalone applications (as means of achieving "private" communication), and as building blocks for more complicated cryptographic tasks (e.g., secure multiparty computation). At an intuitive level, a private-key encryption scheme employs a secret key k to "garble" a message m into a ciphertext c, such that only a user who knows the key k can recover the message m from c, and any other user learns "nothing" about the message m. Modern notions of security (e.g., semantic security [33] or CCA security [23, 58, 65]) formulated this intuition in a very strong way, granting the adversary full control of almost all aspects of the system including the ability to encrypt messages and to decrypt ciphertexts at his choice. These formulations (and others) have led to increasingly strong notions of security. However, in all these notions the adversary is assumed to have no control on the secret keys. That is, security is guaranteed as long as keys are chosen truly at random and are kept secret.

In the past decade, this requirement has been relaxed to capture scenarios where some information about the keys is leaked to the adversary either directly (cf.[1, 22, 24, 53, 57, 62, 67]) or indirectly in the form of key-dependent messages (cf. [2-4, 12-15, 17, 36, 37, 39]). The present paper continues this line of research by studying another relaxation of the "random key assumption." Specifically, we study the security of encryption schemes under *related-key attacks* (RKA). In such attacks, the adversary attempts

to break the encryption scheme by invoking it with several secret keys which satisfy some known relation. For example, the adversary may ask for encryptions under a tuple of keys $(k_1, \ldots, k_t)$ whose XOR-differences $\Delta_i = k_1 \oplus k_i$ are known, or even chosen by the adversary during the attack.

RKAs are widely used in the area of applied cryptography, especially in the cryptanalysis of block ciphers (and typically with respect to the XOR-relation). Such attacks were first considered by Biham [8] and Knudsen [44] in the early 1990's. They were intensively studied in the last decade [9, 10, 26, 42, 70], until the point where today RKA security is considered to be an important goal in the design of block ciphers [21]. Motivated by this state of affairs, Bellare and Kohno [7] initiated a theoretical study of RKA security for block ciphers, theoretically modeled by pseudorandom functions (PRFs) and pseudorandom permutations (PRPs). They defined RKA security with respect to a class of related-key-deriving (RKD) functions Φ which specify the key-relations available to the adversary, and considered an active (and adaptive) adversary who can choose the relation from Φ during the attack.

Despite some limited positive results, obtained in [7, 51] and more recently in [31], it turned out that even for relatively simple relations (such as the XOR relation) achieving RKA security is very challenging. (See Section 1.2.) Only very recently, this goal was met by Bellare and Cash [6], who constructed RKA-secure block ciphers based on a standard cryptographic assumption (i.e., hardness of the DDH/DLIN problem). While their construction forms an important feasibility result, it is relatively inefficient. Also, it is restricted to linear relations over groups of the form $\mathbb{Z}_q^*$ (or $\mathbb{Z}_q$) where q is a large prime, rather than XOR-related attacks which may be considered to be more realistic (as they manipulate individual bits).

1.1 Our contribution

We continue the study of RKA-secure primitives but shift the focus to *randomized* encryption schemes. That is, instead of asking for *pseudorandomness* under RKAs we examine *semantic security* under RKAs. Apart from being a natural question which deserves study in its own right, it turns out that a direct treatment of semantic security has an important advantage: it leads to simpler and more efficient schemes for richer classes of key relations. Furthermore, we show that such randomized encryption schemes can serve as useful building blocks for several applications. Specifically, we reconsider several high-level protocols from the literature which originally employ strong pseudorandom objects (typically implemented by random oracles), and show that these protocols can be modified to rely on RKA semantically secure (randomized) encryption schemes. This not only serves as additional motivation for our study, but also further motivates the general work in the cryptanalysis community on the RKA security of practical ciphers. Following is a more detailed account of our results.

1.1.1 Definitions and constructions

We begin by giving a general definition for semantic security under RKAs. Following [7], we consider RKA security with respect to a class of related-key-deriving (RKD) functions Φ which specify the key-relations available to the adversary. Roughly speaking, we let the adversary apply a chosen plaintext attack with respect to a set of keys $k_1, \ldots, k_t$ which are derived from a master key k via a known function ϕ taken from an RKD family Φ. The adversary's success is tested via a distinguishing game. We present two main variants: a passive RKA (PRKA) in which ϕ is chosen by the challenger, and an adaptive RKA (ARKA) in which the adversary can choose many ϕ's by himself adaptively during the game.

This work focuses mostly on RKA security for linear relations Φ^+ which generalize the aforementioned XOR relation to an arbitrary Abelian group. We show that variants of encryption schemes from the literature are in fact secure against such classes of RKAs. The security of these schemes can be based on number theoretic assumptions such as DDH, or coding/lattices assumptions such as learning parity with noise (LPN) or Regev's learning with errors (LWE) assumption [61, 66]. More concretely, these constructions can provide ARKA-security against linear functions over various different groups including $\mathbb{Z}_2^n$ and $\mathbb{Z}_{2^n}$ (for which we get the standard XOR and $+ \mod 2^n$ relations), as well as additive groups of the form $\mathbb{Z}_q^n$ for any $2 \leq q \leq 2^{\mathrm{poly}(n)}$ (not necessarily a prime). Security is achieved by exploiting key-homomorphism properties, i.e., the ability to transform an encryption of a message m under a key k into an encryption of the same message under $k + \Delta$. This property easily leads to RKA semantic security (while being seemingly insufficient for RKA-security of PRFs).

RKAs and the power of randomization

Our results indicate that there is a significant difference between randomized primitives (e.g., randomized encryption) which use some private randomness in addition to the private key, and deterministic primitives (e.g., PRGs, PRFs, or PRPs). Indeed, although symmetric encryption is typically considered to be more "complicated" than pseudorandom generators, it seems that in the RKA setting the latter are harder to construct. This view is also supported by the results of [31].

