

Homomorphic Secret-Sharing with Certified Deletion

Nikhil Pappu[†]

[†]Portland State University
nikpappu@pdx.edu

October 14, 2024

Abstract

Contents

1	Preliminaries	3
2	HSS with Certified Deletion	3
2.1	HSS-CD Syntax	3
2.2	Additive HSS-CD Syntax	4
2.3	Security Definitions	4
3	Feasibility Results	5
3.1	FHE-CD based Construction	5
3.2	Impossibility Results	8

1 Preliminaries

2 HSS with Certified Deletion

Unless otherwise specified, we will consider the following kind of HSS schemes by default:

- Those that work for all PPT computable circuits.
- Are 2-out-of-2 secret-sharing schemes.
- Allow evaluation for a single secret.

An HSS scheme with certified deletion must have the following syntax and correctness requirements:

2.1 HSS-CD Syntax

A scheme satisfying the HSS-CD syntax is a tuple of 5 algorithms $\text{HSS-CD} = \text{HSS-CD}(\text{Share}, \text{Eval}, \text{Del}, \text{Vrfy}, \text{Rec})$ with the following properties:

Syntax:

$\text{Share}(s) \rightarrow (sh_0^0, vk_0), (sh_1^0, vk_1)$: The sharing algorithm outputs quantum (possibly-entangled) secret-shares sh_0^0, sh_1^0 encoding an input secret s . It also outputs the corresponding classical verification keys vk_0, vk_1 .

$\text{Eval}(C_j, i, sh_i^{j-1}) \rightarrow sh_i^j$: The evaluation algorithm takes the description of a PPT computable circuit C_j , an index $i \in \{0, 1\}$, and an input share sh_i^{j-1} . It outputs a possibly-altered output share sh_i^j .

$\text{Del}(i, sh_i^j) \rightarrow \text{cert}_i$: The deletion algorithm takes an index $i \in \{0, 1\}$, a corresponding share sh_i^j , and produces a deletion certificate cert_i .

$\text{Vrfy}(i, vk_i, \text{cert}_i) \rightarrow \top / \perp$: The verification algorithm takes an index $i \in \{0, 1\}$, the corresponding verification key vk_i and a certificate cert_i . It outputs \top or \perp .

$\text{Rec}(sh_0^q, sh_1^q) \rightarrow (d_1, \dots, d_q)$: The reconstruction algorithm takes two evaluated input shares sh_0^q, sh_1^q and outputs a q -tuple (d_1, \dots, d_q) .

Evaluation Correctness: \forall PPT C , the following condition holds for all $q = \text{poly}(\lambda)$:

$$\Pr \left[(d_1, \dots, d_q) = (C_1(s), \dots, C_q(s)) : \begin{array}{l} (sh_0^0, vk_0), (sh_1^0, vk_1) \leftarrow \text{Share}(s) \\ \forall i, j \in \{0, 1\} \times [q] : sh_i^j \leftarrow \text{Eval}(C_j, i, sh_i^{j-1}) \\ (d_1, \dots, d_q) \leftarrow \text{Rec}(sh_0^q, sh_1^q) \end{array} \right] \geq 1 - \text{negl}(\lambda)$$

Deletion Correctness: The following condition holds for all $i \in \{0, 1\}$ and $q = \text{poly}(\lambda)$:

$$\Pr \left[\text{Vrfy}(i, vk_i, \text{cert}_i) \rightarrow \top : \begin{array}{l} (sh_0^0, vk_0), (sh_1^0, vk_1) \leftarrow \text{Share}(s) \\ \forall j \in [q] : sh_i^j \leftarrow \text{Eval}(C_j, i, sh_i^{j-1}) \\ \text{cert}_i \leftarrow \text{Del}(i, sh_i^q) \end{array} \right] \geq 1 - \text{negl}(\lambda)$$

