# Updatable Public-Key Encryption from Class Groups: New Constructions with Better Efficiency

Harish Karthikeyan and Antigoni Polychroniadou

JPMorgan AI Research, JPMorgan AlgoCRYPT CoE
New York, NY, USA

**Abstract.** Updatable Public-Key Encryption (UPKE) enables forward-secure key evolution, protecting past messages in case of key compromise. We present efficient UPKE schemes, including a standard-model construction that can replace RTreeKEM (Alwen *et al.*, CRYPTO 2020) for secure group messaging. Our CPA-secure scheme under the Hard Subgroup Membership (HSM) assumption, requiring no trusted setup, achieves 1.9× smaller key and ciphertext sizes than the most efficient standard-model construction of Haidar *et al.* (CCS 2022, HLP). Encryption, decryption, and key-update operations are faster, with at least 2× improvement at 192-bit security. For CCA-like security in the random-oracle model, our construction is 2× faster than HLP at 128-bit security and has ciphertexts 4× smaller than LWE-based alternatives (Haidar *et al.*, ASIACRYPT 2023). We further propose an IND-CU-CCA UPKE scheme with one-round verifiable encryption, improving on prior multi-round approaches (Camenisch and Damgård, ASIACRYPT 2000). Our constructions combine practical efficiency, minimal trust assumptions, and strong forward- and post-compromise security, making them directly applicable to real-world secure group messaging systems.

## 1 Introduction

Forward security is essential in long-term communication, ensuring that past secrets remain protected even if current state is compromised. In symmetric cryptography, this is efficiently achieved via pseudorandom generators (PRGs) that evolve key states while hiding past outputs. Extending this property to public-key encryption (PKE), however, has proven significantly more challenging. Canetti *et al.* [13] formalized forward-secure PKE (FS-PKE), requiring synchronized evolution of public and secret keys across epochs. While realizations based on hierarchical IBE [26,22] exist, they remain complex and inefficient compared to standard PKE. To address these challenges, Jost *et al.* [27] introduced *updatable PKE* (UPKE), a relaxation of FS-PKE where senders assist receivers in evolving their secret keys using compact update ciphertexts.

**UPKE and Levels of Security.** Since its introduction by Jost *et al.* [27], both the syntax and the security definitions have evolved. In this work, we use the syntax formalized by Dodis *et al.* [20] (hereafter, DKW21) with five algorithms: key

Table 1: Taxonomy of UPKE Security Levels. An adversary is given access to the update oracle $\mathcal{O}_U$ and in some cases to the decryption oracle.

| **Notion** | **Input to $\mathcal{O}_U$** | **Oracle action** | **Dec. oracle** |
|---|---|---|---|
| CR-CPA | randomness $r_i$ | $\mathsf{Upd\text{-}Pk}(;r_i)$ then $\mathsf{Upd\text{-}Sk}$ | ✗ |
| CR-CCA | randomness $r_i$ | $\mathsf{Upd\text{-}Pk}(;r_i)$ then $\mathsf{Upd\text{-}Sk}$ | ✓ |
| CU-CPA | $(\mathsf{pk}_i, \mathsf{up}_i)$ | $\mathsf{Upd\text{-}Sk}(\mathsf{up}_i)$ | ✗ |
| CU-CCA | $(\mathsf{pk}_i, \mathsf{up}_i)$ | $\mathsf{Upd\text{-}Sk}(\mathsf{up}_i)$ | ✓ |

generation (U-Gen), encryption (U-Enc), decryption (U-Dec), public key update (Upd-Pk), and secret key update (Upd-Sk). Here, Upd-Pk uses randomness to produce an update ciphertext upct and new public key, while Upd-Sk consumes upct to evolve the secret key. We also focus on the four security levels they formalized. IND-CR-CPA (Chosen Randomness CPA) is the base level where the adversary selects the randomness used by the sender to generate update ciphertexts. IND-CR-CCA strengthens this by adding resilience against chosen-ciphertext attacks, allowing the adversary access to a decryption oracle. IND-CU-CPA (Chosen Update CPA) is a significantly stronger model where the adversary is not limited to choosing randomness but can maliciously craft their own update ciphertexts and public key pairs. Finally, IND-CU-CCA represents the most robust security level, combining the adversary's ability to provide malicious updates with access to a decryption oracle, requiring the scheme to remain secure even under active, malicious state evolution. The taxonomy is summarized in Table 1.

**Applications of UPKE.** UPKE has emerged as a central primitive in modern secure messaging and financial cryptography: it underlies the Signal protocol [28], enables post-compromise forward security in group messaging [1], and has recently been used to obtain forward-secure private blockchain transactions [23]. It is also a core component of the IETF Messaging Layer Security (MLS) standard [6,7]. While TreeKEM evolves group keys via a ratcheted tree, its security analysis [1,2] revealed gaps in forward secrecy and post-compromise security; integrating UPKE, as in RTreeKEM, yields stronger guarantees and motivates more efficient UPKE designs. Recent work has further expanded adversarial models to include insider attacks [3,19], highlighting the need for publicly verifiable update mechanisms. Devigne *et al.* [19] proposed a robust CGKA protocol that detects cheating updaters, resolving an open MLS question, but relies on generic verifiable encryption [11] and requires 136 rounds to reach $2^{-80}$ soundness. We summarize the comparison with relevant related work on UPKE in Table 2 while differing an expansive comparison with all related work to the full version of the paper. We emphasize that public verifiability is essential for update security: without it, receivers alone validate updates, enabling denial-of-service via inconsistent public keys. The scheme of [4] (based on Fujisaki–Okamoto [21]) lacks public verifiability, while [25] only sketches chosen-update security without a full construction.

Table 2: Comparison of UPKE constructions. Our scheme achieves strong security with public verifiability and efficient parameters.

| Work | Assumptions | Public Setup | CR-CPA | CR-CCA | CU-CCA | Impl. |
|---|---|---|---|---|---|---|
| [27,1] | CDH | ✓ | ROM | ✗ | ✗ | ✓ |
| [20] | DDH, LWE | ✓ | SM | SM | SM | ✗ |
| [24] | DCR | ✗ | SM | ROM | ROM | ✓ |
| [25] | LWE | ✓ | SM | ROM | ✗ | ✗ |
| [4] | CDH | ✓ | ROM | ROM | ✗ | ✗ |
| **Our Work** | HSM | ✓ | SM | ROM | ROM | ✓ |

**Our Motivation.** The research is motivated by the critical need for efficient forward security and post-compromise security in long-term communication systems, such as the Messaging Layer Security (MLS) standard and the Signal protocol. While symmetric cryptography achieves key evolution efficiently, extending these properties to public-key encryption has historically resulted in complex and inefficient constructions. Prior UPKE schemes often relied on heavy trust assumptions or resulted in unwieldy parameter sizes that made them impractical for real-world use. To this end, we design UPKE schemes under new assumptions, working over class groups that have been extensively used in cryptography, which improve efficiency and achieve strong security. Motivated by Devigne *et al.* [19], we further present an encryption scheme with efficient verification that avoids the repetitions inherent in the generic verifiable encryption of Camenisch–Damgård [11]. This work seeks to bridge the gap between theoretical security and practical performance by proposing new constructions from class groups that achieve up to 4x smaller ciphertexts and 2x faster operations while removing the requirement for a trusted setup.

**Our Contributions.** We now summarize our contributions:

1. **IND-CR-CPA from class groups.** We extend the Castagnos–Laguillaumie homomorphic encryption [17] and its variants [18,14,15,32,9] to obtain an IND-CR-CPA secure UPKE in the standard model. At 128-bit security, our parameters (Table 3) outperform all prior standard-model schemes. Notably, the underlying HSM assumption requires no trusted setup, unlike DCR.
2. **IND-CR-CCA in the ROM.** Applying the Naor–Yung transformation [30] with the generalized $\Sigma$-protocol of Braun *et al.* [10], we achieve IND-CR-CCA security in the random oracle model under the Strong Root and Rough Order assumptions. Our parameters (Table 3) significantly improve over DCR- and LWE-based schemes, though remain larger than the FO-based construction of Asano–Watanabe [4].
3. **IND-CU-CCA with verifiable updates.** We further obtain IND-CU-CCA security by augmenting the IND-CR-CCA scheme with a $\Sigma$-protocol proving update well-formedness. Unlike [4], which lacks verifiable updates, and [25], which omits a full construction, our scheme is complete and fully specified. Table 3 compares parameters with [24].