We do, however, also make progress on the question of realizing deterministic primitives by presenting a simple and efficient construction of an RKA-secure PRG (aka correlation robust hash function [40]). Roughly speaking, an RKA-secure PRG is a function h such that for a secret seed s and *public* random offsets Δ_i, the values $(h(\Delta_1 + s), \ldots, h(\Delta_t + s))$ are pseudo-random. We show that a function of the form $h(x) = g^{x^t}$ satisfies this requirement under a variant of the DDH assumption that was considered in several previous works (e.g., [18, 28, 34]). The assumption asserts that, in a suitable group, the power sequence $(g, g^x, g^{x^2}, \ldots, g^{x^i})$ is pseudorandom, for a random generator g, a random x, and any polynomial i. This construction yields RKA-secure *one-time* symmetric encryption scheme with optimal ciphertext size. (This should be contrasted with our randomized DDH-based construction in which the ciphertext must contain a pair of elements in a DDH group even when the message is much smaller.) It should be noted that the notion of one-time security in the RK context allows to encrypt a single message *for each* related key. Hence, this primitive is quite strong and, it implies *stateful* deterministic RKA-secure encryption scheme with optimal ciphertext length, or alternatively a randomized stateless scheme with some additional overhead in the ciphertext length.[1]

1.1.2 Applications

We show that several previous cryptographic protocols which were based on random oracles or non-standard primitives can be instantiated (after some modifications) by encryption schemes which provide RKA-security with respect to linear functions. Intuitively, such encryption schemes are useful in protocols in which one party prepares many different ciphertexts, out of which only some will be revealed to the other parties. Standard encryption schemes provide security only if the keys are fully independent, whereas RKA-secure encryption allows to partially "recycle" keys by deriving new keys from old ones via the known relation. This additional flexibility naturally leads to improvements in communication and computation; furthermore, as we will see, in some scenarios a clever usage of these properties allows to distribute the keys to the participants in a way that significantly reduces the computational overhead and/or provides stronger security guarantees. We now elaborate on these applications.

The Naor-Pinkas Adaptive OT

Oblivious transfer (OT) [16, 25, 43, 63] is a central cryptographic primitive which allows a receiver R to obtain a subset of the data items held by a sender S, without letting S know which items were selected. Naor and Pinkas [54] studied an adaptive version of k out of N OT, in which the receiver selects k out of N items adaptively one-by-one. (Subsequent constructions obtained better efficiency under stronger assumptions; see [18, 35] and references therein.) They described a construction which is based on a special new primitive called *Operation respecting synthesizer* and showed that such a primitive can be realized under the DDH assumption or by using a random oracle. We observe that the Naor-Pinkas protocol can be instantiated by a special form of encryption scheme which, in turn, can be realized from any symmetric encryption scheme which is ARKA secure over linear functions. Hence, we obtain lattice-based and LPN-based instantiations for their protocol.

The IKNP Batch OT

Efficiency is particularly crucial for oblivious transfer due to its extensive use in both protocols for general secure computation (e.g., [32, 41, 43, 49, 68]), as well as more specialized or practically-oriented protocols (e.g., [30, 47, 56]). Indeed, OTs, which typically require computationally expensive public-key operations, form the efficiency bottleneck in many secure computation protocols. This fact motivated [40] (following [5, 55]) to present a batch-OT protocol which efficiently extends a small number of OTs to many OTs. The construction from [40] uses a random oracle or alternatively a XOR-correlation robust hash function – a nonstandard primitive that can be seen

[1] To achieve stateful deterministic RKA-secure encryption, encrypt the i-th message m_i by $\mathsf{Enc}_s(m_i) = m_i \oplus h(s + i)$. A randomized stateless encryption can be achieved by letting $\mathsf{Enc}_s(m; r) = (r, m_i \oplus h(s + r))$. See Lemma 3.2.

as an RKA-secure PRG with respect to XORs. This primitive was presented in [40] with no concrete instantiation, except for suggesting that practical hash functions may serve as good heuristic instances. A similar primitive is also used in subsequent OT extension protocols which offer better efficiency in the case of security against malicious parties [38, 41, 59].

We show that the use of correlation-robust hash functions in these OT extension protocols can be instantiated with PRKA-secure one-time encryption scheme with respect to XORs. We also describe a modification of the construction that can be based on RKA-security with respect to linear relations over general groups. By plugging in our RKA-secure constructions, we get security in the standard model under the DDH assumption, LPN, or LWE. To the best of our knowledge, this is the first instantiation of the IKNP protocol or its variants in the standard model. As noted above, we also present a very efficient construction of correlation robust hash functions under a variant of the DDH assumption.

Optimized garbled circuit constructions

Yao's garbled circuit (GC) technique [69] (see [48]) is a powerful tool that allows to securely evaluate any two-party functionality represented as a Boolean circuit. Recently, progress has been made on improving the efficiency of GC-based protocols [49, 60], including some practical implementations [50, 52, 63]. In [46] it is shown how to eliminate the overhead of handling XOR-gates by relying on a random oracle or XOR-correlation robust functions. It turns out that here too, the primitive is used as a one-time encryption scheme and therefore one can use PRKA-secure encryption scheme instead. Although this leads "only" to an efficiency improvement by a constant factor, such savings can still be very beneficial especially for large or medium size circuits as demonstrated in [45, 65]. Another GC optimization which relies on XOR-correlation robust function can be found in [60]. In this application the use of related keys allows to protect the protocol against malicious parties by using an improved cut-and-choose technique.

Heuristic instantiations

We believe that the results of this work are useful even if one decides, due to efficiency considerations, to instantiate the above applications with a heuristic implementation (e.g., a practical hash function such as SHA128). This is for two reasons: First, knowing that such primitives can be instantiated under standard assumptions gives better confidence in the plausibility of heuristic constructions. Second, viewing the primitive as a non-adaptive RKA-secure scheme allows to rely on other heuristic solutions such as block ciphers, for which RKA security is well studied. Indeed, the security of, say, AES under *passive*-RKAs for linear functions is considered to be a very conservative assumption. This may be better than relying on non-standard (yet plausible) properties of a hash function such as correlation robustness. Moreover, as said before, the above applications further motivate the practical study of RKA-security for block ciphers.

1.2 Related work

Bellare and Kohno [7] were the first to study related-key attacks in a theoretical framework. Other than providing a formal definition for RKA-secure PRPs and PRFs and pointing to some of their applications, [7] attempted to characterize the classes of RKDs Φ under which RKA security is possible. They showed that RKA security (for PRFs and PRPs) is impossible even with respect to relatively simple relations, while for other classes of attacks they proved possibility results in the ideal cipher model. They also gave constructions in the standard model that resist *partial*-RKAs (i.e., key-relations that leave some part of the key untouched).

Lucks [51] further studied partial-RKAs and, in addition, showed that RKA-security with respect to linear relations can be achieved under non-standard number-theoretic assumptions. Goldenberg and Liskov [31] studied RKA security for more basic symmetric primitives such as one-way functions and pseudorandom generators. Their results indicate that the way from RKA-secure one-way functions to RKA-secure PRFs or even PRPs is "blocked" at the hard-core bit level. Specifically, while a single related-secret pseudorandom bit is sufficient and necessary to create RKA-secure block ciphers, such hard-core bits *cannot* be constructed via typical (black-box) proof techniques.

