CONJURA · OPEN PROBLEM

The Exact Number of Decryption Updates in Compact Registration-Based Encryption

Closing the loglog factor for fixed update times

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. For schemes with fixed update times and polylogarithmic public parameters the number of updates is known to lie between $\Omega(\log n/\log\log n)$ (the paper's Corollary 4.2) and $O(\log n)$ (all known constructions), neither endpoint is known to be the truth, and the paper's own combinatorial tool is proved tight, so it cannot decide the question.

Category. *Information-Theoretic Cryptography.*

Conjecture (informal). *In registration-based encryption, a key curator keeps a short public parameter summarising the public keys of all registered parties, and a party who registered earlier must occasionally fetch "decryption update" information in order to keep decrypting. Every known scheme with a polylogarithmic public parameter makes each party fetch about* $\log n$ *updates over the course of* n *registrations. The best known lower bound is weaker by a* $\log\log n$ *factor: it only shows that some party must fetch about* $\log n/\log\log n$ *updates. The conjecture is that the constructions are the truth and the lower bound is what should move: with a polylogarithmic public parameter, logarithmically many updates are unavoidable. What makes this a concrete target rather than a wish is that the source paper proves the combinatorial engine of its lower bound — that a directed graph of small out-degree on many vertices contains a long "skipping sequence" — is exactly tight, so no sharper analysis of that engine can close the gap; a proof must find a different route, and a refutation must be a scheme with a short public parameter that beats* $\log n$ *updates.*

1 The setting

In registration-based encryption [GHMR18], parties generate their own keys and register their public keys with a key curator, who maintains a compact public parameter pp_n that anyone can use to encrypt to any of the n registered identities. Compactness forces registered parties to occasionally fetch *decryption update* information from the curator. All known schemes have the same efficiency profile: $|\mathsf{pp}_n| \le \mathrm{poly}(\kappa, \log n)$ together with $\Theta(\log n)$ updates per party — the obfuscation-based scheme of [GHMR18], the standard-assumption schemes of [GHM19], and the verifiable variant of [GV20]. In [GHMR18] the public parameter is the root of a Merkle tree hashing the registered public keys, so a party must refresh the opening (decommitment) to its own key as the tree grows, which yields $\Theta(\log n)$ updates (p. 3); and, as the paper's Remark 1.3 (pp. 9–10) explains, in all known constructions the accumulator consists of subsets whose sizes — and hence the update times — depend only on the number of identities registered so far, which is why all of them have fixed update times.

The source paper proves a lower bound in the opposite direction for all schemes with *fixed update times*, i.e. schemes whose update schedule is an update graph G fixed in advance. It states the trade-off in the clean form

$$\binom{|\mathsf{pp}_n| + d}{d+1} \;\ge\; n$$

as its Theorem 1.1 (p. 4), where $|\mathsf{pp}_n|$ is assumed non-decreasing (without loss of generality, by padding; cf. its footnote 5). The technical form is its Theorem 4.1 (p. 17), which bounds the success probability of an attack on a 0-corruption secure scheme with $|\mathsf{pp}_i| \le \alpha$ and $\deg^+(G_n) \le d$ whenever $n \ge \binom{\ell+d}{d+1}$, for a free integer parameter ℓ, and from which the proof of its Corollary 4.2 derives $\alpha(n) \ge \ell/10$ when $n = \binom{\ell+d}{d+1}$ and $\rho > 0.99$. Hence $d \ge \Omega(\log n / \log\log n)$ for $\alpha \le \mathrm{poly}(\log n)$, and $\alpha \ge \Omega(n^{1/(d+1)})$ for constant d. So for compact schemes with fixed update times the answer is pinned between $\Omega(\log n/\log\log n)$ and $O(\log n)$, and the paper flags closing this gap as open twice: once in the introduction, and once at the head of the section developing its combinatorial tool.

That tool is the notion of a *skipping sequence*: a set $\{u_1 < \cdots < u_k\}$ of vertices of a forward DAG such that for every $t < k$, no edge

leaves u_t into $\{u_{t+1}, u_{t+1}+1, \ldots, u_k\}$. Skipping sequences are what let the attacker group identities so that one group can manufacture the updates needed by its own first member, and the paper shows that every forward DAG with $\binom{k+d}{d+1}$ vertices and out-degree at most d contains a skipping sequence of size k. The length of the skipping sequence is what the information-theoretic half of the attack converts into a small mutual information with pp_n, so a longer guaranteed skipping sequence would directly improve the update lower bound. The obstruction the paper records is that this cannot happen: it constructs, for every k and d, a forward DAG with $\binom{k+d}{d+1} - 1$ vertices and out-degree d whose skipping sequences all have size at most $k-1$. The skipping-sequence route is therefore exhausted, and the $\log\log n$ factor has to be removed either by a genuinely different lower-bound technique or by a construction that meets the current bound.