4. **Implementation and performance.** We implement all three schemes and empirically demonstrate substantial improvements in parameter sizes and running times over prior work (Section 6).
5. **Applications to secure group messaging.** Our IND-CR-CPA scheme can directly replace the UPKE used in forward-secure secure group messaging [1], enabling public setup, efficient key-update decryption, and natural support for verifiable encryption.
6. **Efficient verifiable encryption.** Finally, our IND-CU-CCA (or IND-CU-CPA) construction yields an efficient verifiable encryption protocol [11,12], ensuring consistency between update ciphertexts and public keys. This resolves an open problem of Devigne *et al.* [19], whose CGKA protocol relied on inefficient generic techniques. Further details appear in the full version.

## 2 Technical Overview

Our construction builds on the linear homomorphic encryption of Castagnos and Laguillaumie [17], whose security relies on the DDH assumption in a group with an easy DL subgroup. Castagnos *et al.* [18] later formalized this under the weaker Hard Subgroup Membership (HSM) assumption. Formally defined in Section 3.1, we consider a group $\mathbb{G}$ that factors as $\mathbb{G}^p$ (generated by $g_p$) and $\mathbb{F}$ (generated by $f$), where $\mathbb{F}$ has prime order $p$ and efficient discrete logs, while $\mathbb{G}^p$ has unknown order bounded by $\bar{s}$. Informally, HSM states that elements from $\mathbb{G}$ and $\mathbb{G}^p$ are computationally indistinguishable. To sample from the exponent space of $\mathbb{G}^p$, we use distributions $\mathcal{D}_p$ (resp. $\mathcal{D}$) such that $h \leftarrow_\$ \mathbb{G}^p$ is $2^{-\mu}$-close to $g_p^x$ for $x \leftarrow_\$ \mathcal{D}_p$ (resp. $h \leftarrow_\$ \mathbb{G}$ and $g^x$ for $x \leftarrow_\$ \mathcal{D}$). For concreteness, we set $\mathcal{D}_p$ to be uniform over $[\bar{s} \cdot 2^{40}]$ (Section 3.2).

**Circular Secure + Leakage Resilient (CS+LR) PKE.** The key generation samples $\mathsf{sk} \leftarrow_\$ \mathcal{D}_p$ and sets $\mathsf{pk} = g_p^{\mathsf{sk}}$. Messages $m \in \mathbb{Z}/p\mathbb{Z}$ are encrypted in an ElGamal-style encryption as $f^m$: the ciphertext is $(g_p^r, \mathsf{pk}^r \cdot f^m)$ for $r \leftarrow_\$ \mathcal{D}_p$. Decryption computes $M = w/c^{\mathsf{sk}}$ and recovers $m$ using the efficient DL solver in $\mathbb{F}$.

We prove CS+LR security for leakage $\ell = \mathsf{sk} + \delta^*$ with challenge ciphertexts $C = (g_p^r, \mathsf{pk}^r \cdot f^{m_b})$ and $C' = (g_p^t, \mathsf{pk}^t \cdot f^{\delta^*})$ via a hybrid argument:

1. Replace $\mathsf{sk} \leftarrow_\$ \mathcal{D}_p$ with $\mathcal{D}$; distributions are $2 \cdot 2^{-\mu}$-close.
2. Set $\mathsf{sk} = -\delta^*$, $\mathsf{pk}' = g_p^{-\delta^*}$, and adjust ciphertexts $(c, w) \mapsto (c, w \cdot (g_p^t)^{-\ell})$.
3. Apply the Smudging Lemma to make $\ell$ uniform, exploiting the large support of $\mathcal{D}$.
4. Use circular security under HSM: replace $C'$ with $D' = (g_p^t, (\mathsf{pk}')^t \cdot f^{-\delta^*})$, which is indistinguishable from $(f \cdot g_p^t, (\mathsf{pk}')^t)$.

The final step reduces to standard IND-CPA security of the PKE.

**CS+LR Secure PKE to IND-CR-CPA Secure UPKE in Standard Model.** We have described the key generation, encryption, and decryption algorithms. The $\mathsf{Upd\text{-}Pk}$ algorithm samples $\delta \leftarrow_\$ [p]$ and $r \leftarrow_\$ \mathcal{D}_p$ to compute the

Table 3: Comparison of parameter sizes (at 128-bit security) for UPKE constructions. Numbers for [25] are estimated assuming zero-correctness error and support for $2^{10}$ updates.

| Security Level | Work | $\lvert\mathsf{sk}\rvert$ | $\lvert\mathsf{pk}\rvert$ | $\lvert\mathsf{ct}\rvert$ | $\lvert\mathsf{upct}\rvert$ | Assumption |
|---|---|---|---|---|---|---|
| IND-CR-CPA | [27,1] | 256b | 512b | 512b | 512b | CDH |
| | [20] | 1.3kb | 328kb | 328kb | 419Mb | DDH |
| | [24] | 4.6kb | 6.1kb | 12kb | 12kb | DCR |
| | [25] | 3kb | 106kb | 24kb | 24kb | LWE |
| | **This Work** | **953b** | **3.1kb** | **6.2kb** | **6.2kb** | HSM |
| IND-CR-CCA | [4] (with [27]) | 256b | 640b | 768b | 512b | DDH |
| | [24] | 4.6kb | 6.1kb | 66kb | 12kb | DCR |
| | [25] | 3kb | 106kb | 24kb | 96kb | LWE |
| | **This Work** | **953b** | **3.1kb** | **14.6kb** | **6.2kb** | HSM |
| IND-CU-CCA | [24] | 4.6kb | 9.2kb | 91kb | 105kb | DCR |
| | **This Work** | **953b** | **6.2kb** | **14.6kb** | **14.6kb** | HSM |

update ciphertext $(g_p^r, \mathsf{pk}^r \cdot f^\delta)$, and updates the public key as $\mathsf{pk}' = \mathsf{pk} \cdot g_p^\delta$. The update secret key algorithm decrypts the update ciphertext to recover $\delta$ and updates the secret key consistently.

Our security proof adopts a simplified approach, bypassing earlier bit-by-bit offset encryption restrictions, which yields significantly better parameters (Table 3). For 128-bit security, Biasse *et al.* [8] conjectured a class group discriminant size $\Delta_k$ of 1872 bits, and RSA requires 3072 bits. In our scheme, $\bar{s}$ is 914 bits and the prime $p$ is 128 bits. Parameter sizes for related work are taken from HLP22.

**IND-CR-CPA to IND-CR-CCA Secure UPKE in the ROM.** We strengthen our construction to achieve IND-CR-CCA security via the Naor-Yung (NY) transform using suitable NIZKs. Note that ROM-based IND-CR-CPA schemes (e.g., hashed ElGamal) do not admit efficient NIZKs for NY.

To enable NY, we augment the IND-CR-CPA scheme with a second generator $h'$. Key generation remains unchanged, while encryption of a message $m$ under public key pk now outputs $(\mathbf{C}, \mathbf{D}, \pi_{\mathsf{NY}})$, where $\mathbf{C}$ encrypts $m$ under pk, $\mathbf{D}$ encrypts $m$ under $h'$, and $\pi_{\mathsf{NY}}$ is a NIZK proving equality of the underlying plaintexts. The update algorithms Upd-Pk and Upd-Sk remain unchanged. The generator $h'$ enables simulation of decryption queries and ensures simulation soundness against malicious ciphertexts.