Finally, two very recent related works that were done concurrently to our work are [6, 11]. In [6] Bellare and Cash provided the first construction of a block cipher which is provably RKA-secure against linear relations based on a standard assumption (i.e., hardness of the DDH or the DLIN problem). Bitan-

sky and Canetti [11] studied a new notion of obfuscators and, among other things, showed that obfuscators for multibit point functions give rise to encryption schemes which remain secure under passive key-related attacks (extending similar connections that were made in [19, 20]). They also presented an obfuscator, whose security follows from a strong (non-standard) variant of the DDH assumption, which gives rise to an encryption scheme that satisfies *passive* RKA security with respect to a wide family of relations as well as active RKA security for linear relations.[2]

Organization

The rest of this paper is structured as follows. In Section 2 we define the notion of semantic-security under Related-Key Attacks and discuss some basic aspects of this notion. In Section 3 we present general tools for constructing RKA secure schemes, and use them to obtain constructions based on concrete cryptographic assumptions. The last two sections are devoted to applications of RKA security — batch-OT is constructed in Section 4, and adaptive-OT in Section 5.

2 Definitions

Symmetric encryption (syntax)

Typically, symmetric encryption schemes can be solely defined by a pair of encryption and decryption algorithms where keys are just random bit strings whose length is equal to the security parameter. However, for our number-theoretic constructions it will be convenient to assume that keys are drawn from other domains (e.g., some group $\mathbb{G}$) whose description is public and possibly generated randomly by some set-up algorithm once and for all. Formally, symmetric encryption scheme consists of three probabilistic-polynomial time algorithms $(\mathsf{SetUp}, \mathsf{Enc}, \mathsf{Dec})$ as follows: (1) The randomized algorithm SetUp is given a security parameter 1^n and outputs the description of the key space K from which secret keys are sampled uniformly at random. The description of K includes its size, and a circuit for sampling a random element. Typically, K is assumed to be an Abelian group and in this case its description also includes a circuit for implementing the group operations. The key space and its full description are given as a public parameter and are also passed implicitly to the encryption and decryption algorithms. (2) For $K \in \mathsf{SetUp}(1^n)$, the randomized encryption algorithm Enc takes a message m of length $\mathrm{poly}(n)$ and a secret key $k \in K$ and outputs a ciphertext c. The randomized decryption algorithm Dec takes a ciphertext c and a secret key $k \in K$ and outputs a plaintext. (3) Correctness: for every message m, the error probability $\Pr_{k \xleftarrow{R} K}[\mathsf{Dec}_k(\mathsf{Enc}_k(m)) \neq m]$, taken over the randomness of $\mathsf{Enc}, \mathsf{Dec}$ and $K \xleftarrow{R} \mathsf{SetUp}(1^n)$, is negligible in n.

[2] It should be mentioned, however, that passive RKA seems much weaker than active RKA. See Section 2.

Related-key-deriving functions (RKDs)

Our formal definition is based on the notion of related-key-deriving (RKD) functions [7]. Let Φ be a family of related-key-deriving (RKD) functions $\phi : K \to K^t$ which map a key to a tuple of related t keys. Formally, we think of these objects as infinite families which are indexed by all possible key-spaces $K \in \mathsf{SetUp}(1^n)$; i.e., for every $K \in \mathsf{SetUp}(1^n)$ the family Φ_K contains functions of the form $\phi : K \to K^{t(n)}$. We will always assume that Φ is equipped with an efficient algorithm B and a canonical (and typically natural) representation that allows to specify a function $\phi \in \Phi_K$ by a string $\sigma \in \{0,1\}^{\mathrm{poly}(n)}$ where $B(K, \sigma, \cdot)$ evaluates ϕ.

Adaptive RKA Security

Adaptive Related-Key (ARK) security is defined with respect to Φ via the following game that takes place between a challenger and an adversary $\mathcal{A}$. For a security parameter n the game proceeds as follows:

- **Initialization.** The challenger chooses a key space K by invoking the algorithm $\mathsf{SetUp}(1^n)$. Then it randomly chooses a secret key $k_0 \xleftarrow{R} K$ and a challenge bit $b \xleftarrow{R} \{0,1\}$. The challenger sends K to the adversary.
- **Queries.** The adversary asks polynomially-many queries, where each query is of the form $(\phi, m_0, m_1, \ldots, m_t)$ where $\phi \in \Phi_K$. For $1 \leq i \leq t$, let k_i be the i-th entry of $\phi(k)$. The challenger responds with the tuple

$$c \xleftarrow{R} \begin{cases} (\mathsf{Enc}(k_j, m_j))_{j=0..t} & \text{if } b = 1, \\ (\mathsf{Enc}(k_j, 0^{|m_j|})_{j=0..t} & \text{if } b = 0. \end{cases}$$

- **Final phase.** The adversary attempts to guess b and outputs a bit $b' \in \{0,1\}$.

Definition 2.1. (*ARKA-secure encryption*) *A symmetric encryption scheme* $(\mathsf{SetUp}, \mathsf{Enc}, \mathsf{Dec})$ *is semantically-secure under Adaptive Related-Key attacks* (*in short, ARKA-secure*) *with respect to an RKD ensemble* Φ *if every polynomial-time attacker* $\mathcal{A}$ *has no more than negligible advantage over* $\frac{1}{2}$ *in guessing the value of the bit* b *in the above game* (*where the running time and the advantage are measured as functions of the security parameter* n).

Remarks

- (Avoiding trivialities.) Our syntactic definition requires that all but negligible fraction of the keys in the key space K be valid, i.e., respect correctness. Without this property (which is also crucial for applications), ARKA-security can be easily achieved (even for arbitrary functions) by adding some redundancy to the keys such that $\phi(k)$ will result in an invalid key.[3]
- (RKA $\Rightarrow$ Semantic security.) It is not hard to show that for any family Φ breaking ARKA-security is at least as hard as breaking standard semantic security. Indeed, a standard chosen-plaintext attack can be emulated by an ARK attack in which the adversary restricts its attention to ciphertexts generated under k_0 and ignores all other ciphertexts.
- (Impossible families) There are RKD families for which RKA security is impossible to realize. Consider, for example, the function ϕ_0 which maps the key k to the all zero key $k_1 = 0$. In such a case, since k_1 is known to the adversary, it is easy to distinguish the real mode ($b = 1$) from the dummy mode ($b = 0$). (E.g., ask for an encryption c of some message $m \neq 0$ under k_1, and then check whether the ciphertext c decrypts to m under $k_1 = 0$.) More generally, if the function $\phi(k)$ does not leave enough entropy in each of the related keys (when $k_0 \xleftarrow{R} K$), then RKA security is impossible to achieve.

