Homomorphic Secret-Sharing with Certified Deletion

Nikhil Pappu[†]

[†]Portland State University
nikpappu@pdx.edu

October 18, 2024

Abstract

Contents

1 Preliminaries

2 HSS with Certified Deletion

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

- 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 for a PPT circuit family $\mathcal{C}$ is a tuple of 5 algorithms $\mathsf{HSS\text{-}CD} = \mathsf{HSS\text{-}CD}.(\mathcal{S}hare, \mathcal{E}val, \mathcal{D}el, \mathsf{Vrfy}, \mathcal{R}ec)$ with the following properties:

Syntax:

$\mathcal{S}hare(s) \rightarrow (sh_0^0, \mathsf{vk}_0), (sh_1^0, \mathsf{vk}_1)$: 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 $\mathsf{vk}_0, \mathsf{vk}_1$.

$\mathcal{E}val(C_j, i, sh_i^{j-1}) \rightarrow sh_i^j$: 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.

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

$\mathsf{Vrfy}(i, \mathsf{vk}_i, \mathsf{cert}_i) \rightarrow \top/\bot$: 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 $\bot$.

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

Evaluation Correctness: $\forall q = \mathrm{poly}(\lambda)$ and $\forall\ (C_1, \ldots, C_q) \in \mathcal{C}^q$, the following condition holds:

$$\Pr\left[(d_1, \cdots, d_q) = (C_1(s), \cdots, C_q(s)) \ : \ \begin{array}{l} (sh_0^0, \mathsf{vk}_0), (sh_1^0, \mathsf{vk}_1) \leftarrow \mathcal{S}hare(s) \\ \forall i, j \in \{0,1\} \times [q] : sh_i^j \leftarrow \mathcal{E}val(C_j, i, sh_i^{j-1}) \\ (d_1, \cdots, d_q) \leftarrow \mathcal{R}ec(sh_0^q, sh_1^q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

Deletion Correctness: The following condition holds for all $i \in \{0,1\}$, $q = \mathrm{poly}(\lambda)$ and $(C_1, \ldots, C_q) \in \mathcal{C}^q$:

$$\Pr\left[\mathsf{Vrfy}(i, \mathsf{vk}_i, \mathsf{cert}_i) \rightarrow \top \ : \ \begin{array}{l} (sh_0^0, \mathsf{vk}_0), (sh_1^0, \mathsf{vk}_1) \leftarrow \mathcal{S}hare(s) \\ \forall j \in [q] : sh_i^j \leftarrow \mathcal{E}val(C_j, i, sh_i^{j-1}) \\ \mathsf{cert}_i \leftarrow \mathcal{D}el(i, sh_i^q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

Compactness: The following condition holds for all $i \in \{0,1\}$, $q = \mathrm{poly}(\lambda)$ and $(C_1, \ldots, C_q) \in \mathcal{C}^q$, where l_q denotes the output length of the circuit C_q:

$$\left[|sh_i^q| - |sh_i^{q-1}| = \mathrm{poly}(1^\lambda, l_q) \ : \ \begin{array}{l} (sh_0^0, \mathsf{vk}_0), (sh_1^0, \mathsf{vk}_1) \leftarrow \mathcal{S}hare(s) \\ \forall j \in [q] : sh_i^j \leftarrow \mathcal{E}val(C_j, i, sh_i^{j-1}) \end{array}\right]$$

2.2 Additive HSS-CD Syntax

A scheme satisfying the additive HSS-CD syntax for a PPT circuit family $\mathcal{C}$ is a tuple of 5 algorithms $\mathsf{HSS\text{-}CD} = \mathsf{HSS\text{-}CD}.(\mathcal{Share}, \mathcal{Eval}, \mathcal{Obs}, \mathcal{Del}, \mathsf{Vrfy})$ with the following properties:

Syntax:

$\mathcal{Share}(s) \to (\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1)$: The sharing algorithm outputs quantum (possibly-entangled) secret-shares $\mathcal{sh}_0^0, \mathcal{sh}_1^0$ encoding an input secret s. It also outputs the corresponding classical verification keys $\mathsf{vk}_0, \mathsf{vk}_1$.

$\mathcal{Eval}(C_j, i, \mathcal{sh}_i^{j-1}) \to \mathcal{sh}_i^j$: The evaluation algorithm takes the description of a PPT computable circuit C_j, an index $i \in \{0,1\}$ and a share $\mathcal{sh}_i^{j-1}$. It outputs a quantum share $\mathcal{sh}_i^j$.

$\mathcal{Obs}(i, \mathcal{sh}_i^j) \to (y_0^1, \ldots, y_0^j)$: The observation algorithm takes an index $i \in \{0,1\}$ and a quantum state $\mathcal{sh}_i^j$ and produces a j-tuple of classical shares.

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

$\mathsf{Vrfy}(i, \mathsf{vk}_i, \mathsf{cert}_i) \to \top/\bot$: 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 $\bot$.

