CONJURA · OPEN PROBLEM

Update Lower Bounds for Registration-Based Encryption with Key-Dependent Update Times

Dropping the fixed-update-times hypothesis

Status. Statement: AI-written from Mohammad Mahmoody, Wei Qi and Ahmadreza Rahimi, *Lower bounds for the number of decryption updates in registration-based encryption*, IACR Cryptology ePrint Archive; the acknowledgements thank the anonymous reviewers of TCC 2022, so the paper appeared at TCC 2022, 2022, not yet checked by a human. Settled when the update schedule is a fixed function of the registration times (the paper's Theorem 4.1), and by its Theorem 4.12 also when the schedule is any fixed function of the sequence of registered identity names and of the CRS; open exactly when the schedule may depend on the sampled public keys, equivalently under the standard RBE completeness notion in which updates are fetched on demand after decryption returns GetUpd.

Category. *Information-Theoretic Cryptography.*

Conjecture (informal). *In registration-based encryption, each party generates its own keys and registers the public one with a key curator, who maintains a short public parameter that anybody can use to encrypt to any registered identity. Because that parameter is short, a party who registered earlier may have to come back to the curator from time to time and fetch "decryption update" information in order to keep decrypting. Every known scheme makes a party fetch about* $\log n$ *updates over the course of* n *registrations, and there is a nearly matching lower bound — but the lower bound only covers schemes whose update schedule is fixed in advance, so that a party knows from its own registration time and the number of parties registered so far exactly when it will need an update. The conjecture is that this restriction is inessential: the same trade-off between the number of updates and the length of the public parameter holds for every scheme, including one whose update schedule is allowed to react to the public keys that were registered. Concretely, a secure scheme whose public parameter stays polylogarithmic must force some party to fetch almost logarithmically many updates, however cleverly its update schedule depends on the keys.*

1 The setting

Registration-based encryption (RBE) was introduced by Garg, Hajiabadi, Mahmoody and Rahimi [GHMR18] to obtain the functionality of identity-based encryption without a private-key generator holding a master secret. Parties generate their own keys, register their public keys with a *key curator* (KC) who runs a deterministic, transparent algorithm, and the KC maintains a short public parameter pp_n that suffices to encrypt to any of the n registered identities. Since pp_n is required to be compact — $|\mathsf{pp}_n| \le \mathrm{poly}(\kappa, \log n)$ is the default level of compactness proposed in [GHMR18] — a registered party may need to periodically fetch *decryption update* information from the KC in order to keep decrypting ciphertexts created under later public parameters. If compactness is dropped, RBE is trivial (pp_n can simply be the concatenation of all public keys) and no updates are needed, so the interesting question is the trade-off between $|\mathsf{pp}_n|$ and the number of updates.

All known constructions — [GHMR18] from indistinguishability

obfuscation, [GHM19] from standard assumptions such as CDH or LWE, and the verifiable variant of Goyal and Vusirikala [GV20] — realise $|\mathsf{pp}_n| \le \mathrm{poly}(\kappa, \log n)$ with $\Theta(\log n)$ updates per party, because pp_n is (essentially) a Merkle-tree commitment to the registered public keys and a party's opening changes as that structure is updated. (The source paper describes the public parameter as a Merkle-tree root, p. 3; the precise shape of the accumulator in each construction is that work's, not this paper's.)

The source paper proves an almost matching lower bound: for any RBE whose *update times are fixed* — i.e. whether the i-th registered identity needs an update after the j-th registration is determined in advance by an "update graph" G on the registration indices — security forces $\binom{|\mathsf{pp}_n|+d}{d+1} \ge n$, and hence $d \ge \Omega(\log n / \log\log n)$ when $|\mathsf{pp}_n| \le \mathrm{poly}(\log n)$. The attack combines an information-theoretic part (the public parameter is short, so it has small mutual information with the keys of *some* group of parties, whose keys the adversary may therefore re-sample) with a combinatorial part (a low-out-degree forward DAG contains a long "skipping sequence", which tells the adversary which group to re-sample and which fake updates to manufacture for itself). The paper also extends the bound to update graphs that are an arbitrary fixed function of the registered identity names and of the CRS, and to schemes where only an average-case bound on the number of updates holds.

What remains is exactly the case where the update times may depend on the sampled *public keys*, equivalently the standard RBE completeness notion in which a party fetches an update precisely when decryption returns GetUpd. The paper isolates why its technique fails there: the partition of the keys into groups is derived from the update graph, so if the graph can be correlated with pp_n the graph itself carries information about the keys; and if one instead samples the graph from the execution, then re-sampling the keys of a group can change the update times, so even an adversary holding the *real* keys of that group may fail to decrypt. The paper points to data-dependent memory-hard functions [BH22], where provable barriers for data-*independent* graphs were overcome by making the graph depend on the input, as a precedent for the possibility that key-dependent update schedules genuinely help. Note also that a merely contrived key-dependence does not suffice to refute the statement: the refuting scheme must beat the trade-off, not just have

update times that formally mention the keys.