Relaxations

For some of our applications it suffices to consider a relaxed notion of *passive* RKA (PRKA for short) in which the function ϕ is randomly chosen by the challenger. This relaxation is obtained by modifying the ARKA game as follows. At the initialization phase the challenger generates K and chooses k_0 and b as before. In addition, it chooses a KDM function $\phi \xleftarrow{R} \Phi_K$ and defines a vector of t keys by $(k_1, \ldots, k_t) = \phi(k_0)$. It sends K and ϕ to the adversary. At the query phase the adversary is allowed to ask polynomially any queries of the form (i, m) where $0 \leq i \leq t$ and m is in the message space, and the challenger responds with either with a real encryption $\mathsf{Enc}(k_i, m)$ or with a dummy encryption $\mathsf{Enc}(k_i, 0^{|m|})$ depending on the value of b. As before the goal of the adversary is to guess the bit b. (See Appendix A for formal definition.) This notion can be further relaxed to the case of one-time encryption schemes by restricting the adversary to use only a single encryption query for each key k_i. To avoid trivialities, we assume that the length of t messages is bigger than the key length. (Otherwise a *perfectly* PRKA-secure one-time scheme can be constructed by letting disjoint parts of the key act in each invocation.) We note that this notion suffices for some applications, and that all the remarks made above about ARKA security also apply to the case of PRKA one-time encryption.

3 Constructions

We will focus on RKA security with respect to the RKD family of linear functions Φ_t^+. In this case, we think about the key space K as a group $\mathbb{G}$, and for each $\Delta = (\Delta_1, \ldots, \Delta_t) \in \mathbb{G}^t$ define $\phi_\Delta : \mathbb{G} \to \mathbb{G}^t$ to be the mapping $k \mapsto (k + \Delta_1, \ldots, k + \Delta_t)$. Passive RKA-security implies that the adversary cannot break the scheme when given the differences of a $(t + 1)$-tuple of random keys. Adaptive RKA provides this guarantee even if the adversary chooses the differences Δ_i by himself. Note that ARKA security under the family Φ_1^+ easily implies security under Φ_t^+ for any polynomial t. (As any RK query with $\phi_{(\Delta_1,\ldots,\Delta_t)}$ can be emulated by t calls to Φ_{Δ_i} for $i \in [t]$.) Hence, in such a case we say that the scheme is simple Φ^+ ARKA secure. Finally, observe that for $\mathbb{G} = \mathbb{F}_2^\ell$ we get the standard XOR-family $\Phi_t^\oplus$.

3.1 Generic tools

We will rely on two generic approaches for constructing RKA-secure encryption scheme, described in Section 3.1.1 and Section 3.1.2.

[3] At the extreme, consider a scheme in which the key space K contains only a single valid key k (uniquely defined via some information given as part of the public parameters such as point obfuscator). The encryption and decryption algorithm will encrypt/decrypt only after verifying that the given key is the right one. Such a scheme provides security against any Φ but does not satisfy our syntactic definition.

3.1.1 Key-homomorphism

Let $\mathcal{E} = (\mathsf{SetUp}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric encryption scheme where the key space is a family of groups $\mathbb{G}_n$. We say that $\mathcal{E}$ has *key-homomorphism* if there exists an efficient algorithm (key-homomorphism) H that maps a ciphertext c and a shift amount Δ into a new ciphertext c' such that for every $k, \Delta \in \mathbb{G}_n$ and message m the random variable $c = \mathsf{Enc}_{k+\Delta}(m)$ is distributed identically to the random variable $c' = H(\mathsf{Enc}_k(m), \Delta)$ where the distribution is induced by the random coins of Enc and H.

Lemma 3.1. *A semantically-secure encryption scheme with key-homomorphism is adaptively-RKA secure with respect to linear RKD's, i.e.,* Φ^+.

Sketch. We use the homomorphism to convert an RKA adversary into an adversary that uses only queries to the original key k_0 (i.e., a CPA adversary). When the RKA adversary $\mathcal{A}$ asks for an encryption under the key $k + \Delta$ we will ask for $\mathsf{Enc}_k(m)$ and use the key-homomorphism H to translate it into $\mathsf{Enc}_{k+\Delta}(m)$. We will end with the same output of $\mathcal{A}$. Since the view of $\mathcal{A}$ is distributed exactly as in the real game in both cases, namely when b is either zero or one, we get a CPA adversary that breaks semantic security with the same advantage of the RKA adversary. □

Note that, for the special case of *non-adaptive* RKA security under Φ^+_t it suffices to use a weaker notion of homomorphism in which H generates (and outputs) the random shift Δ by itself rather than taking it as an input. We will later apply Lemma 3.1 to (variants) of known encryption schemes and get schemes that achieve ARKA security for linear functions under the DDH assumption (Lemma 3.3) and under the LPN and LWE assumptions (Construction 3.6).

3.1.2 Correlation robust generators

RKA-security can be also obtained from correlation robust generators [40], which we define below. (The original term used in [40] is correlation robust hash function.) Let $t = t(n)$ be a polynomial, and let $\mathbb{G}_n, \mathbb{H}_n$ be two sequences of groups. We say that an efficiently computable function $h : \mathbb{G}_n \to \mathbb{H}_n$ is *t-correlation robust* if for a random and independent choice of $t(n)$ elements $s, \Delta_1, \ldots, \Delta_t(n) \in \mathbb{G}_n$, the joint distribution $(h(\Delta_1 + s), \ldots, h(\Delta_t + s))$ is pseudo-random *given* $\Delta_1, \ldots, \Delta_t$. More formally, the ensemble $(\Delta_1, h(\Delta_1 + s), \ldots, \Delta_{t(n)}, h(\Delta_{t(n)} + s))_n$ is computationally indistinguishable from $(\Delta_1, y_1, \ldots, \Delta_{t(n)}, y_{t(n)})_n$ where s and the Δ_i's are chosen uniformly and independently at random from $\mathbb{G}_n$ and the y_i's are chosen uniformly and independently at random from $\mathbb{H}_n$. The function h is *correlation robust* if it is t-correlation robust for every polynomial $t(\cdot)$.

We now observe that correlation robust generators give rise to both a *deterministic, one-time* RKA-secure encryption scheme with optimal ciphertext size and a randomized unlimited use ARKA-secure scheme.