We instantiate $\pi_{\mathsf{NY}}$ using a standard $\Sigma$-protocol for plaintext equality, following Braun *et al.* [10], achieving statistical soundness, SHVZK, and quasi-unique responses under the Strong Root Assumption. Applying the Fiat–Shamir transform yields a simulation-sound NIZK in the ROM. Together, these components yield an efficient IND-CR-CCA-secure UPKE scheme based on the underlying IND-CR-CPA construction.

**IND-CR-CCA Secure to IND-CU-CCA Secure UPKE in ROM.** Finally, we build our IND-CU-CCA secure UPKE from the previous IND-CR-CCA UPKE using the Naor-Yung transformation. We employ two generators, $h_p$ for

double encryption in the encryption algorithm and $h'_p$ for the update public key. To ensure update ciphertext well-formedness, we prove that the sampled $\delta$ is correctly used in both $\mathsf{upct}$ and $\mathsf{pk}' = \mathsf{pk} \cdot g_p^{\delta}$, using the generic $\Sigma$-protocol from [10] to construct $\pi_{\mathsf{upct}}$.

Our security proof leverages several observations: (1) the challenger generates updates only at the end of the game, so simulated NIZKs cannot be exploited for ill-formed updates, requiring only computational soundness; (2) the third public key $h'_p$ for update encryption prevents adversaries from using the challenge ciphertext in update queries and acts as a trapdoor to recover update randomness, reducing IND-CU-CCA security to IND-CR-CCA. The remainder of the proof follows standard techniques, with these insights being key to achieving full security.

**Implementation and Performance.** We provide a proof-of-concept implementation of our three constructions, focusing on demonstrating practicality rather than optimization. Benchmarks are reported for IND-CR-CPA, IND-CR-CCA, and IND-CU-CCA security across four security levels (112, 128, 192, 256 bits) and four operations (encryption, decryption, public key update, secret key update). We also compare our IND-CR-CPA and IND-CR-CCA constructions against Haidar *et al.* [24]. At 128-bit security, our IND-CR-CCA scheme is at least twice as fast, while our IND-CR-CPA scheme is at least twice as fast at 192 bits and outperforms three out of four operations at 128 bits. Table 5 summarizes these comparisons. For completeness, we also benchmark our IND-CU-CCA construction.

**Applications to Secure Messaging.** Prior work [1] showed that UPKE enables secure TreeKEM, a core component of SGM, allowing our IND-CR-CPA scheme to instantiate SGM in a black-box manner. Existing approaches based on hashed ElGamal are inefficient for proving update well-formedness, enabling denial-of-service attacks by malicious members. Devigne *et al.* [19] addressed this via NIZKs and verifiable encryption [11], but at the cost of many repetitions to reduce soundness error and additional checks for public key evolution when using UPKE. In contrast, our IND-CU-CCA secure UPKE (or IND-CU-CPA with proofs of well-formedness) yields an efficient, publicly verifiable instantiation of verifiable encryption without these overheads.

## 3 Preliminaries

We denote by $[x]$ for integer $x$ the following set: $\{0, \ldots, x-1\}$. Further, for a distribution $X$, we denote by $x \leftarrow\!\!\$\ X$ that $x$ is a random sample drawn from the distribution $X$, and similarly for $x \leftarrow\!\!\$\ S$ that $x$ is randomly sampled from the set $S$. We denote by $U_d$ the uniform distribution over $\{0,1\}^d$. For an element $g$ of a group $\mathbb{G}$, we denote by $\langle g \rangle$ the subgroup generated by $g$.

**Lemma 1 (Smudging Lemma [5]).** *Let $B_1 = B_1(\kappa)$ and $B_2 = B_2(\kappa)$ be positive integers and let $e_1 \in [-B_1, B_1]$ be a fixed integer. Let $e_2 \leftarrow\!\!\$\ [-B_2, B_2]$ be chosen uniformly at random. Then the distribution of $e_2$ is statistically indistinguishable from $e_1 + e_2$ as long as $B_1/B_2 = \mathsf{negl}(\kappa)$.*

### 3.1 CL Framework for Groups of Unknown Order

Before we present the CL framework, we look at the elements of the framework to better understand the variables involved. The elements of this framework are the following:

- *Input Parameters*: Computational parameter $\kappa$, statistical parameter $\lambda$, a prime $p > 2^\kappa$, and public randomness $\rho$ used by $\mathsf{CLGen}$.
- *Groups*: $\widehat{\mathbb{G}}$ is a finite abelian group; $\mathbb{G} \leqslant \widehat{\mathbb{G}}$ is cyclic; $\mathbb{F} \leqslant \mathbb{G}$ with order $p$; $\mathbb{G}^p = \{x^p : x \in \mathbb{G}\}$ with order $s$. Thus, $\mathbb{G} = \mathbb{F} \times \mathbb{G}^p$.
- *Orders*: $|\widehat{\mathbb{G}}| = p \cdot \widehat{s}$, $|\mathbb{G}| = p \cdot s$, with $s \mid \widehat{s}$, $\gcd(p, \widehat{s}) = \gcd(p, s) = 1$.
- *Generators*: $f$ generates $\mathbb{F}$, $g$ generates $\mathbb{G}$, $g_p$ generates $\mathbb{G}^p$, with $g = f \cdot g_p$.
- *Bounds*: Only an upper bound $\bar{s}$ of $\widehat{s}$ (and $s$) is known.
- *Encodings*: Only encodings of $\widehat{\mathbb{G}}$ are recognizable as valid; $s, \widehat{s}$ remain unknown.
- *Distributions*: $\mathcal{D}$ (resp. $\mathcal{D}_p$) are integer distributions such that $\{g^x : x \leftarrow\!\!\$\, \mathcal{D}\}$ (resp. $\{g_p^x : x \leftarrow\!\!\$\, \mathcal{D}_p\}$) is within distance $2^{-\lambda}$ of uniform over $\mathbb{G}$ (resp. $\mathbb{G}^p$).

**Definition 1 (CL Framework).** *The framework consists of* $(\mathsf{CLGen}, \mathsf{CLSolve})$*:*

- $\mathsf{pp} = (p, \kappa, \lambda, \bar{s}, f, g_p, \widehat{\mathbb{G}}, \mathbb{F}, \mathcal{D}, \mathcal{D}_p, \rho) \leftarrow\!\!\$\, \mathsf{CLGen}(1^\kappa, 1^\lambda, p; \rho)$,[1]
- $\mathsf{CLSolve}$ *efficiently solves discrete logarithms in* $\mathbb{F}$*, i.e.,*

$$\Pr\left[ x = x' \;\middle|\; \begin{array}{l} \mathsf{pp} \leftarrow\!\!\$\, \mathsf{CLGen}(1^\kappa, 1^\lambda, p; \rho), \\ x \leftarrow\!\!\$\, \mathbb{Z}/p\mathbb{Z},\ X = f^x, \\ x' \leftarrow \mathsf{CLSolve}(\mathsf{pp}, X) \end{array} \right] = 1.$$

We also have the following lemma from Castagnos, Imbert, and Laguillaumie [16] which defines how to sample from a discrete Gaussian distribution.