2 Notation and parameters

κ is the security parameter and $\mathrm{negl}(\kappa)$ denotes a negligible function. n is the number of registered identities, pp_i is the public parameter held by the key curator after i identities have registered, and

$$\alpha(n) \;=\; \max_{i \le n} |\mathsf{pp}_i|$$

is the maximum length (in bits) of the public parameter over the first n registrations, maximised over executions. $d(n)$ is the maximum, over executions in which at most n identities register, of the number of times the update algorithm Upd is invoked for a single identity. ρ is the completeness probability of the scheme (Definition 2 below), $\ell \in \mathbb{N}$ is a free parameter of the statement, δ is an inverse-polynomial slack, $\binom{a}{b}$ is a binomial coefficient, ln is the natural logarithm and log is the base-2 logarithm.

The parameters of the statement are:

- κ, the security parameter of the RBE scheme.
- n, the number of identities registered with the key curator.
- $d(n)$, the maximum number of decryption updates a single identity needs over the first n registrations.
- $\alpha(n)$, the maximum bit length of the public parameter over the first n registrations.
- ℓ, a free integer parameter of the trade-off; the binomial condition $n \ge \binom{\ell + d(n)}{d(n)+1}$ trades ℓ against $d(n)$.
- ρ, the completeness probability of the scheme.
- δ, an inverse-polynomial slack in the attack's success probability.

3 The conjecture

Definition 1 (Registration-based encryption, syntax). A *registration-based encryption* (RBE) scheme is a tuple of PPT algorithms $\Pi = (\mathsf{Gen}, \mathsf{Reg}, \mathsf{Enc}, \mathsf{Upd}, \mathsf{Dec})$, used together with a common random string $\mathsf{crs} \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}$ that is sampled once and published:

- $\mathsf{Gen}(1^\kappa) \to (\mathsf{pk}, \mathsf{sk})$: run locally by a party that wishes to register.
- $\mathsf{Reg}^{[\mathsf{aux}]}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}, \mathsf{pk}) \to \mathsf{pp}'$: *deterministic*, run by the *key curator* (KC) with read/write access to an auxiliary state aux that it may modify; it outputs the updated public parameter. The system is initialised with $\mathsf{pp} = \mathsf{aux} = \bot$.
- $\mathsf{Enc}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}, m) \to \mathsf{ct}$.
- $\mathsf{Upd}^{\mathsf{aux}}(\mathsf{pp}, \mathsf{id}, \mathsf{pk}) \to u$: *deterministic*, run by the KC with read-only access to aux; it outputs update information for id.
- $\mathsf{Dec}(\mathsf{sk}, u, \mathsf{ct}) \to m$: *deterministic*; it outputs a message $m \in \{0,1\}^*$, or $\bot$ (syntax error), or the special symbol GetUpd, indicating that more recent update information than u may be needed.

Definition 2 (ρ-completeness with on-demand updates). Consider the following game between a challenger $\mathcal{C}$ and a computationally unbounded adversary $\mathcal{A}$ that is limited to $\mathrm{poly}(\kappa)$ rounds of interaction. $\mathcal{C}$ sets $\mathsf{pp} = \mathsf{aux} = u = \bot$, $\mathcal{D} = \emptyset$, $\mathsf{id}^* = \bot$, $t = 0$, samples crs and sends it to $\mathcal{A}$. In each round $\mathcal{A}$ performs exactly one of:

1. **Registering a non-target identity.** $\mathcal{A}$ sends $(\mathsf{id}, \mathsf{pk})$ with $\mathsf{id} \notin \mathcal{D}$ (the key may be arbitrary and adversarial); $\mathcal{C}$ sets $\mathsf{pp} := \mathsf{Reg}^{[\mathsf{aux}]}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}, \mathsf{pk})$ and $\mathcal{D} := \mathcal{D} \cup \{\mathsf{id}\}$.
2. **Registering the target identity** (allowed only if $\mathsf{id}^* = \bot$). $\mathcal{A}$ sends $\mathsf{id}^* \notin \mathcal{D}$; $\mathcal{C}$ samples $(\mathsf{pk}^*, \mathsf{sk}^*) \leftarrow \mathsf{Gen}(1^\kappa)$,

sets $\mathsf{pp} := \mathsf{Reg}^{[\mathsf{aux}]}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}^*, \mathsf{pk}^*)$, $\mathcal{D} := \mathcal{D} \cup \{\mathsf{id}^*\}$, and returns pk^* to $\mathcal{A}$.