2 Notation and parameters

κ is the security parameter and $\mathrm{negl}(\kappa)$ a negligible function. Identities are numbered by their registration order, so identity i is the i-th identity to register; n is the total number of registered identities and pp_i is the public parameter after i registrations. We write

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

for the maximum bit length of the public parameter over the first n registrations, maximised over executions. G is the (infinite) update graph of the scheme and G_n its restriction to the vertex set $[n] = \{1, \ldots, n\}$; $\deg^+(u) = |\{v : (u,v) \in G\}|$ is the out-degree of a vertex and $\deg^+(G_n) = \max_{u \in [n]} |\{v \le n : (u,v) \in G\}|$ is the maximum out-degree of G_n, which is exactly the largest number of decryption updates that a single identity receives while the first n identities register. ρ is the completeness probability, p a polynomial, c a positive constant, 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.

- G, G_n, the scheme's fixed update graph and its restriction to the first n registration indices.
- $\deg^+(G_n)$, the maximum out-degree of G_n, i.e. the worst-case number of updates an identity needs over n registrations.
- $\alpha(n)$, the maximum bit length of the public parameter over the first n registrations.
- ρ, the completeness probability of the scheme.
- p, a polynomial bounding the length of the public parameter as $p(\kappa, \log n)$.
- c, the positive constant in the conjectured logarithmic lower bound.

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 $\mathsf{crs} \leftarrow \{0,1\}^{\mathrm{poly}(\kappa)}$ sampled once and published:

- $\mathsf{Gen}(1^\kappa) \to (\mathsf{pk}, \mathsf{sk})$, run locally by a registering party;
- $\mathsf{Reg}^{[\mathsf{aux}]}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}, \mathsf{pk}) \to \mathsf{pp}'$, *deterministic*, run by the *key curator* (KC) with read/write access to an auxiliary state aux it may modify; the system starts 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;
- $\mathsf{Dec}(\mathsf{sk}, u, \mathsf{ct}) \to m$, *deterministic*, outputting a message $m \in \{0,1\}^*$, or $\bot$ (syntax error), or GetUpd (indicating that more recent update information than u may be needed).

Definition 2 (Forward DAG). A *forward DAG* is a directed acyclic graph G with vertex set $\mathbb{N}$ (or $[n]$) such that every edge $(i, j) \in G$ satisfies $i \le j$.

Definition 3 (ρ-completeness with fixed update graph G). Let G be an infinite forward DAG. Consider the following game between a challenger C and a computationally unbounded adversary $\mathcal{A}$ limited to $\mathrm{poly}(\kappa)$ rounds. 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 an identity.** Either (i) $\mathcal{A}$ sends a pair $(\mathsf{id}, \mathsf{pk})$ with $\mathsf{id} \notin \mathcal{D}$, where pk may be arbitrary and adversarial, and C sets $\mathsf{pp} := \mathsf{Reg}^{[\mathsf{aux}]}(\mathsf{crs}, \mathsf{pp}, \mathsf{id}, \mathsf{pk})$, $\mathcal{D} := \mathcal{D} \cup \{\mathsf{id}\}$; or (ii) if $\mathsf{id}^* = \bot$, $\mathcal{A}$ sends $\mathsf{id}^* \notin \mathcal{D}$ and 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 $\mathcal{A}$. In either case, if $\mathsf{id}^* \neq \bot$, letting i be the registration index of id^* and j the registration index of the identity just registered, C sets $u := \mathsf{Upd}^{\mathsf{aux}}(\mathsf{pp}, \mathsf{id}^*, \mathsf{pk}^*)$ whenever $(i, j) \in G$, and does nothing otherwise.

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

3. **Decrypting for the target.** $\mathcal{A}$ sends $j \in [t]$; C sets $m'_j := \mathsf{Dec}(\mathsf{sk}^*, u, \mathsf{ct}_j)$.

$\mathcal{A}$ wins if $m'_j \neq m_j$ for some $j \in [t]$; in particular $\mathcal{A}$ wins if decryption ever returns GetUpd, since updates are pushed only as G dictates. The scheme is *ρ-complete with fixed update graph G* if every such $\mathcal{A}$ wins with probability at most $1 - \rho$. Thus the number of updates received by identity i while the first n identities register is $|\{j \le n : (i, j) \in G\}|$, and the worst case over identities is $\deg^+(G_n)$.

Definition 4 (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 a secret key. Then $\mathcal{A}$ sends a target id^* and 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 (Logarithmically many updates are necessary). *Let Π be an RBE scheme as in Definition 1 and let G be an infinite forward DAG (Definition 2) such that Π is ρ-complete with fixed update graph G (Definition 3) for some $\rho \geq 0.99$, and such that Π is 0-corruption secure (Definition 4). Suppose further that there is a polynomial p with $\alpha(n) \leq p(\kappa, \log n)$ for all $n \in \mathbb{N}$.*
Then for every κ there are a constant $c > 0$ (depending on Π, p and κ) and infinitely many $n \in \mathbb{N}$ such that

$$\deg^+(G_n) \;\geq\; c \log n.$$

Equivalently: $\deg^+(G_n)$, the worst-case number of decryption updates that an identity needs over the first n registrations, is not $o(\log n)$.