**Lemma 2.** *Let* $\mathbb{G}$ *be a cyclic group of order* $n$*, generated by* $g$*. Consider the random variable* $X$ *sampled uniformly from* $\mathbb{G}$*; as such it satisfies* $\Pr[X = h] = \frac{1}{n}$ *for all* $h \in \mathbb{G}$*. Now consider the random variable* $Y$ *with values in* $\mathbb{G}$ *as follows: draw* $y$ *from the discrete Gaussian distribution* $\mathcal{D}_{\mathbb{Z},\sigma}$ *with* $\sigma \geqslant n\sqrt{\frac{\ln(2(1+1/\epsilon))}{\pi}}$ *and set* $Y := g^y$*. Then, it holds that:* $\Delta(X, Y) \leqslant 2\epsilon$

### 3.2 Hardness Assumptions

Note that the framework is parameterized by additional distributions $\mathcal{D}, \mathcal{D}_p$. In groups of known order, elements can be sampled by choosing exponents modulo the order. Here, however, neither $|\mathbb{G}| = ps$ nor $|\mathbb{G}^p| = s$ is known. Instead, we rely on the upper bound $\bar{s}$ of $s$ to define $\mathcal{D}$ and $\mathcal{D}_p$, which provides flexibility in proofs. Specifically, we later set $\mathcal{D}_p$ to the uniform distribution over $[B]$ with

[1] The randomness $\rho$ is part of $\mathsf{pp}$, so no trusted setup is required.

$B = 2^\lambda \cdot \bar{s}$.[2] By Lemma 2, these distributions are within $2^{-\lambda}$ of uniform in $\mathbb{G}^p$. We will later optimize this sampling, and refer readers to [32, §3.1.3,§3.7] for further discussion. We will now introduce the various assumptions that we rely on for the security of our constructions.

**Definition 2 (Hard Subgroup Membership Assumption (HSM) Assumption [18]).** *Let $\kappa$ be a the security parameter with prime $p$ such that $|p| \geqslant \kappa$. Let* $(\mathsf{CLGen}, \mathsf{CLSolve})$ *be the group generator algorithms as defined in Definition 1, then the* ***HSM*** *assumption requires that that* ***HSM*** *problem be hard in $\mathbb{G}$ even with access to the* Solve *algorithm. More formally, let $\mathcal{D}$ (resp. $\mathcal{D}_p$) be a distribution over the set of integers such that the distribution $\{g^x : x \leftarrow\!\!\$\ \mathcal{D}\}$ (resp. $\{g_p^x : x \leftarrow\!\!\$\ \mathcal{D}_p\}$) is at most distance $2^{-\kappa}$ from the uniform distribution over $\mathbb{G}$ (resp. $\mathbb{G}^p$). Then, we say that the* ***HSM*** *problem is hard if for all PPT adversaries $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\kappa)$ *such that:*

$$\Pr\left[b = b' \;\middle|\; \begin{array}{c} \mathsf{pp} \leftarrow\!\!\$\ \mathsf{CLGen}(1^\kappa, 1^\lambda, p; \rho) \\ x \leftarrow\!\!\$\ \mathcal{D}, x' \leftarrow\!\!\$\ \mathcal{D}_p \\ b \leftarrow\!\!\$\ \{0,1\}; Z_0 = g^x; Z_1 = g_p^{x'} \\ b' \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{CLSolve}(\mathsf{pp},\cdot)}(\mathsf{pp}, Z_b) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\kappa)$$

**Definition 3 (Strong Root Assumption [15]).** *Let $\kappa$ be a the security parameter with prime $p$ such that $|p| \geqslant \kappa$. Let* $(\mathsf{CLGen}, \mathsf{CLSolve})$ *be the group generator algorithms as defined in Definition 1. Then, the Strong Root Assumption (****SR****) states that it is not possible to produce value $X, e$ such that $Y = X^e$ for a randomly chosen $Y$ from $\mathbb{G}^p$. More formally, for all PPT adversaries $\mathcal{A}$ there exists a negligible function such that:*

$$\Pr\left[\begin{array}{c} Y = X^e \wedge X \in \widehat{\mathbb{G}} \\ e \in \mathbb{Z} \wedge e > 0 \\ e \neq 2^k\ \forall k \end{array}\middle| \begin{array}{c} \mathsf{pp} \leftarrow\!\!\$\ \mathsf{CLGen}(1^\kappa, 1^\lambda, p; \rho) \\ Y \leftarrow\!\!\$\ \langle g_p \rangle \\ (X, e) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{CLSolve}(\mathsf{pp},\cdot)}(\mathsf{pp}, Y) \end{array}\right] \leqslant \mathsf{negl}(\kappa)$$

**Definition 4 (Rough Order ($\mathtt{RO}_C$) Assumption [10]).** *Let $\kappa$ be a the security parameter with prime $p$ such that $|p| \geqslant \kappa$. Let* $(\mathsf{CLGen}, \mathsf{CLSolve})$ *be the group generator algorithms as defined in Definition 1. Then, the Rough Order Assumption (****$\mathtt{RO}_C$****) states that the class groups with with no small prime factors in their order are indistinguishable from class groups in general. More formally, define $\mathcal{D}_C^{RO}$ to be the uniform distribution of the following set: $\{\rho \in \{0,1\}^\kappa | \mathsf{pp} \leftarrow\!\!\$\ \mathsf{CLGen}(1^\kappa, 1^\lambda, p; \rho) \wedge \forall \mathsf{prime} < \mathfrak{C} : \mathsf{prime} \nmid (p \cdot \widehat{s})\}$. Then,we say that the* ***$\mathtt{RO}_C$*** *problem is hard if for all PPT adversaries $\mathcal{A}$, there exists a negligible function* $\mathsf{negl}(\kappa)$ *such that:*

$$\Pr\left[b = b' \;\middle|\; \begin{array}{l} \rho_0 \leftarrow\!\!\$\ \{0,1\}^\kappa, \rho_1 \leftarrow\!\!\$\ \mathcal{D}_C^{RO} \\ b \leftarrow\!\!\$\ \{0,1\}; \\ b' \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{CLSolve}(\mathsf{pp},\cdot)}(1^\kappa, \rho_b) \end{array}\right] \leqslant \frac{1}{2} + \mathsf{negl}(\kappa)$$

[2] By definition, $\{g^x : x \leftarrow\!\!\$\ \mathcal{D}\}$ is statistically indistinguishable from $\{g^y : y \leftarrow\!\!\$\ \{0, \ldots, ps-1\}\}$, so $\{x \bmod ps : x \leftarrow\!\!\$\ \mathcal{D}\}$ is indistinguishable from uniform over $\{0, \ldots, ps-1\}$. Similarly, $\{x \bmod s : x \leftarrow\!\!\$\ \mathcal{D}_p\}$ is indistinguishable from uniform over $\{0, \ldots, s-1\}$. Since $s \mid ps$, sampling $x \leftarrow\!\!\$\ \mathcal{D}$ is also indistinguishable from uniform over $\{0, \ldots, s-1\}$.

Consider the relation

$$\mathcal{R} = \Big\{ (Y_i, \mathbf{X}_i)_{i=1}^{n}; \mathbf{w} \;\Big|\; Y_i = \prod_{j=1}^{m} X_{i,j}^{w_j},\ Y_i \in \widehat{\mathbb{G}},\ \mathbf{X}_i \in \widehat{\mathbb{G}}^m,\ \mathbf{w} \in \mathbb{Z}^m \Big\}.$$

**Theorem 1 (Theorem 9, [10]).** *For $\mathcal{R}$, there exists a canonical $\Sigma$-protocol with parameters $A, C \in \mathbb{N}$ that is:*

- *Sound with error $1/C + \mathsf{negl}(\kappa)$ in $C$-rough class groups.*
- *Complete for $\mathbf{w} \in [-S, +S]^m$.*
- *Statistical special honest-verifier zero-knowledge if $SC/A$ is negligible.*

## 4 Key-Dependent-Message-Secure Encryption Scheme

### 4.1 Constructions from CL Framework

We begin by presenting the construction of Castagnos et al. [18]. This generalizes the original construction of Castagnos and Laguillaumie [17], offering improved parameter choices for the underlying distributions and relying on the HSM assumption rather than DDH. We refer the reader to [18] for a detailed comparison between the two constructions.[3]

**Theorem 2.** *The PKE Scheme $\mathcal{E} = (\mathsf{U\text{-}Setup}, \mathsf{U\text{-}Gen}, \mathsf{U\text{-}Enc}, \mathsf{U\text{-}Dec})$ with the algorithms defined in Figure 1 is CS+LR Secure, under the HSM assumption.*

A brief summary of the proof technique was provided in Section 2. Due to space constraints the complete proof can be found in the full version of the paper.