3. **Encrypting for the target** (allowed only if $\mathsf{id}^* \neq \bot$). C sets $t := t + 1$; $\mathcal{A}$ sends $m_t \in \{0,1\}^*$; C sets $m'_t := m_t$ and returns $\mathsf{ct}_t \leftarrow \mathsf{Enc}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}^*, m_t)$.

4. **Decrypting for the target.** $\mathcal{A}$ sends $j \in [t]$; C sets $m'_j := \mathsf{Dec}(\mathsf{sk}^*, u, \mathsf{ct}_j)$, and if $m'_j = \mathsf{GetUpd}$ then C sets $u := \mathsf{Upd}^{\mathsf{aux}}(\mathsf{pp}, \mathsf{id}^*, \mathsf{pk}^*)$ and re-sets $m'_j := \mathsf{Dec}(\mathsf{sk}^*, u, \mathsf{ct}_j)$.

$\mathcal{A}$ wins if $m'_j \neq m_j$ for some $j \in [t]$. The scheme is ρ-*complete with on-demand updates* if every such $\mathcal{A}$ wins with probability at most $1 - \rho$. The quantity $d(n)$ of the notation is the maximum number of invocations of Upd in step 4 across all executions of this game in which $|\mathcal{D}| \leq n$; no restriction is placed on *when* those invocations occur, so the update times may depend on the sampled keys and on crs.

Definition 3 (0-corruption security). Consider the following game between a PPT adversary $\mathcal{A}$ and a challenger C. C sets $\mathsf{pp} = \mathsf{aux} = \bot$, $\mathcal{D} = \emptyset$, samples crs and sends it to $\mathcal{A}$. Repeatedly, $\mathcal{A}$ sends some $\mathsf{id} \notin \mathcal{D}$; C samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\kappa)$, sets $\mathsf{pp} := \mathsf{Reg}^{[\mathsf{aux}]}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}, \mathsf{pk})$, $\mathcal{D} := \mathcal{D} \cup \{\mathsf{id}\}$, and returns pk to $\mathcal{A}$; the adversary never receives any secret key. Then $\mathcal{A}$ sends a target id^* (registered or not) together with two messages m_0, m_1 with $|m_0| = |m_1|$; C samples $b \leftarrow \{0,1\}$ and returns $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}^*, m_b)$; $\mathcal{A}$ outputs b' and wins if $b' = b$. The scheme is 0-*corruption secure* if every PPT $\mathcal{A}$ wins with probability at most $\frac{1}{2} + \mathrm{negl}(\kappa)$.

Conjecture 1 (Update lower bound for key-dependent update times). *Let* $\Pi = (\mathsf{Gen}, \mathsf{Reg}, \mathsf{Enc}, \mathsf{Upd}, \mathsf{Dec})$ *be an RBE scheme as in Definition 1 that is* ρ*-complete with on-demand updates as in Definition 2, and let* $n, d(n), \alpha(n)$ *be as in the notation. Then for*

every $n \in \mathbb{N}$, every $\ell \in \mathbb{N}$ with

$$n \geq \binom{\ell + d(n)}{d(n) + 1},$$

and every $\delta = 1/\mathrm{poly}(\kappa)$, there is a $\mathrm{poly}(\kappa)$-time adversary $\mathcal{A}$ that registers at most n identities and wins the 0-corruption security game of Definition 3 with probability at least

$$\rho - \sqrt{\frac{\alpha(n) \ln 2}{2\ell}} - \delta,$$

that is, with advantage at least $\rho - \frac{1}{2} - \sqrt{\alpha(n) \ln 2/(2\ell)} - \delta$. In particular, for every fixed κ: if Π is ρ-complete with $\rho \geq 0.99$, is 0-corruption secure, and satisfies $\alpha(n) \leq \mathrm{poly}(\kappa, \log n)$ for all n, then $d(n)$ is not $o(\log n / \log\log n)$; i.e. there are a constant $c > 0$ (depending on Π and κ) and infinitely many n with $d(n) \geq c \log n / \log\log n$.