Compactness: The following condition holds for all $i \in \{0, 1\}$ and $q = \text{poly}(\lambda)$, where l_q denotes the output length of the circuit C_q :

$$\left[|sh_i^q| - |sh_i^{q-1}| = \text{poly}(1^\lambda, l_q) : \begin{array}{l} (sh_0^0, vk_0), (sh_1^0, vk_1) \leftarrow \text{Share}(s) \\ \forall j \in [q] : sh_i^j \leftarrow \text{Eval}(C_j, i, sh_i^{j-1}) \end{array} \right]$$

2.2 Additive HSS-CD Syntax

A scheme satisfying the additive HSS-CD syntax is a tuple of 4 algorithms $\text{HSS-CD} = \text{HSS-CD}(\text{Share}, \text{Eval}, \text{Del}, \text{Vrfy})$ with the following properties:

Syntax:

$\text{Share}(s) \rightarrow (sh_0^0, vk_0), (sh_1^0, vk_1)$: The sharing algorithm outputs quantum (possibly-entangled) secret-shares sh_0^0, sh_1^0 encoding an input secret s . It also outputs the corresponding classical verification keys vk_0, vk_1 .

$\text{Eval}(C_j, i, (sh_i^{j-1}, sh_i^{j-1})) \rightarrow (sh_i^j, sh_i^j)$: The evaluation algorithm takes the description of a PPT computable circuit C_j , an index $i \in \{0, 1\}$, a quantum input share sh_i^{j-1} , and a classical input share sh_i^{j-1} where $sh_i^0 = \perp$. It outputs a quantum output share sh_i^j and a classical output share sh_i^j .

$\text{Del}(i, sh_i^j) \rightarrow \text{cert}_i$: The deletion algorithm takes an index $i \in \{0, 1\}$, a corresponding quantum share sh_i^j , and produces a deletion certificate cert_i .

$\text{Vrfy}(i, vk_i, \text{cert}_i) \rightarrow \top / \perp$: The verification algorithm takes an index $i \in \{0, 1\}$, the corresponding verification key vk_i and a certificate cert_i . It outputs \top or \perp .

Evaluation Correctness: \forall PPT C , the following condition holds for all $q = \text{poly}(\lambda)$:

$$\Pr \left[sh_0^q \oplus sh_1^q = C_1(s) \parallel \dots \parallel C_q(s) : \begin{array}{l} (sh_0^0, vk_0), (sh_1^0, vk_1) \leftarrow \text{Share}(s) \\ \forall i, j \in \{0, 1\} \times [q] : (sh_i^j, sh_i^j) \leftarrow \text{Eval}(C_j, i, (sh_i^{j-1}, sh_i^{j-1})) \end{array} \right] \geq 1 - \text{negl}(\lambda)$$

In the case of additive HSS-CD, we will consider the following deletion guarantee by default, which is weaker than the standard deletion correctness guarantee:

Delete-before-Eval Correctness: The following condition holds for all $i \in \{0, 1\}$:

$$\Pr \left[\text{Vrfy}(i, vk_i, \text{cert}_i) \rightarrow \top : \begin{array}{l} (sh_0^0, vk_0), (sh_1^0, vk_1) \leftarrow \text{Share}(s) \\ \text{cert}_i \leftarrow \text{Del}(i, sh_i^0) \end{array} \right] \geq 1 - \text{negl}(\lambda)$$

Deletion Correctness (Optional): The following condition holds for all $i \in \{0, 1\}$ and $q = \text{poly}(\lambda)$:

$$\Pr \left[\text{Vrfy}(i, vk_i, \text{cert}_i) \rightarrow \top : \begin{array}{l} (sh_0^0, vk_0), (sh_1^0, vk_1) \leftarrow \text{Share}(s) \\ \forall j \in [q] : (sh_i^j, sh_i^j) \leftarrow \text{Eval}(C_j, i, (sh_i^{j-1}, sh_i^{j-1})) \\ \text{cert}_i \leftarrow \text{Del}(i, sh_i^q) \end{array} \right] \geq 1 - \text{negl}(\lambda)$$