## 5 Constructions of Efficient UPKE

This section presents our UPKE constructions; formal syntax and security definitions are deferred to the full version of the paper. We first give an IND-CR-CPA secure scheme in Section 5.1, followed by the NIZK tools underlying our CCA constructions. Section 5.2 develops a Naor–Yung-based argument of knowledge for class groups, and Section 5.3 proves update ciphertext well-formedness and consistent public key evolution. The complete constructions appear in the full version of the paper. Our approach upgrades security in two steps: (i) applying Naor–Yung with Fiat–Shamir to obtain IND-CR-CCA from IND-CR-CPA, and and (ii) combining this with well-formedness proofs to achieve IND-CU-CCA security.

[3] Bouvier et al. [9] propose a generic framework supporting arbitrary message spaces $\mathbb{Z}/M\mathbb{Z}$ (including prime, prime-power, and power-of-two moduli), subsuming the Camenisch–Shoup construction [12]. In the RSA setting $N = pq$, the HSM assumption coincides with Paillier's DCR assumption [31]. Haidar et al. [24] further build on this line of work with additional modifications.

**Protocol** IND-CR-CPA Secure UPKE

U-Setup$(1^\kappa)$: $\mathsf{pp} \leftarrow \mathsf{CLGen}(1^\kappa, 1^\lambda, p; \rho)$; return $\mathsf{pp}$.
U-Gen$(\mathsf{pp})$: Sample $sk \leftarrow [B]$; set $\mathsf{sk}_0 = (0, sk)$; $pk \leftarrow g_p^{sk}$; $\mathsf{pk}_0 = (0, pk)$; return $(\mathsf{pk}_0, \mathsf{sk}_0)$.
U-Enc$(\mathsf{pk}, m)$: Sample $r \leftarrow [B]$; parse $\mathsf{pk} = (\mathsf{ep}, pk)$; set $\mathsf{ct} = (g_p^r, pk^r \cdot f^m)$; return $\mathsf{ct}$.
U-Dec$(\mathsf{sk}, \mathsf{ct})$: Parse $\mathsf{ct} = (c_1, c_2)$; compute $M \leftarrow c_2 / c_1^{\mathsf{sk}}$; $m \leftarrow \mathsf{Solve}(\mathsf{pp}, M)$; return $m$.
Upd-Pk$(\mathsf{pk})$: Sample $\delta \leftarrow [p]$; parse $\mathsf{pk} = (\mathsf{ep}, pk)$; set $pk' = pk \cdot g_p^\delta$; sample $t \leftarrow [B]$; $\mathsf{upct} = (g_p^t, \mathsf{pk}^t \cdot f^\delta)$; $\mathsf{pk}' = (\mathsf{ep}+1, pk')$; return $(\mathsf{upct}, \mathsf{pk}')$.
Upd-Sk$(\mathsf{sk}, \mathsf{upct})$: Parse $\mathsf{upct} = (u_0, u_1)$, $\mathsf{sk} = (\mathsf{ep}, sk)$; compute $M \leftarrow u_1 / u_0^{sk}$; $\delta \leftarrow \mathsf{Solve}(\mathsf{pp}, M)$; $sk' \leftarrow sk + \delta$; $\mathsf{sk}' = (\mathsf{ep}+1, sk')$; return $\mathsf{sk}'$.

Fig. 1: IND-CR-CPA Secure UPKE Construction. Here, $\mathsf{pp} = (p, \kappa, \lambda, \bar{s}, f, g_p, \widehat{\mathbb{G}}, \mathbb{F}, \mathcal{D}, \mathcal{D}_p, \rho)$. Here $\mathcal{D}_p := [B]$.

### 5.1 IND-CR-CPA Secure Construction

**Construction 1.** *Consider* $\mathsf{UPKE}_{IND\text{-}CR\text{-}CPA} = (\mathsf{U\text{-}Setup}, \mathsf{U\text{-}Gen}, \mathsf{U\text{-}Enc}, \mathsf{U\text{-}Dec}, \mathsf{Upd\text{-}Pk}, \mathsf{Upd\text{-}Sk})$ *with the algorithms defined in Figure 1.*

**Theorem 3.** *If* $\mathcal{E} = (\mathsf{U\text{-}Setup}, \mathsf{U\text{-}Gen}, \mathsf{U\text{-}Enc}, \mathsf{U\text{-}Dec})$ *is CS+LR Secure, then Construction 1 is IND-CR-CPA Secure UPKE.*

**Proof Sketch.** We defer the complete proofs to the full version of the paper.

Let $\mathcal{A}$ be an adversary that can win, with non-negligible advantage, against the IND-CR-CPA challenger. Then, we use $\mathcal{A}$ to build an adversary $\mathcal{B}$ that can win the CS+LR security game. Implicitly, $\mathcal{B}$ sets the $\delta^*$ chosen by the CS+LR challenger as the value for the eventual update that $\mathcal{A}$ expects. Meanwhile, $\mathcal{B}$ buffers all the updates it receives from $\mathcal{A}$. $\mathcal{B}$ also uses $\mathcal{A}$'s challenge messages as its own. However, note that $\mathcal{B}$ receives an encryption of the challenge message, under the first public key $\mathsf{pk}_0$. Therefore, it relies on the key homomorphism property of the PKE scheme to transform the ciphertext so that it is consistent with the updated key. To do this, it simply uses $\mathcal{A}$'s inputs to $\mathcal{O}_U$. It does a similar argument to generate the final update ciphertext, using the encryption of $\delta^*$ (under $\mathsf{pk}_0$) that $\mathcal{B}$ receives from the CS+LR challenger. We present the reduction in Figure 2.

### 5.2 Proofs of Plaintext Equality

We rely on the classical Naor–Yung transformation [30] to upgrade IND-CPA security to IND-CCA security. While achieving IND-CCA2 typically requires simulation-sound proofs, our setting necessitates additional care due to the unknown group order. We therefore parameterize the challenge space size by $C = |\mathfrak{C}|$ and control soundness via parameter selection rather than repetition.