Lemma 3.2. *If $h : \mathbb{G}_n \to \mathbb{H}_n$ is t-correlation robust then the symmetric encryption scheme* $(\mathsf{Enc}_s(m) = h(s) + m, \mathsf{Dec}_s(c) = c - h(s))$ *is* one-time *RKA secure with respect to Φ^+_{t-1}. Furthermore, if h is correlation robust then the scheme $\mathsf{Enc}_s(m; r) = (r, h(s + r) + m)$ is ARKA secure with respect to Φ^+.*

Proof. An adversary $\mathcal{A}$ that breaks the one-time RKA security of the first scheme can be used the break the pseudorandomness of h as follows. Given a challenge $(\Delta_1, c_1, \ldots, \Delta_t, c_t)$ we emulate the one-time RKA game with $\phi_{\Delta'_2,\ldots,\Delta'_t}$ where $\Delta'_i = \Delta_i - \Delta_1$ and toss the challenge coin b. (We pretend that $s + \Delta_1$ is the original key k to be attacked.) When the adversary asks for an encryption of m under the i-th key (which can happen only once per key) we answer with $c_{i+1} + m$ if $b = 1$, and with c_{i+1} otherwise. At the end we output "pseudorandom" if and only if the output b' of the adversary equals to b. Observe that if the challenges $c_1, \ldots, c_t$ were truly random then the adversary cannot win with probability better than $\frac{1}{2}$ as in both modes, $b = 0$ and $b = 1$, the ciphertext distribution is uniform (and independent of the messages). On the other hand, if the challenge vector is pseudorandom then the view of the adversary is distributed exactly as in the real game where $k_i = s + \Delta_{i+1}$. Hence, an adversary which breaks the scheme with advantage ε results in a distinguisher with similar advantage, and the first part of the claim follows.

We move to the second scheme. Recall that to prove ARKA security with respect to Φ^+ it suffices to show ARKA security with respect to Φ^+_1. Furthermore, we can assume, wlog, that each query is of the form (δ_i, m_i) and is answered by $\mathsf{Enc}_{k_0+\delta_i}(m_i)$ as queries to $\mathsf{Enc}_{k_0}(\cdot)$ can be emulated by letting $\delta_i = 0$. Given an adversary $\mathcal{A}$ that breaks the ARKA security with respect to Φ^+_1 by making at most ℓ queries we break the ℓ-correlation robustness of h. Given $(\Delta_i, c_i)_{1 \le i \le \ell}$ we emulate the RKA game as follows. We think of s, the seed of h, as the "master key" k_0, and toss a challenge coin b. Given the i-th query of the adversary (δ_i, m_i),

we let $r_i = \Delta_i - \delta_i$, and answer the query with the ciphertexts $y_i = (r_i, c_i + b \cdot m_i)$. At the end we output "pseudorandom" if and only if the output b' of the adversary equals to b.

Again, if the challenge is truly random, the adversary's view in the case where $b = 0$ and $b = 1$ is identical, and consists only of random strings. Hence, in this case the adversary cannot guess the bit b with probability better than $\frac{1}{2}$. It remains to show that when the input is pseudorandom the view of the adversary is distributed identically to the real view. (As in this case, an adversary with advantage ε breaks the correlation robust PRG with advantage ε.)

Indeed, if $c_i = h(s + \Delta_i)$ then $y_i = \mathsf{Enc}_{k_0+\delta_i}(m_i \cdot b; r_i)$ where $r_i = \Delta_i - \delta_i$ is distributed uniformly and independently. Hence, the view of the adversary is distributed exactly as in the real game. □

3.2 Number-theoretic constructions

3.2.1 Decisional Diffie-Hellman

Our first concrete construction is based on a private-key version of El-Gamal [27]. Let GrpGen (for group generator) be an efficient probabilistic algorithm that given a security parameter 1^n generates parameters for some cyclic multiplicative group $\mathbb{G}$, including the group order q which is an n-bit integer, a generator g, and an efficient algorithm (e.g., circuit) for multiplication (and thus also exponentiation). We say that GrpGen satisfies the DDH assumption if the ensemble $(g, g^x, g^y, g^{xy})_n$ is computationally indistinguishable from a random tuple $(g, g^x, g^y, g^z)_n$ where g and the other public parameters are chosen by $\mathsf{GrpGen}(1^n)$ and $x, y, z \stackrel{R}{\leftarrow} \mathbb{Z}_{q_n}$.

Lemma 3.3 (Symmetric El-Gamal) *Consider the encryption scheme in which (1) public parameters $\mathbb{G}, g, q$ are generated via $\mathsf{GrpGen}(1^n)$; (2) a secret key k is chosen uniformly at random from $\mathbb{Z}_q$; (3) a message $m \in \mathbb{G}$ is encrypted by the pair $(a, a^k \cdot m)$ where $a \stackrel{R}{\leftarrow} \mathbb{G}$; and (4) a ciphertext (a, b) is decrypted by dividing b by a^k. Then, assuming that GrpGen satisfies the DDH assumption, the above construction is adaptively-RKA secure with respect to linear RKDs Φ^+ where addition is over $\mathbb{Z}_q$.*

Proof. The proof will follow from Lemma 3.1. First we show that the scheme is semantically secure under the DDH assumption. To see this, note that we could equivalently describe the encryption algorithm, as $\mathsf{Enc}_k(m) = (g^r, g^{rs} \cdot m)$ where $r \stackrel{R}{\leftarrow} \mathbb{Z}_q$ is the randomness of the scheme. This is exactly the El-Gamal scheme whose semantic security follows from DDH. (This is true even in the public-key version where g^k is public.) It remains to describe a key-homomorphism. Indeed, given a ciphertext (a, b) and $\Delta \in \mathbb{Z}_q$ we let $H((a, b), \Delta)$ be $(a, b \cdot a^{\Delta})$. The resulting ciphertext equals to $(a, a^{k+\Delta} \cdot m)$ which is the output of $\mathsf{Enc}_{k+\Delta}(m)$ when the randomness a is used, as required. □

Note that the construction makes only a single exponentiation for both encryption and decryption. However, it requires to transmit a group element even if we are interested in much smaller message space. Since ciphertext length is quite important for some of our applications (e.g., the batch OT), we present an alternative construction for one-time encryption scheme that relies on a variant of the DDH assumption that was used in several previous works. This construction employs a "correlation robust generator" as defined in Section 3.1.2.