2.3 Security Definitions

Statistical/Computational Deletion Security wrt Share j : The following security notion is defined wrt a non-local quantum adversary $(\mathcal{A}_0, \mathcal{A}_1)$:

$\text{Expt}_{\text{HSS-CD}, (\mathcal{A}_0, \mathcal{A}_1)}^{\text{del}}(1^\lambda, j, b)$:

1. \mathcal{A}_0 sends $(s_0, s_1) \in \{0, 1\}^\lambda$ to the challenger.
2. The challenger runs $(sh_0^0, vk_0), (sh_1^0, vk_1) \leftarrow \text{Share}(s_b)$ and sends each sh_i^0 to party P_i .
3. \mathcal{A}_j sends (cert_j, R_j) and \mathcal{A}_{1-j} sends R_{1-j} where R_0, R_1 are some registers.
4. If $\text{Vrfy}(j, vk_j, \text{cert}_j) = \top$, then output (R_1, R_2) .

Statistical Deletion Security wrt Share j holds if the following holds:

$$TD\left(\text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del}}(1^\lambda, j, 0), \text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del}}(1^\lambda, j, 1)\right) \leq \text{negl}(\lambda)$$

Computational Deletion Security wrt Share j holds if the following holds for all QPT \mathcal{A} :

$$\left| \Pr \left[\mathcal{A} \left(\text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del}}(1^\lambda, j, 0) \right) = 1 \right] - \Pr \left[\mathcal{A} \left(\text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del}}(1^\lambda, j, 1) \right) = 1 \right] \right| \leq \text{negl}(\lambda)$$

Statistical/Computational Double-Deletion Security: The following security notion is defined wrt a non-local quantum adversary $(\mathcal{A}_0, \mathcal{A}_1)$:

$\text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del-2}}(1^\lambda, b)$:

- 1.
2. \mathcal{A}_0 sends $(s_0, s_1) \in \{0, 1\}^\lambda$ to the challenger. The challenger runs $(sh_0^0, vk_0), (sh_1^0, vk_1) \leftarrow \text{Share}(s_b)$ and sends each sh_i^0 to party P_i .
3. \mathcal{A}_0 sends (cert_0, R_0) and \mathcal{A}_1 sends (cert_1, R_1) where R_0, R_1 are some registers.
4. If $\text{Vrfy}(0, vk_0, \text{cert}_0) = \text{Vrfy}(1, vk_1, \text{cert}_1) = \top$, then output (R_0, R_1) .

Statistical Double-Deletion Security holds if the following holds:

$$TD\left(\text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del-2}}(1^\lambda, 0), \text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del-2}}(1^\lambda, 1)\right) \leq \text{negl}(\lambda)$$

Computational Double-Deletion Security holds if the following holds for all QPT \mathcal{A} :

$$\left| \Pr \left[\mathcal{A} \left(\text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del-2}}(1^\lambda, 0) \right) = 1 \right] - \Pr \left[\mathcal{A} \left(\text{Exp}_{\text{HSS-CD},(\mathcal{A}_0,\mathcal{A}_1)}^{\text{del-2}}(1^\lambda, 1) \right) = 1 \right] \right| \leq \text{negl}(\lambda)$$

Hereafter, we will use *stat* to denote statistical security and *comp* to denote computational security.

Definition 2.1 ((Additive) (X, Y)-HSS-CD scheme). An (Additive) (X, Y) -HSS-CD scheme for $X, Y \in \{\text{stat}, \text{comp}\}^2$ is a scheme that satisfies the (Additive) HSS-CD syntax, the X deletion security for share 0, and the Y deletion security for share 1.

Definition 2.2 ((Additive) (X)-HSS-CD scheme). An (Additive) (X) -HSS-CD scheme for $X \in \{\text{stat}, \text{comp}\}$ is a scheme satisfying the (Additive) HSS-CD syntax and the X double-deletion security.