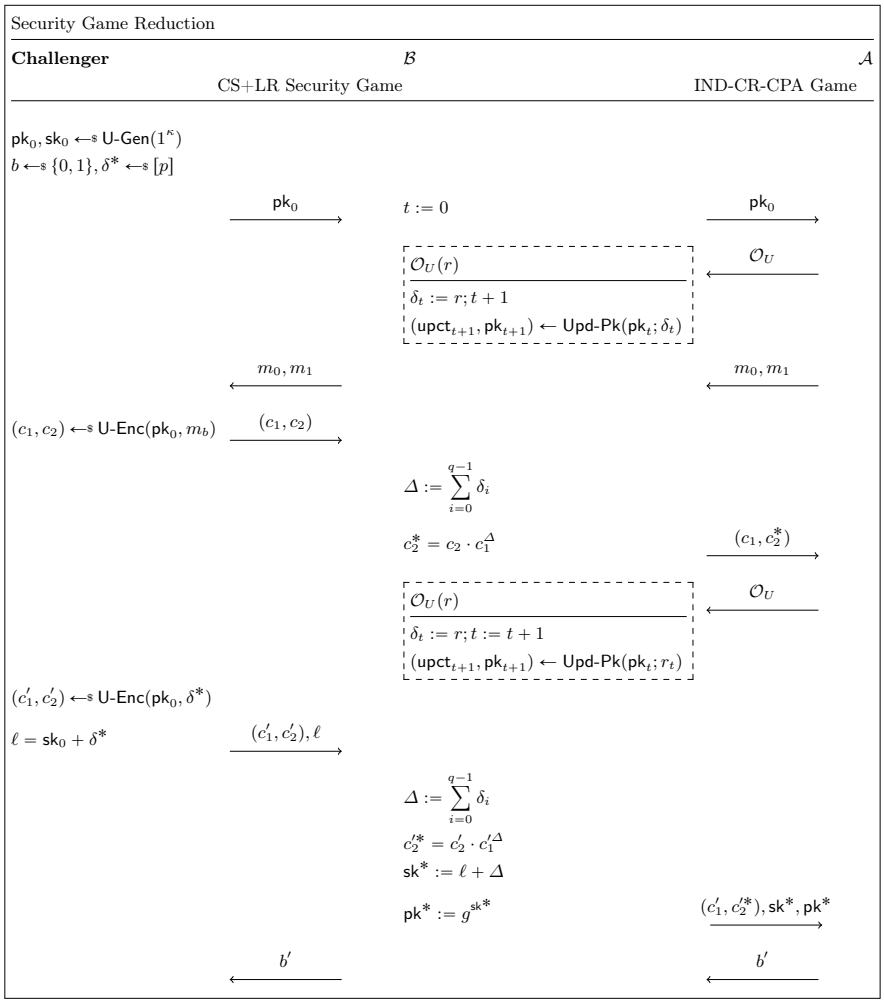


Fig. 2: The security game reduction, reducing the IND-CR-CPA security to the CS+LR security.

Specifically, secret keys and exponents are sampled from $\mathcal{D}_p = [B]$ with $B = \bar{s} \cdot 2^{40}$. To apply Theorem 1, we define $A = 2^{40} \cdot B \cdot C$, ensuring that $CB/A$ is negligible; $A$ is fixed during setup. The setup is further modified by defining the generator $g_p$ as a random exponentiation of the generator $\widetilde{g_p}$ output by $\mathsf{CL\ Gen}$.

**Construction 2 (Interactive Proof of Plaintext Equality).** *We define an interactive proof for the relation $\mathcal{R}_{\mathsf{NY}}$ consisting of*

$$\mathsf{pk}_C, \mathsf{pk}_D, C = (C_0, C_1), D = (D_0, D_1) \in \widehat{\mathbb{G}}$$

*such that there exist* $r_C, r_D \in \mathbb{Z}$ *and* $m \in [p]$ *with*

$$C_0 = g_p^{r_C}, \quad C_1 = \mathsf{pk}_C^{r_C} \cdot f^m, \qquad D_0 = g_p^{r_D}, \quad D_1 = \mathsf{pk}_D^{r_D} \cdot f^m.$$

- ***Setup*** $\mathsf{Setup}_c(1^\kappa)$*: Run* $\mathsf{pp} \leftarrow\!\!\$\ \mathsf{CLGen}(1^\kappa, 1^{40}, p; \rho)$*, where* $\mathcal{D}_p$ *is uniform over* $[B]$ *with* $B = \bar{s} \cdot 2^{40}$*. Sample* $\tau \leftarrow\!\!\$\ [B]$ *and set* $g_p = \widetilde{g}_p^{\tau}$*. Let* $A := 2^{40} \cdot B \cdot C$ *and output* $\mathsf{crs} = (\mathsf{pp}, A, B, C)$*.*
- ***Commit*** $\mathcal{P}(\mathsf{crs}; r_C, r_D, m)$*: Sample* $k_C, k_D \leftarrow\!\!\$\ [A]$ *and* $k_m \leftarrow\!\!\$\ [p]$*, and compute*

$$C_0' = g_p^{k_C}, \quad C_1' = \mathsf{pk}_C^{k_C} \cdot f^{k_m}, \qquad D_0' = g_p^{k_D}, \quad D_1' = \mathsf{pk}_D^{k_D} \cdot f^{k_m}.$$

  *Send* $(C_0', C_1', D_0', D_1')$ *to* $\mathcal{V}$*.*
- ***Challenge*** $\mathcal{V}$*: Sample* $c \leftarrow\!\!\$\ [C]$*.*
- ***Response*** $\mathcal{P}$*: Compute:* $s_C = k_C + cr_C, \qquad s_D = k_D + cr_D, \qquad s_m = k_m + cm \bmod p$*, and send* $(s_C, s_D, s_m)$*.*
- ***Verify*** $\mathcal{V}$*: Check* $s_C, s_D \in [BC(2^{40}+1)]$ *and* $s_m \in [p]$*, and verify*

$$C_0' \stackrel{?}{=} C_0^{-c} \cdot g_p^{s_C}, \quad D_0' \stackrel{?}{=} D_0^{-c} \cdot g_p^{s_D}, \quad C_1' \stackrel{?}{=} C_1^{-c} \cdot \mathsf{pk}_C^{s_C} \cdot f^{s_m}, \quad D_1' \stackrel{?}{=} D_1^{-c} \cdot \mathsf{pk}_D^{s_D} \cdot f^{s_m}.$$

We can apply Theorem 1 to get the following corollary:

**Corollary 1.** *Assuming that* $\mathtt{RO}_{\mathfrak{C}}$ *assumption is true, Construction 2 is sound with soundness error* $1/C + \mathsf{negl}(\kappa)$*. Furthermore, the construction achieves completeness and has statistical special honest-verifier zero-knowledge because* $m \in [p], r_C, r_D \in [B]$*,* $C \cdot B/A$ *is negligible (by choice of A), and* $s_m \in [p]$*.*

**Lemma 3.** *Under* $\mathtt{SR}$ *assumption, Construction 2 has quasi-unique responses.*

*Proof.* Let us assume that it does not have quasi-unique responses. Then, an adversary can find two valid transcripts: $(C_0', C_1', D_0', D_1', c, s_m, s_C, s_D)$ and $(C_0', C_1', D_0', D_1', c, s_m', s_C', s_D')$ for some statement $(\mathsf{pk}, h_p, C_0, C_1, D_0, D_1)$ where $(s_m, s_C, s_D) \neq (s_m', s_C', s_D')$.

Let us assume that $s_C \neq s_C'$. Then, from the verification equation for $C_0$, we get that: $C_0' = C_0^{-c} \cdot g_p^{s_C} = C_0^{-c} \cdot g_p^{-s_C'}$. This gives us $g_p^{s_C - s_C'} = 1$. Indeed, it follows that $g_p^{2(s_C - s_C')} = 1$ In other words, $g_p^{2(s_C - s_C')+1} = g_p$. Therefore, for $X = g_p$ and $e = 2 \cdot |s_C - s_C'| + 1$, we have $Y = g_p = X^e$. Note that $2|s_C - s_C'|$ is always even, therefore $e$ is odd and is therefore not a power of 2. One can perform a similar analysis if $s_D \neq s_D'$.

If $s_m \neq s_m'$ while $s_C = s_C'$, then we get from the verification for $C_1'$ that $f^{s_m - s_m'} = 1$. However, note that the order of this subgroup is known, which is $p$. So, we get $s_m - s_m' = 0 \mod p$. Recall that $s_m, s_m'$ are such that $0 < s_m, s_m' < p$. It follows that it is impossible to have $s_m \neq s_m'$ while satisfying $s_m - s_m' = 0 \mod p$ and $0 < s_m, s_m' < p$.

### 5.3 Well-Formedness of Update Ciphertext

As before, let $C = |\mathcal{C}|$ denote the size of the challenge space. We compute the values $A$ and $B$ analogously to the previous section and again use a random exponent of the generator output by $\mathsf{CLGen}$ as the group generator. Finally, we set $A' = 2^{40} \cdot C \cdot p$ to ensure that the value $u_2$ is generated with $r_2 \leftarrow A'$, so that $C \cdot p/A'$ remains negligible.