Remark 1 (What is known). The paper states the trade-off $\binom{|\mathsf{pp}_n|+d}{d+1} \geq n$ for schemes with fixed update graphs as its Theorem 1.1 (p. 4), assuming $|\mathsf{pp}_n|$ non-decreasing (without loss of generality, by padding); its technical form is Theorem 4.1 (p. 17), from which the proof of Corollary 4.2 derives $\alpha(n) \geq \ell/10$ when $n = \binom{\ell+d}{d+1}$ and $\rho > 0.99$. This gives $d = \Omega(\log n / \log\log n)$ for $\alpha \leq \mathrm{poly}(\log n)$ and $\alpha = \Omega(n^{1/(d+1)})$ for constant d (Corollary 4.2). Appendix C proves the combinatorial half is exactly tight: for all $k \geq 1$ and $d \geq 0$ there is a forward DAG on $\binom{k+d}{d+1} - 1$ vertices with out-degree at most d having no skipping sequence of size k (Theorem C.1, Lemma C.3), so the skipping-sequence approach cannot yield more than $\log n / \log\log n$. The upper-bound side is unchanged since the original constructions: $\Theta(\log n)$ updates with

$\mathrm{poly}(\kappa, \log n)$ public parameters.

Remark 2 (Openness). The paper's statement of what it proves: "As a corollary, we find that RBE systems with fixed update times and public parameters of length $\mathrm{poly}(\log n)$, require $\Omega(\log n/\log\log n)$ decryption updates, which is optimal up to a $O(\log\log n)$ factor" (p. 1). On the gap: "In addition, it remains open to close the rather small gap of $1/\log\log n$ factor between our lower bound and the upper bounds of previously constructed RBE schemes" (p. 4), and again "This leaves open to close the gap between our lower bound and the upper bound of $\log n$ updates for future work" (p. 14). On why its own technique cannot close it: "In Section C, we prove that the bounds of this section are tight. This means that our approach of using skipping sequences cannot improve our lower bound of $\log n/\log\log n$ updates in RBEs" (p. 14). On the upper-bound side: "All the RBE schemes so far have the same asymptotic efficiency barriers built into them: they all use the same level of $\mathrm{poly}(\kappa, \log n)$ compactness for the public parameter and require $\Theta(\log n)$ number of updates to guarantee successful decryption" (p. 3).

Remark 3 (Caveats for a reviewer). The paper poses this as "close the gap" and does not commit to a side; the lower-bound direction has been chosen here so that there is a definite claim to prove or refute, and a reviewer may reasonably object that the paper's own phrasing is direction-neutral. On the substance, one should bet against the conjecture as stated: replacing the binary-counter Merkle forest of the known constructions by a b-ary counter with $b = \mathrm{poly}(\log n)$ plausibly gives $\mathrm{poly}(\kappa, \log n)$ public parameters (the roots of the $(b-1)\log_b n$ trees) with only $\log_b n = O(\log n/\log\log n)$ merges affecting any given party, hence $O(\log n/\log\log n)$ updates, which would meet the paper's bound and refute this statement. That observation is not the paper's, and it has not been checked whether the obfuscation- or garbling-based "delayed encryption" layer survives the larger arity with polylogarithmic ciphertexts and update lengths; if it does, the problem is much less hard than the paper's framing suggests, and this candidate should probably be dropped. Relatedly, this record has not been verified against the substantial post-2022 RBE literature to see whether such a scheme has already been published; if it has, this is resolved. Finally, note that the gap is a $\log\log n$ factor, so a reviewer who requires that

settling a conjecture change the qualitative state of knowledge may judge this below the bar even though the paper flags it as open twice.

Remark 4 (Why it is worth settling). The statement asks for the exact update complexity of a primitive whose whole point is that updates are the price of compactness, and the two candidate answers correspond to two different beliefs about the primitive: that Merkle-style constructions are optimal, or that a fundamentally cheaper update schedule exists. The paper has already done the work of showing that its own technique is spent, which is unusual and useful: whoever attacks the lower-bound side knows in advance that a new tool is required, and whoever attacks the construction side knows exactly what parameters to hit. The statement itself is a single inequality about one integer-valued quantity of a scheme — the maximum out-degree of its update graph, which is literally the number of updates a party needs — under two standard hypotheses (polylogarithmic public parameter, security). It needs only the RBE syntax, the fixed-update-graph completeness game and a weak security notion, all of which fit on a page, and it is falsifiable by a single construction.

4 Bibliography

[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.