Remark 2.3. Observe that a (stat, comp)-HSS-CD scheme is also a (stat)-HSS-CD scheme. Likewise, a (comp, comp)-HSS-CD scheme is also a (comp)-HSS-CD scheme.

3 Feasibility Results

3.1 FHE-CD based Construction

We construct a (stat, comp)-HSS-CD scheme $\text{HSS-CD} = \text{HSS-CD}(\text{Share}, \text{Eval}, \text{Del}, \text{Vrfy}, \text{Rec})$ using the following building blocks.

- Fully Homomorphic Encryption with Certified Deletion (FHE-CD) scheme $\text{FHE-CD} = \text{FHE-CD}(\text{Setup}, \text{Enc}, \text{Dec}, \text{Eval}, \text{Del}, \text{Vrfy})$.

- Secret Sharing with Certified Deletion (SS-CD) scheme $\text{SS-CD} = \text{SS-CD}(\text{Share}, \text{Rec}, \text{Del}, \text{Vrfy})$.

The construction is as follows.

$\text{HSS-CD.Share}(s)$:

1. Generate $(\text{pk}, \text{sk}) \leftarrow \text{FHE-CD.Setup}(1^\lambda)$.
2. Compute $(\text{fhecd.ct}^0, \text{fhecd.vk}) \leftarrow \text{FHE-CD.Enc}(s)$.
3. Compute $(\text{sscd.sh}, \text{sscd.csh}), \text{sscd.vk} \leftarrow \text{SS-CD.Share}(\text{sk})$.
4. Set $\text{sh}_0^0 := (\text{fhecd.pk}, \text{fhecd.ct}^0, \text{sscd.csh})$ and $\text{vk}_0 := \text{fhecd.vk}$.
5. Set $\text{sh}_1^0 := \text{sscd.sh}$ and $\text{vk}_1 := \text{sscd.vk}$.
6. Output $(\text{sh}_0^0, \text{vk}_0), (\text{sh}_1^0, \text{vk}_1)$.

$\text{HSS-CD.Eval}(C_j, i, \text{sh}_i^{j-1})$: If $i = 1$, set $\text{sh}_1^j := \text{sh}_1^{j-1}$. Else, execute the following:

1. Parse sh_0^{j-1} as $(\text{fhecd.pk}, \text{fhecd.ct}^{j-1}, \text{sscd.csh})$.
2. Compute $\text{fhecd.ct}^j \leftarrow \text{FHE-CD.Eval}(\text{fhecd.pk}, C_j, \text{fhecd.ct}^{j-1})$.
3. Set $\text{sh}_0^j := (\text{fhecd.pk}, \text{fhecd.ct}^j, \text{sscd.csh})$.
4. Output sh_i^j .

$\text{HSS-CD.Del}(i, \text{sh}_i^j)$:

1. If $i = 0$, execute the following:
 - (i) Parse sh_0^j as $(\text{fhecd.pk}, \text{fhecd.ct}^j, \text{sscd.csh})$.
 - (ii) Compute and output $\text{cert}_0 \leftarrow \text{FHE-CD.Del}(\text{fhecd.ct}^j)$.
2. If $i = 1$, execute the following:
 - (i) Parse sh_1^j as sscd.sh .
 - (ii) Compute and output $\text{cert}_1 \leftarrow \text{SS-CD.Del}(\text{sscd.sh})$.

$\text{HSS-CD.Vrfy}(i, \text{vk}_i, \text{cert}_i)$:

1. If $i = 0$, output $\text{ans}_0 \leftarrow \text{FHE-CD.Vrfy}(\text{vk}_0, \text{cert}_0)$.
2. If $i = 1$, output $\text{ans}_0 \leftarrow \text{SS-CD.Vrfy}(\text{vk}_1, \text{cert}_1)$.