Remark 1 (What is known). The paper proves the stated trade-off under the fixed-update-graph hypothesis, with the same conclusion and the same quantitative attack success probability (its Theorem 4.1), and derives the corollaries that $d = O(1)$ forces public parameters of length $\Omega(n^{1/(d+1)})$ and that polylogarithmic public parameters force $d = \Omega(\log n / \log\log n)$ (its Corollary 4.2). Its Section 4.4 removes the hypothesis in two directions: the update graph may be an arbitrary fixed function of the identity names and of the CRS (its Theorem 4.12), and it suffices that at least $\binom{\ell+d}{d+1}$ of the n identities receive at most d updates, which yields an amortized version (if the expected number of updates is d, then at least $n/2$ parties receive at most $2d$). The lower bound is proved against a weakened security notion (0-corruption, where the adversary gets no secret keys), which makes it stronger.

Remark 2 (Openness). The paper's own formulation, on the statement side: "Our main result provides an answer to the question above by proving an almost tight lower bound for the number of updates of any RBE schemes in which the update times are fixed" (p. 4). On the open case: "Our result leaves it open to either extend

our lower bound to RBE schemes with dynamic update times that depend on the public keys or to invent new RBE schemes with dynamic update times that bypass our lower bound" (p. 4); and again, in terms of the completeness notion, "Our completeness is stronger than (and implies) the standard completeness definition of RBEs; in our definition, the update times are fixed. It remains open to extend our lower bounds to the standard completeness definition of RBE or to find a new construction that bypasses our lower bound" (p. 11). On what might make a bypass possible: "As it remains open to potentially bypass our lower bound by leveraging on dynamic update times that depend on the registered keys, we point out a success story that might have some resemblance" (p. 10), and, on the failed repair, "Alternatively, one might try to first sample and fix the graph G based on the execution of the system" (p. 9).

Remark 3 (Caveats for a reviewer). Two things to check hardest. First, the formalisation of "at most d updates" once update times are dynamic: the worst case over executions of the on-demand completeness game has been taken here, counting invocations of Upd for the target identity. The paper does not fix this definition for the dynamic case (its Definition 2.4 is graph-based), so an average-case or with-high-probability variant is defensible; the paper's own amortized extension suggests such variants should also be in scope, and someone attacking the problem should be told which version they are settling. Second, direction: the paper poses this as a disjunction (extend the bound, or construct a scheme that bypasses it) and does not say which it believes; if anything its pointer to data-dependent memory-hard functions hints that the authors consider a bypass plausible, so the conjecture as stated may well be refuted rather than proved, and it should be presented as "prove or refute". Note also that the paper phrases this open problem twice (p. 4 in terms of dynamic update times, p. 11 in terms of the standard completeness definition); these have been treated as one problem, which appears right since on-demand updates are exactly key-dependent update times, but a reader who thinks they differ would see two candidates here. Finally, the RBE literature grew substantially after 2022 (registered ABE, pairing- and lattice-based schemes); no follow-up that either removes this hypothesis or exhibits a genuinely key-dependent update schedule is known to this record, but this

has not been verified against the post-2022 literature and someone should.

Remark 4 (Why it is worth settling). The fixed-update-times hypothesis is the single assumption standing between this theorem and an unconditional statement about all registration-based encryption. Settling it decides whether the logarithmic update cost of RBE is a real barrier for the primitive or an artifact of the fact that every known construction is built from Merkle forests whose shape depends only on how many parties have registered. A refutation would be at least as valuable: it would be the first RBE whose update schedule is genuinely key-dependent, a new design principle for the primitive, and the direct analogue of how data-dependent memory-hard functions escaped the barriers proved for data-independent ones. The statement is the paper's own main theorem with exactly one hypothesis deleted; nothing else about the model, the security notion or the quantitative bound changes. Everything needed (RBE syntax, standard completeness with GetUpd, 0-corruption security) is standard and fits on a page, and both resolutions are recognisable: a proof is an explicit polynomial-time attack on every compact RBE, a refutation is a construction that beats the trade-off.

4 Bibliography

[BH22] Jeremiah Blocki and Blake Holman. *Sustained space and cumulative complexity trade-offs for data-dependent memory-hard functions.* Cryptology ePrint Archive, Paper 2022/832, 2022.

[GHM19] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, Ahmadreza Rahimi, and Sruthi Sekar. *Registration-based encryption from standard assumptions.* PKC 2019: 22nd International Conference on Theory and Practice of Public Key Cryptography, Part II, LNCS 11443, pages 63–93, Springer, 2019.

[GHMR18] Sanjam Garg, Mohammad Hajiabadi, Mohammad Mahmoody, and Ahmadreza Rahimi. *Registration-based encryption: Removing private-key generator from IBE.* TCC 2018: 16th Theory of Cryptography Conference, Part I, LNCS 11239, pages 689–718, Springer, 2018.

[GV20] Rishab Goyal and Satyanarayana Vusirikala. *Verifiable registration-based encryption.* CRYPTO 2020, Part I, LNCS 12170, pages 621–651, Springer, 2020.