**Construction 3 (IP of Well-Formedness of Update Ciphertext).** *We define an interactive proof for*

$$\mathcal{R}_{\mathsf{upct}} = \{\, \mathsf{pk}, \mathsf{pk}', \mathsf{upct}; \delta, t \mid \mathsf{pk}' = \mathsf{pk} \cdot g_p^{\delta},\ \mathsf{upct} = (g_p^t, \mathsf{pk}^t \cdot f^{\delta}),\ \mathsf{pk}, \mathsf{pk}', \mathsf{upct} \in \widehat{\mathbb{G}},\ \delta, t \in \mathbb{Z} \,\}.$$

- ***Setup*** $\mathsf{Setup}_{\mathsf{upct}}(1^\kappa)$*: Run* $\mathsf{pp} \leftarrow\!\!\$\ \mathsf{CLGen}(1^\kappa, 1^{40}, p; \rho)$ *with* $\mathcal{D}_p$ *uniform over* $[B]$, $B = \bar{s} \cdot 2^{40}$. *Sample* $\tau \leftarrow\!\!\$\ [B]$, *set* $g_p = \widetilde{g_p}^{\tau}$, *and compute* $A := 2^{40} \cdot B \cdot C$, $A' := 2^{40} \cdot C \cdot p$. *Output* $\mathsf{crs} = (\mathsf{pp}, A, A', B, C)$.
- ***Commit*** $\mathcal{P}(\mathsf{crs}, (\mathsf{pk}, \mathsf{pk}', \mathsf{upct}), (\delta, t))$*: Sample* $r_1 \leftarrow\!\!\$\ [A], r_2 \leftarrow\!\!\$\ [A']$ *and compute*

$$t_1 = g_p^{r_1}, \quad t_2 = \mathsf{pk}^{r_1} \cdot f^{r_2 \bmod p}, \quad t' = g_p^{r_2}.$$

  *Send* $(t_1, t_2, t')$ *to* $\mathcal{V}$.
- ***Challenge*** $\mathcal{V}$*: Sample* $k \leftarrow\!\!\$\ [C]$.
- ***Response*** $\mathcal{P}$*: Compute:* $u_1 = r_1 + k \cdot t, \quad u_2 = r_2 + k \cdot \delta$ *and send* $(u_1, u_2)$.
- ***Verify*** $\mathcal{V}$*: Check*

$$\mathsf{upct} \in \widehat{\mathbb{G}}^2,\ \mathsf{pk}, \mathsf{pk}' \in \widehat{\mathbb{G}}, \quad u_1 \in [BC(2^{40}+1)],\ u_2 \in [C \cdot p \cdot (2^{40}+1)],$$

  *and verify*

$$t_1 \stackrel{?}{=} C_0^{-k} \cdot g_p^{u_1}, \quad t_2 \stackrel{?}{=} C_1^{-k} \cdot \mathsf{pk}^{u_1} \cdot f^{u_2 \bmod p}, \quad t' \stackrel{?}{=} g_p^{u_2} \cdot (\mathsf{pk}'/\mathsf{pk})^{-k}.$$

We can apply Theorem 1 [4] to get the following corollary:

**Corollary 2.** *Construction 3 is sound with soundness error* $1/C + \mathsf{negl}(\kappa)$. *Further, note that* $\delta \in [p], t \in [B]$ *is complete if* $\delta \in [p], t \in [B]$, *and by choice of* $A, A'$ $C \cdot B/A$ *and* $C \cdot p/A'$ *are negligible, the protocol also achieves completeness and has statistical special honest-verifier zero-knowledge.*

In the full version of the paper, we further augment this to build a verifiable encryption scheme to replace the one from [19].

[4] Our security proofs does not require witness extraction but only soundness. Therefore, we use the results from Braun *et al.* [10] rather than using the proof from Castagnos *et al.* [15]. The computational soundness unfortunately means that we cannot create an efficient IND-CU-CPA scheme. Further, we do not need the property of simulation-soundness.

Table 4: Running times (ms) of our IND-CU-CCA construction across security levels. Prior works do not report timings for IND-CU-CCA protocols. Note that the encryption and decryption times are the same as CR-CCA constructions.

| **Operation** | **112 Bits** | **128 Bits** | **192 Bits** | **256 Bits** |
|---|---|---|---|---|
| Upd-Pk | 98.7 | 171.1 | 570.9 | 1544.0 |
| Upd-Sk | 94.6 | 152.5 | 461.7 | 1142.9 |

Table 5: Comparison of running times (ms) of our constructions vs. Haidar *et al.*[24] for IND-CR-CPA and IND-CR-CCA security. For CR-CPA, we show encryption, decryption, and update times; for CR-CCA, encryption and decryption. Security levels: 112, 128, 192, and 256 bits.

| Security | Operation | 112 Bits | | 128 Bits | | 192 Bits | | 256 Bits | |
|---|---|---|---|---|---|---|---|---|---|
| | | Ours | HLP22 [24] | Ours | HLP22 [24] | Ours | HLP22 [24] | Ours | HLP22 [24] |
| CR-CPA | U-Enc | 7.1 | **5.7** | **13.3** | 16.5 | **47.4** | 166.5 | **136.6** | 920.7 |
| | Upd-Pk | 23.3 | **11.3** | 43.1 | **33.1** | **153.5** | 330.1 | **414.4** | 1830.6 |
| | Upd-Sk | 15.8 | **12.5** | **29.8** | 36.4 | **114.1** | 364.3 | **334.5** | 1999.9 |
| CR-CCA | U-Enc | **37.0** | 66.4 | **64.2** | 187.7 | **215.4** | 1825.6 | **587.8** | 9815.1 |
| | U-Dec | **51.9** | 69.9 | **86.7** | 200.7 | **271.6** | 1823.1 | **724.3** | 10097.6 |

## 6 Implementation and Benchmarks

We implemented our constructions in C/C++ as a proof-of-concept on top of the BICYCL library [9], which in turn uses GMP. The implementation is artifact-attached, publicly available, and unoptimized (no parallelization). Experiments were run on an Apple M1 Pro (16 GB, macOS 13) using CMake 3.25.2 and AppleClang 14.0. Reported times (ms) are averages over 1000 trials.

**Performance Comparison.** Our constructions outperform prior work [24] across all relevant operations and security levels. For IND-CU-CCA, encryption and decryption are faster than even HLP22's CR-CCA counterparts at 128-bit security, and key-update operations (Upd-Pk and Upd-Sk) are at most as costly as CR-CCA encryption/decryption, with Upd-Pk and Upd-Sk actually faster than HLP22's respective operations. Similarly, our IND-CR-CPA algorithms consistently outperform the DCR-based scheme of [24], except at 112-bit security and Upd-Pk at 128 bits. IND-CR-CCA encryption and decryption are also faster at all levels. Table 5 summarizes these comparisons, while Table 4 reports running times of our IND-CU-CCA construction, adapted from the Camenisch–Shoup class [12,29].

**Disclaimer.** This paper was prepared for informational purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co and its affiliates ("J.P. Morgan") and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment ad-

vice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation would be unlawful.