$\text{HSS-CD.Rec}(\text{sh}_0^q, \text{sh}_1^q)$:

1. Parse sh_0^q as $(\text{fhecd.pk}, \text{fhecd.ct}^q, \text{sscd.csh})$.
2. Parse sh_1^q as sscd.sh .
3. Compute $\text{sk} \leftarrow \text{SS-CD.Dec}(\text{sscd.sh}, \text{sscd.csh})$.
4. Compute and output $(d_1, \dots, d_q) \leftarrow \text{FHE-CD.Dec}(\text{sk}, \text{fhecd.ct}^q)$.

Theorem 3.1. *There exists a (stat, comp)-HSS-CD scheme assuming the existence of a fully homomorphic encryption scheme with certified deletion (FHE-CD), and a secret-sharing scheme with certified deletion (SS-CD).*

Proof. We will prove that the construction HSS-CD is a (stat, comp)-HSS-CD scheme. First, we will assume that $(\mathcal{A}_0, \mathcal{A}_1)$ is a non-local adversary that breaks the statistical deletion security of share 0. We will use this adversary to break the certified deletion security of the FHE-CD scheme FHE-CD. Consider a QPT reduction \mathcal{R} that runs as follows in the FHE-CD game:

Execution of $\mathcal{R}^{(\mathcal{A}_0, \mathcal{A}_1)}$ in $\text{Exp}_{\text{FHE-CD}, \mathcal{R}}^{\text{fhe-cd}}(1^\lambda, b)$:

1. \mathcal{A}_0 sends $(s_0, s_1) \in \{0, 1\}^\lambda$ to \mathcal{R} , which \mathcal{R} forwards to the challenger.
2. The challenger samples $(pk, sk) \leftarrow \text{Setup}(1^\lambda)$ and sends pk to \mathcal{R} .
3. The challenger encrypts s_b as $ct \leftarrow \text{Enc}(pk, s_b)$ and sends ct to \mathcal{R} .
4. \mathcal{R} computes $sh_0^0 := (pk, ct, \text{sscd.csh})$, where $\text{sscd.csh} \leftarrow \text{SS-CD.Sim}(1^\lambda)$.
5. \mathcal{R} runs \mathcal{A}_0 on input sh_0^0 . If \mathcal{A}_0 outputs (cert_0, R_0) , \mathcal{R} sends cert_0 to the challenger.
6. The challenger computes $\text{ans} \leftarrow \text{Vrfy}(vk, \text{cert}_0)$. If $\text{ans} = \top$, it sends sk to \mathcal{R} . Else, it outputs \perp .
7. \mathcal{R} computes sscd.sh conditioned on $(\text{sscd.sh}, \text{sscd.sh})$ encoding sk .
8. \mathcal{R} sends $sh_1^0 := \text{sscd.sh}$ to \mathcal{A}_1 . If \mathcal{A}_1 outputs R_1 , send (R_0, R_1) to the challenger.

We will now argue that if $(\mathcal{A}_0, \mathcal{A}_1)$ break statistical security wrt share 0, then \mathcal{R} breaks the certified-deletion security of FHE-CD. Observe that the view of \mathcal{A}_0 in the reduction is identically distributed to its view in $\text{Exp}_{\text{HSS-CD}, (\mathcal{A}_0, \mathcal{A}_1)}^{\text{del}}(1^\lambda, 0, b)$. Now, notice that if $\text{HSS-CD.Vrfy}(0, vk_0, \text{cert}_0)$ passes, then $\text{FHE-CD.Vrfy}(vk, \text{cert}_0)$ also passes. Consequently, \mathcal{R} receives the secret key sk . By the information-theoretic secrecy of the scheme SS-CD, the view of \mathcal{A}_1 is identically distributed to that in the original experiment. As a result, (R_0, R_1) are identically distributed to that of the HSS-CD game. By assumption, there exists an unbounded algorithm that can use (R_0, R_1) to guess b with non-negligible probability. This breaks the certified-deletion security of FHE-CD.