3.2.2 Power Diffie-Hellman

The PDH assumption

Let $t = t(n)$ be a polynomial. We say that GrpGen satisfies the *t-Power-Diffie-Hellman* (PDH) assumption if the ensemble $(g, g^x, g^{x^2}, \ldots, g^{x^{t(n)}})_n$ is computationally indistinguishable from a random tuple $(g, g^{a_1}, g^{a_2}, \ldots, g^{a_{t(n)}})$ where g and the other public parameters are chosen by $\mathsf{GrpGen}(1^n)$ and $x, a_1, \ldots, a_t \stackrel{R}{\leftarrow} \mathbb{Z}^*_{q_n}$.

Lemma 3.4 (PDH-construction). *Suppose that GrpGen satisfies the t-PDH assumption for some polynomial $t(\cdot)$. Let $\mathbb{G}, g, q$ be public parameters generated by $\mathsf{GrpGen}(1^n)$ and let $h(k) = g^{k^t}$ where $k \stackrel{R}{\leftarrow} \mathbb{Z}_q$. Then the function h is a t-correlation robust generator with respect to addition in $\mathbb{Z}_q$.*

The proof of the lemma is based on the fact that, for any choice of distinct Δ_i, both the PDH tuple $(g^{k^i})_{i=1}^t$, and the t-correlated output of the generator $(g^{(\Delta_i+k)^t})_{i=1}^t$ represent a tuple of polynomials in k (hidden in the exponents) which form a basis for the set of polynomials of degree at most t. Hence, given the Δ_i's one can transform the first tuple to the second one. This transformation also takes the uniform distribution to itself and so reduces the security of the t-correlation robust generator to the PDH assumption. Formally, we will need the following standard fact:

Fact 3.5. *Let $\Delta_1, \ldots, \Delta_t \in \mathbb{Z}_q$ be distinct non-zero elements. Let $p_i(x)$ be the polynomial $(x+\Delta_i)^i$. Then the set of polynomials $P = \{p_0(x), \ldots, p_t(x)\}$ forms a basis for the linear space of polynomials of degree at most t.*

Proof of fact. Since P consists of $t+1$ polynomials of degree at most t it suffices to show that P is linearly independent. To see this, arrange the coefficients in an $(t+1) \times (t+1)$ matrix M whose j-th row consists of the $t+1$ coefficients of p_j. By the binomial theorem, the i-th coefficient of p_j is $\binom{t}{i}\Delta_j^{t-i}$ hence M is a Vandermonde matrix which has full rank since all the Δ_i's are distinct non-zero elements. □

Proof of Lemma 3.4 Let $t = t(n), q = q(n)$ and $k, \Delta_1, \ldots, \Delta_t \xleftarrow{R} \mathbb{Z}_q$ and $c_1, \ldots, c_t \xleftarrow{R} \mathbb{G}$. We will prove that the tuple

$$(g, \Delta_1, h(\Delta_1 + k), \ldots, \Delta_t, h(\Delta_t + k))$$

is computationally indistinguishable from

$$(g, \Delta_1, c_1, \ldots, \Delta_t, c_t)$$

based on the pseudorandomness of $(g, g^x, g^{x^2}, \ldots, g^{x^t})$. In fact, it suffices to prove this conditioned on the event that the Δ_i's are all distinct non-zero elements as this event happens with all but negligible probability t^2/q.

Let $(y_0, y_1, \ldots, y_t)$ be our PDH challenge. First, we choose a tuple of t random distinct non-zero elements $\Delta = \Delta_1, \ldots, \Delta_t \xleftarrow{R} \mathbb{Z}_q$. By Fact 3.5, there exists an invertible linear transformation L_Δ which maps the polynomials $(1, x^2, \ldots, x^t)$ to the polynomials $(h(x+\Delta_1), \ldots, h(x+\Delta_t))$. Since t is polynomially bounded we can also compute L_Δ efficiently. Now we apply L_Δ to the "exponent" of the y_i's. I.e., instead of summing we multiply and instead of multiplying by a constant c we raise to the power of c. Let $z = (z_0, \ldots, z_t)$ be the result. Suppose that the input was a PDH tuple. Then, the joint distribution of z and Δ is identical to the distribution $(g, \Delta_1, h(\Delta_1 + k), \ldots, \Delta_t, h(\Delta_t + k))$ (conditioned on the Δ_i's being distinct non-zero elements). On the other hand, if the input was a random tuple then the z_i's are truly random as the linear transformation L_Δ is of full rank. Hence, a distinguisher for h allows to break the PDH assumption. □

By Lemma 3.2, we get the following efficient construction of encryption scheme with one-time RKA security family with respect to $\Phi^+_{1,t}$. The key k is chosen uniformly at random from $\mathbb{Z}_q^*$, to encrypt a message $m \in \mathbb{G}$ we output the value $(g^{k^t} \cdot m)$, and to decrypt the ciphertext c divide it by g^{k^t}. To optimize efficiency one can take a small exponent $t = \text{poly}(n)$ which upper bounds the required related-key security. More importantly, if the message space is smaller than $\mathbb{G}$, we can hash down g^{k^t} and reduce the ciphertext length.

3.3 LPN/LWE-based constructions

The *learning parity with noise* problem is parameterized by positive integers n, t, and noise parameter $0 < \varepsilon < \frac{1}{2}$. The input to the problem is a random matrix $A \xleftarrow{R} \mathbb{F}_2^{t \times n}$ and a vector $y = As + e \in \mathbb{F}_2^t$ where $s \xleftarrow{R} \mathbb{F}_2$ and each entry of e is chosen independently according to the error distribution $\mathsf{Ber}_\varepsilon^t$ in which each entry is chosen to be 1 independently with probability ε. We say that the problem $\mathsf{LPN}_{t,\varepsilon}$ is *hard*, if there is no efficient adversary that can recover s from the input with more than negligible success probability.[4] We say that LPN_ε is *hard* if $\mathsf{LPN}_{t,\varepsilon}$ is hard for every polynomial $t(\cdot)$. We describe the symmetric encryption scheme of [2] which is a variant of the scheme of [29].