## References


1. Alwen, J., Coretti, S., Dodis, Y., Tselekounis, Y.: Security analysis and improvements for the IETF MLS standard for group messaging. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part I. LNCS, vol. 12170, pp. 248–277. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2020). https://doi.org/10.1007/978-3-030-56784-2_9
2. Alwen, J., Coretti, S., Dodis, Y., Tselekounis, Y.: Modular design of secure group messaging protocols and the security of MLS. In: Vigna, G., Shi, E. (eds.) ACM CCS 2021. pp. 1463–1483. ACM Press, Virtual Event, Republic of Korea (Nov 15–19, 2021). https://doi.org/10.1145/3460120.3484820
3. Alwen, J., Jost, D., Mularczyk, M.: On the insider security of MLS. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022, Part II. LNCS, vol. 13508, pp. 34–68. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 15–18, 2022). https://doi.org/10.1007/978-3-031-15979-4_2
4. Asano, K., Watanabe, Y.: Updatable public key encryption with strong cca security: Security analysis and efficient generic construction. Cryptology ePrint Archive, Paper 2023/976 (2023), https://eprint.iacr.org/2023/976, https://eprint.iacr.org/2023/976
5. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg, Germany, Cambridge, UK (Apr 15–19, 2012). https://doi.org/10.1007/978-3-642-29011-4_29
6. Barnes, R., Beurdouche, B., Robert, R., Millican, J., Omara, E., Cohn-Gordon, K.: The Messaging Layer Security (MLS) Protocol. Tech. Rep. 9420 (Jul 2023). https://doi.org/10.17487/RFC9420, https://www.rfc-editor.org/info/rfc9420
7. Beurdouche, B., Rescorla, E., Omara, E., Inguva, S., Duric, A.: The Messaging Layer Security (MLS) Architecture. Internet-Draft draft-ietf-mls-architecture-11, Internet Engineering Task Force (Jul 2023), https://datatracker.ietf.org/doc/draft-ietf-mls-architecture/11/, work in Progress
8. Biasse, J.F., Jacobson, M.J., Silvester, A.K.: Security estimates for quadratic field based cryptosystems. In: Steinfeld, R., Hawkes, P. (eds.) ACISP 10. LNCS, vol. 6168, pp. 233–247. Springer, Heidelberg, Germany, Sydney, NSW, Australia (Jul 5–7, 2010)
9. Bouvier, C., Castagnos, G., Imbert, L., Laguillaumie, F.: I want to ride my BICYCL : BICYCL implements CryptographY in CLass groups. Journal of Cryptology **36**(3), 17 (Jul 2023). https://doi.org/10.1007/s00145-023-09459-1
10. Braun, L., Damgård, I., Orlandi, C.: Secure multiparty computation from threshold encryption based on class groups. In: Handschuh, H., Lysyanskaya, A. (eds.) Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023,

Proceedings, Part I. Lecture Notes in Computer Science, vol. 14081, pp. 613–645. Springer (2023). https://doi.org/10.1007/978-3-031-38557-5_20, https://doi.org/10.1007/978-3-031-38557-5_20
11. Camenisch, J., Damgård, I.: Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 331–345. Springer, Heidelberg, Germany, Kyoto, Japan (Dec 3–7, 2000). https://doi.org/10.1007/3-540-44448-3_25
12. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2003). https://doi.org/10.1007/978-3-540-45146-4_8
13. Canetti, R., Halevi, S., Katz, J.: A forward-secure public-key encryption scheme. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 255–271. Springer, Heidelberg, Germany, Warsaw, Poland (May 4–8, 2003). https://doi.org/10.1007/3-540-39200-9_16
14. Castagnos, G., Catalano, D., Laguillaumie, F., Savasta, F., Tucker, I.: Two-party ECDSA from hash proof systems and efficient instantiations. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019, Part III. LNCS, vol. 11694, pp. 191–221. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2019). https://doi.org/10.1007/978-3-030-26954-8_7
15. Castagnos, G., Catalano, D., Laguillaumie, F., Savasta, F., Tucker, I.: Bandwidth-efficient threshold EC-DSA. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part II. LNCS, vol. 12111, pp. 266–296. Springer, Heidelberg, Germany, Edinburgh, UK (May 4–7, 2020). https://doi.org/10.1007/978-3-030-45388-6_10
16. Castagnos, G., Imbert, L., Laguillaumie, F.: Encryption switching protocols revisited: Switching modulo p. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part I. LNCS, vol. 10401, pp. 255–287. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017). https://doi.org/10.1007/978-3-319-63688-7_9
17. Castagnos, G., Laguillaumie, F.: Linearly homomorphic encryption from DDH. In: Nyberg, K. (ed.) CT-RSA 2015. LNCS, vol. 9048, pp. 487–505. Springer, Heidelberg, Germany, San Francisco, CA, USA (Apr 20–24, 2015). https://doi.org/10.1007/978-3-319-16715-2_26
18. Castagnos, G., Laguillaumie, F., Tucker, I.: Practical fully secure unrestricted inner product functional encryption modulo p. In: Peyrin, T., Galbraith, S. (eds.) ASIACRYPT 2018, Part II. LNCS, vol. 11273, pp. 733–764. Springer, Heidelberg, Germany, Brisbane, Queensland, Australia (Dec 2–6, 2018). https://doi.org/10.1007/978-3-030-03329-3_25
19. Devigne, J., Duguey, C., Fouque, P.A.: MLS group messaging: How zero-knowledge can secure updates. In: Bertino, E., Shulman, H., Waidner, M. (eds.) ESORICS 2021, Part II. LNCS, vol. 12973, pp. 587–607. Springer, Heidelberg, Germany, Darmstadt, Germany (Oct 4–8, 2021). https://doi.org/10.1007/978-3-030-88428-4_29
20. Dodis, Y., Karthikeyan, H., Wichs, D.: Updatable public key encryption in the standard model. In: Nissim, K., Waters, B. (eds.) TCC 2021, Part III. LNCS, vol. 13044, pp. 254–285. Springer, Heidelberg, Germany, Raleigh, NC, USA (Nov 8–11, 2021). https://doi.org/10.1007/978-3-030-90456-2_9
21. Fujisaki, E., Okamoto, T.: Secure integration of asymmetric and symmetric encryption schemes. Journal of Cryptology **26**(1), 80–101 (Jan 2013). https://doi.org/10.1007/s00145-011-9114-1

22. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg, Germany, Queenstown, New Zealand (Dec 1–5, 2002). https://doi.org/10.1007/3-540-36178-2_34
23. Guo, Y., Karthikeyan, H., Polychroniadou, A.: Pride ct: Towards public consensus, private transactions, and forward secrecy in decentralized payments. In: 2024 IEEE Symposium on Security and Privacy (SP). IEEE Computer Society, Los Alamitos, CA, USA (may 2024)
24. Haidar, C.A., Libert, B., Passelègue, A.: Updatable public key encryption from DCR: Efficient constructions with stronger security. In: Yin, H., Stavrou, A., Cremers, C., Shi, E. (eds.) ACM CCS 2022. pp. 11–22. ACM Press, Los Angeles, CA, USA (Nov 7–11, 2022). https://doi.org/10.1145/3548606.3559376
25. Haidar, C.A., Passelègue, A., Stehlé, D.: Efficient updatable public-key encryption from lattices. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology - ASIACRYPT 2023 - 29th International Conference on the Theory and Application of Cryptology and Information Security, Guangzhou, China, December 4-8, 2023, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14442, pp. 342–373. Springer (2023). https://doi.org/10.1007/978-981-99-8733-7_11, https://doi.org/10.1007/978-981-99-8733-7_11
26. Horwitz, J., Lynn, B.: Toward hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg, Germany, Amsterdam, The Netherlands (Apr 28 – May 2, 2002). https://doi.org/10.1007/3-540-46035-7_31
27. Jost, D., Maurer, U., Mularczyk, M.: Efficient ratcheting: Almost-optimal guarantees for secure messaging. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 159–188. Springer, Heidelberg, Germany, Darmstadt, Germany (May 19–23, 2019). https://doi.org/10.1007/978-3-030-17653-2_6
28. Marlinspike, M., Perrin, T.: The double ratchet algorithm. Tech. rep. (2016), https://whispersystems.org/docs/specifications/doubleratchet/doubleratchet.pdf
29. Miao, P., Patel, S., Raykova, M., Seth, K., Yung, M.: Two-sided malicious security for private intersection-sum with cardinality. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 3–33. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2020). https://doi.org/10.1007/978-3-030-56877-1_1
30. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC. pp. 427–437. ACM Press, Baltimore, MD, USA (May 14–16, 1990). https://doi.org/10.1145/100216.100273
31. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT'99. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg, Germany, Prague, Czech Republic (May 2–6, 1999). https://doi.org/10.1007/3-540-48910-X_16
32. Tucker, I.: Functional encryption and distributed signatures based on projective hash functions, the benefit of class groups. Theses, Université de Lyon (Oct 2020), https://theses.hal.science/tel-03021689