Next, we will assume that $(\mathcal{A}_0, \mathcal{A}_1)$ is a non-local adversary that breaks the computational deletion security of share 1. We will use this adversary to break the certified deletion security of the SS-CD scheme SS-CD. Consider a non-local reduction $(\mathcal{R}_0, \mathcal{R}_1)$ that runs as follows:

Execution of $(\mathcal{R}_0, \mathcal{R}_1)^{(\mathcal{A}_0, \mathcal{A}_1)}$ in $\text{Exp}_{\text{SS-CD}, (\mathcal{R}_0, \mathcal{R}_1)}^{\text{ss-cd}}(1^\lambda, b)$:

1. \mathcal{R}_0 samples $(pk, sk) \leftarrow \text{FHE-CD.Setup}(1^\lambda)$. It sets $s_0 := 0^\lambda$ and $s_1 := sk$ and sends (s_0, s_1) to the challenger.
2. The challenger computes $(sh, csh, vk) \leftarrow \text{Share}(s_b)$. It sends csh to \mathcal{R}_0 and sh to \mathcal{R}_1 .
3. \mathcal{R}_0 runs \mathcal{A}_0 . \mathcal{A}_0 sends (s'_0, s'_1) to \mathcal{R}_0 .
4. \mathcal{R}_0 sends (pk, ct, csh) to \mathcal{A}_0 , where $ct \leftarrow \text{FHE-CD.Enc}(pk, s'_c)$ and $c \leftarrow \{0, 1\}$.
5. \mathcal{A}_0 sends R'_0 to \mathcal{R}_0 . \mathcal{R}_0 sets $R_0 := (R'_0, c)$ and sends it to the challenger.
6. \mathcal{R}_1 runs \mathcal{A}_1 on input sh . If \mathcal{A}_1 outputs (cert_1, R'_1) , then \mathcal{R}_1 sets $R_1 := R'_1$ and sends it to the challenger.
7. The challenger computes $\text{ans} = \text{Vrfy}(vk, \text{cert}_1)$. If $\text{ans} = \top$, it outputs (R_0, R_1) .

Consider now the experiment $\text{Exp}_{\text{SS-CD}, (\mathcal{R}_0, \mathcal{R}_1)}^{\text{ss-cd}}(1^\lambda, 0)$. Notice that if there exists a QPT algorithm \mathcal{A} that obtains the registers (R'_0, R'_1) and outputs $c' = c$ with probability $\frac{1}{2} + \text{non-negl}(\lambda)$, then the security of FHE-CD is broken. This is because a reduction can obtain an FHE-CD ciphertext and simulate the view of $\mathcal{A}_0, \mathcal{A}_1$ as needed, because knowledge of sk is not required.

By assumption, there exists a QPT algorithm \mathcal{A} that obtains (R'_0, R'_1) and outputs $c' = c$ with probability $\frac{1}{2} + \text{non-negl}(\lambda)$ in the experiment $\text{Exp}_{\text{SS-CD}, (\mathcal{R}_0, \mathcal{R}_1)}^{\text{ss-cd}}(1^\lambda, 1)$.

Now, consider an algorithm \mathcal{R} that obtains $(R_0 = (c, R'_0), R_1 = R'_1)$. It runs \mathcal{A} on (R'_0, R'_1) and checks if the value c' equals c or not. If it is, then \mathcal{R} outputs $b' = 1$, otherwise it outputs $b' = 0$. Consequently, \mathcal{R} outputs $b' = b$ with probability $\frac{1}{2} + \text{non-negl}(\lambda)$, breaking the security of the scheme SS-CD. This gives us a contradiction. \square

3.2 Impossibility Results

Theorem 3.2. *Any $(stat, stat)$ -HSS-CD scheme is also an information-theoretic HSS scheme.*

Theorem 3.3. *The classical impossibility of [?] extends to the setting of quantum secret-shares.*

Theorem 3.4. *There does not exist a $(comp)$ -HSS-CD scheme with additive reconstruction, given that the algorithm S_{share} only outputs shares sh_1, sh_2 that are non-entangled.*

[GVW12]

References

- [GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 162–179. Springer, Heidelberg, August 2012. (Cited on page [8](#).)