Construction 3.6 (LPN-construction). *Let $\ell = \ell(n)$ and $N = N(n)$ be an arbitrary polynomials. Let $\varepsilon < \frac{1}{2}$ be a constant error parameter and $0 < \delta < \frac{1}{2}$ be a constant. Let $G = \Omega G_\ell$ be an (ensemble of) $t \times \ell$ binary generator matrix of a family of linear error-correcting codes with efficient decoding algorithm D that can correct up to $(\varepsilon + \delta) \cdot t$ errors.*

- Secret-key: *The secret key of the scheme is a matrix S chosen uniformly at random from $\mathbb{F}_2^{n \times N}$.*
- Encryption: *To encrypt a message $M \in \mathbb{F}_2^{\ell \times N}$, choose a random $A \xleftarrow{R} \mathbb{F}_2^{t \times n}$ and a random noise matrix $E \xleftarrow{R} \mathsf{Ber}_\varepsilon^{t \times N}$. Output the ciphertext $(A, A \cdot S + E + G \cdot M)$.*
- Decryption: *Given a ciphertext (A, Z) apply the decoding algorithm D to each of the columns of the matrix $Z - AS$ and output the result.*

Efficiency and security

The scheme is highly efficient as encryption requires only cheap matrix operations and decryption requires

[4] This can be considered to be a "decoding game" where A generates a random linear code and the goal is to recover a random information word s given a noisy codeword y.

in addition to decode the code G. It is shown in [2] that for proper choice of parameters both encryption and decryption can be done in quasilinear time in the message length (for sufficiently long message). The above scheme is semantically secure assuming the intractability of the LPN_ε problem (see [2]). It is not hard to see that the scheme has a key-homomorphism by letting $H((A,Y),\Delta) = (A, Y + A \cdot \Delta)$ where $\Delta \in \mathbb{F}_2^{m \times N}$. Hence, by applying Lemma 3.1, and viewing the key as bit string of length $n \cdot N$, we get:

Lemma 3.7. *Assuming that LPN_ε is hard, the above construction is adaptively RKA-secure with respect to XOR-RKDs $\Phi^{\oplus}$.*

Extension to LWE

The LPN problem can be generalized by replacing the moduli 2 with a larger moduli $q = q(n) \leq 2^{\mathrm{poly}(n)}$, and by choosing each entry of the noise vector e from some distribution χ over $\mathbb{F}_q$. Typically, χ is taken to be $\bar{\Psi}_\alpha$ which is a discrete Gaussian centered around 0 with standard deviation αq. (Formally, we sample from $\bar{\Psi}_\alpha$ by drawing y from the Gaussian probability distribution whose density function is $\exp(-\pi(x/\alpha)^2)/\alpha$ and outputting $\lfloor q \cdot y \rceil \bmod q$). This version of the problem called *learning with error* ($\mathsf{LWE}_{q,\bar{\Psi}_\alpha}$) was introduced by Regev [66], who demonstrated strong evidence for its hardness. Specifically, Regev discovered a *quantum* reduction from approximating well-studied lattice problems to within $\tilde{O}(n/\alpha)$ factors in the *worst case* to solving $\mathsf{LWE}_{q,\bar{\Psi}_\alpha}$, when $\alpha \cdot q \geq n$ and q is polynomial in n. Recently, Peikert [61] also gave a related *classical* reduction for the case where q is exponential in n and all the prime factors of q are polynomially bounded.

Construction 3.6 can be generalized to the LWE variant in a natural way. That is, we choose the entries of the matrices S and A randomly from $\mathbb{F}_q$, and the entries of the matrix E from χ. The message space can be taken to be any arbitrary subset of $\mathbb{F}_q^{\ell \times N}$. Semantic security follows from a Lemma of [66] which shows that, assuming the hardness of $\mathsf{LWE}_{q,\chi}$, the distribution $(A, A \cdot S + E)$ is pseudorandom.[5] To enable decryption, we should employ an error-correcting code which corrects (whp) errors drawn from χ.[6]

[5] In fact, Regev [66] proves this lemma only for the case where $N = 1$. i.e., S is a vector. However, a simple hybrid argument shows that this is true for arbitrary polynomial N. See [2].

[6] For example, one can encode each bit b of the message by a symbol in $\mathbb{F}_q$ via the mapping $b \cdot \lceil q/2 \rceil$ this encoding works as long as $\chi < q/4$ with overwhelming probability. Of course, better ECC's can improve the rate of the encryption as long as the noise rate is not too large.

4 Batch OT from RKA security

4.1 High-level description

Oblivious transfer [25, 64] is a two-party protocol between a *sender* S and a *receiver* R. The sender holds a pair of strings and the receiver holds a selection bit. At the end of the protocol the receiver should learn just the selected string, and the sender should not gain any new information. Batch OT, $\mathtt{OT}_\ell^m$, realizes m (independent) oblivious transfers of ℓ-bit strings. Formally, this can be defined as a secure two-party protocol between a sender S and a receiver R realizing the following $\mathtt{OT}_\ell^m$ functionality: The input of S is m pairs $(x_{j,0}, x_{j,1})$,$1 \leq j \leq m$, where each $x_{j,b}$ is an ℓ-bit string, and the input of R is m selection bits $r = (r_1, \ldots, r_m)$. The output of R is x_{j,r_j} for $1 \leq j \leq m$, while S has no output.

In [40] it was shown how to efficiently extend a small number of OTs to many OTs. The construction uses a random oracle or a correlation robust generator with respect to XOR. We describe a variant of this construction which relies on one-time symmetric encryption with RKA-security under linear functions Φ_k^+ over general groups. Below we give a high level intuitive description of our version of the protocol. More details and proofs are deferred to Section 4.2. We focus for simplicity in the semi-honest setting. (The protocol can be adapted to the malicious model via cut-and-choose-techniques as in [38, 40, 59] or, with only a constant asymptotic overhead, by using the general compiler of [41].)

Our starting point is the standard fact that OT can be easily reduced to a randomized version of OT in which m pairs of random secret keys $(T_{1,0}, T_{1,1}), \ldots, (T_{m,0}, T_{m,1})$ are generated and given to the sender, while the receiver learns the keys $(T_{i,r_i})_{i=1}^m$ where $r = (r_1, \ldots, r_m)$ are the receiver's selection bits. Indeed, given such a functionality $\mathtt{OT}_\ell^m$ can be implemented by letting the sender use symmetric encryption scheme to encrypt the secret $x_{j,b}$ under the key $T_{j,b}$ and send all the ciphertexts to the receiver who can decrypt only the ciphertexts which correspond to the keys that he learned.

The first observation is that if the symmetric encryption scheme satisfies RKA-security under linear functions Φ_m^+ then the reduction still works even if $T_{i,0} = T_{i,1} + s$ for a random s as long as it is being kept hidden from the receiver. Next, we observe that the key-distribution functionality can be implemented in the "reverse" order: Let the receiver choose

the keys $(T_{i,r_i})_{i=1}^m$, let the sender choose the "shift" s and construct a protocol which allows the sender learn the $T_{0,i}$'s. Then have S set $T_{i,1}$ to $T_{i,0} - s$. Hence, for each i we would like S to learn the value $T_{i,r_i} + r_i \cdot s$. In the binary case, where the keys and the shift s are k-bit strings (and the encryption satisfies $\Phi_m^{\oplus}$-RKA security), this operation can be implemented by a single call to $\mathtt{OT}_m^k$ where R plays the role of the sender with input pairs $(T_{i,r_i}, T_{i,r_i} + r_i)$ and S plays the role of the receiver with selection vector s.

Hence we reduced $\mathtt{OT}_\ell^m$ to the "simpler" $\mathtt{OT}_m^k$. The efficiency gain here comes from the fact that the new "batch" parameter (which dominates the efficiency of the OT) depends only in the security parameter k of the symmetric scheme and is independent of the data size m. (The dependency of the length parameter in m has only minor effect on the efficiency.) After resolving some technicalities, it is possible to adapt the above solution the non-binary case with some minor loss in efficiency, i.e., logarithmic in the size of the group.[7]

The feasibility result established in this section can be summarized by the following variant of the main theorem from [40].

Theorem 4.1. *Let k be a security parameter. For any constant $c > 1$, there exists a protocol which reduces k^c instances of $\mathtt{OT}_k$ to k instances of $\mathtt{OT}_k$ and only makes a black-box use of any one-time symmetric encryption scheme which is RKA secure with respect to linear relations.*

4.2 Details and proofs

Our variant of [40] requires one-time RKA security under the RKD family $\Phi_t^{\pm}$ whose members are indexed by a shift vector $\Delta = (\Delta_1, \ldots, \Delta_t \in K^t$ and a sign vector $v \in \Omega\pm 1^t$ and the i-th component of $\phi_{\Delta,v}(k_0)$ is set to $\Delta_i + v_i \cdot k_0$. We note that in the special case, where the linear relation is XOR, this RKD family equals to Φ_t^{+}. We now show that any one-time scheme $(\mathsf{SetUp}, \mathsf{Enc}, \mathsf{Dec})$ which satisfies RKA under Φ_t^{+} can be easily converted to satisfy the new RKA under $\Phi_t^{\pm}$.

Lemma 4.2. *Let $\mathcal{E} = (\mathsf{SetUp}, \mathsf{Enc}, \mathsf{Dec})$ be an RKA secure one-time encryption scheme $(\mathsf{SetUp}, \mathsf{Enc}, \mathsf{Dec})$ under Φ_t^{+}. Consider the scheme $\mathcal{E}' = (\mathsf{SetUp}, \mathsf{Enc}', \mathsf{Dec}')$ where Enc'_k chooses a random bit $b \in \Omega\pm 1$ as part of its randomness, and outputs the pair $(\mathsf{Enc}_{b\cdot k}(m), b)$; and $\mathsf{Dec}'_k(c, b)$ applies $\mathsf{Dec}_{b\cdot k}$ to the first entry of the ciphertext c. Then, $(\mathsf{SetUp}, \mathsf{Enc}', \mathsf{Dec}')$ is a one-time $\Phi_t^{\pm}$-RKA secure scheme.*

Proof. We use an adversary $\mathcal{A}$ that breaks $\mathcal{E}'$ via $\Phi_t^{\pm}$-RKA to break $\mathcal{E}$ via Φ_t^{+}-RKA as follows. We begin the standard Φ_t^{+}-RKA game and let the challenger choose k_0 and publish $\Delta_1, \ldots, \Delta_t$. Then we choose a random sign vector $v \xleftarrow{R} \Omega\pm 1^t$, define $\Delta'_i = v_i \cdot \Delta_i$ and pass the values of the $\Delta'_1, \ldots, \Delta'_t$ to $\mathcal{A}$. Then, whenever the adversary asks for an encryption of a message m_i under the i-th key, we ask for $c_i = \mathsf{Enc}_{k+\Delta_i}(m)$ and pass the ciphertext (v_i, c_i) to the adversary.

Consider the joint view of the adversary: $(\Delta'_i, m_i, (v_i, c_i))_{i\in[t]}$. First observe that for each i the marginal distribution $(\Delta'_i, m_i, (v_i, c_i))$ is distributed properly, i.e., $(v_i, c_i) = \mathsf{Enc}'_{k+v_i\Delta'_i}(m_i)$. This is immidiate when $v_i = 1$, while for $v_i = -1$ it follows by writiting (v_i, c_i) as $\mathsf{Enc}_{k+\Delta_i}(m_i) = \mathsf{Enc}'_{-(k-\Delta'_i)}(m_i)$. Now, observe that all the pairs (Δ'_i, v_i) are distributed uniformly and independently (as the Δ_i's are uniform and are not part of the view). Hence, the view of the adversary is distributed exactly as in a real $\Phi_t^{\pm}$-RKA in both cases where the challenge bit is 0 and 1,and so we break $\mathcal{E}$ with the same advantage as $\mathcal{A}$ breaks $\mathcal{E}'$. □

Let us now present the (modified) IKNP protocol. We focus in the semi-honest version of the construction as the extension to the malicious setting follows easily from [40] or, more generally, from the generic transformation of [41]. Due to the well known "random-self-reducibility" property of OT, we may also assume that the selection bits of the receiver are chosen by the receiver uniformly at random (this version of OT reduces to the standard one via simple and efficient transformation). For simplicity, we begin with the special case where the scheme $(\mathsf{SetUp}, \mathsf{Enc}, \mathsf{Dec})$ is secure under $\Phi_k^{\oplus}$. In the following we adopt the notation of [40]. Let m be the desired number of OTs and $k \ll m$ be a security parameter. In Fig. 1 we describe how to reduce $\mathtt{OT}_\ell^m$ to $\mathtt{OT}_m^k$. We note that $\mathtt{OT}_m^k$ can be easily reduced to $\mathtt{OT}_k^k$ via a standard use of standard one-time symmetric encryption (e.g., pseudorandom generator). (See [40].)

The correctness of the protocol (when both parties are honest) follows directly from the correctness of

[7] In fact, security is a bit more subtle as one needs RKA-security against a KDM family which is slightly larger than the family Φ_t^{+} of linear functions. In particular, one needs to consider one-time RKA security under functions which either maps k_0 to $\Delta_i + k_0$ or to $\Delta_i - k_0$. We show that any RKA secure scheme (wrt linear functions) can be converted into one which supports this RKD family with almost no overhead.