Evaluation Correctness: The following condition holds for all $q = \mathrm{poly}(\lambda)$ and $(C_1, \ldots, C_q) \in \mathcal{C}^q$:

$$\Pr\left[(y_0^1 \oplus y_1^1, \cdots, y_0^q \oplus y_1^q) = (C_1(s), \cdots, C_q(s)) \ :\ \begin{array}{l} (\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{Share}(s) \\ \forall i, j \in \{0,1\} \times [q] : \mathcal{sh}_i^j \leftarrow \mathcal{Eval}(C_j, i, \mathcal{sh}_i^{j-1}) \\ (y_i^1, \ldots, y_i^q) \leftarrow \mathcal{Obs}(i, \mathcal{sh}_i^q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

In the case of additive HSS-CD, we will consider the following weaker deletion guarantee:

Deletion Correctness: The following condition holds for all $i \in \{0,1\}$, $q = \mathrm{poly}(\lambda)$ and $(C_1, \ldots, C_q) \in \mathcal{C}^q$:

$$\Pr\left[\mathsf{Vrfy}(i, \mathsf{vk}_i, \mathsf{cert}_i) \to \top \ :\ \begin{array}{l} (\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{Share}(s) \\ \forall j \in [q] : \mathcal{sh}_i^j \leftarrow \mathcal{Eval}(C_j, i, \mathcal{sh}_i^{j-1}) \\ \mathsf{cert}_i \leftarrow \mathcal{Del}(i, \mathcal{sh}_i^q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

2.3 Security Definitions

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

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

1. $\mathcal{A}_0$ sends $(s_0, s_1) \in \{0,1\}^\lambda$ to the challenger.
2. The challenger runs $(\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{Share}(s_b)$ and sends each $\mathcal{sh}_i^0$ to $\mathcal{A}_i$.
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 $\mathsf{Vrfy}(j, \mathsf{vk}_j, \mathsf{cert}_j) = \top$, then output (R_0, R_1).

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

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

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

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

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

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

1. $\mathcal{A}_0$ sends $(s_0, s_1) \in \{0,1\}^\lambda$ to the challenger. The challenger runs $(\mathit{sh}_0^0, \mathsf{vk}_0), (\mathit{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{S}\mathit{hare}(s_b)$ and sends each sh_i^0 to $\mathcal{A}_i$.
2. $\mathcal{A}_0$ sends (cert_0, R_0) and $\mathcal{A}_1$ sends (cert_1, R_1) where R_0, R_1 are some registers.
3. If $\mathsf{Vrfy}(0, \mathsf{vk}_0, \mathsf{cert}_0) = \mathsf{Vrfy}(1, \mathsf{vk}_1, \mathsf{cert}_1) = \top$, then output (R_0, R_1).

Statistical Double-Deletion Security holds if the following holds:

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

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

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

Computational Secrecy wrt Share j**:** The following security notion is defined wrt a QPT adversary $\mathcal{A}$:

$\mathsf{Expt}^{\mathsf{ind}}_{\mathsf{HSS\text{-}CD},\mathcal{A}}(1^\lambda, j, b)$**:**

1. $\mathcal{A}$ sends $(s_0, s_1) \in \{0,1\}^\lambda$ to the challenger. The challenger runs $(\mathit{sh}_0^0, \mathsf{vk}_0), (\mathit{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{S}\mathit{hare}(s_b)$ and sends sh_i^0 to $\mathcal{A}$.
2. $\mathcal{A}$ sends b' to the challenger. The challenger outputs b'.

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

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

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

Definition 2.2 ((Additive) (X)-HSS-CD scheme for $\mathcal{C}$). *An (Additive) (X)-HSS-CD scheme for $\mathcal{C}$ where X $\in \{stat, comp\}$ is a scheme satisfying the (Additive) HSS-CD syntax for $\mathcal{C}$, the X double-deletion security, and computational secrecy wrt share* 0 *and share* 1*.*

Remark 2.3. Notice that Deletion Security wrt Share j implies Computational Secrecy wrt Share $1 - j$, but the Double Deletion security does not imply Computational Secrecy.

Remark 2.4. Observe that a (stat, comp)-HSS-CD scheme for $\mathcal{C}$ is also a (stat)-HSS-CD scheme for $\mathcal{C}$. Likewise, a (comp, comp)-HSS-CD scheme for $\mathcal{C}$ is also a (comp)-HSS-CD scheme for $\mathcal{C}$.

3 Impossibility Results

Lemma 3.1. *Any (stat, stat)-HSS-CD scheme for $\mathcal{C}$ is also an information-theoretic HSS scheme for $\mathcal{C}$.*

Proof. Suppose there exists a (stat, stat)-HSS-CD scheme that does not satisfy information-theoretic secrecy. Then, there exists an unbounded adversary $\mathcal{D}$ that receives some share sh_i^0 and distinguishes between the secrets s_0, s_1. Then, there exists an adversary $(\mathcal{A}_0, \mathcal{A}_1)$ in the statistical security wrt share $1-i$ game that works as follows. $\mathcal{A}_{1-i}$ honestly deletes its share and outputs a dummy register while $\mathcal{A}_i$ outputs a register containing its share sh_i^0. In the second-stage, the distinguisher $\mathcal{D}$ is run on sh_i^0 to tell apart the secrets s_0, s_1. □

The following theorem shows that the classical impossibility result regarding information-theoretic HSS by [BGI+18] also applies to the setting of quantum shares.

Theorem 3.2. *TBD.*

Theorem 3.3. *There does not exist an Additive (comp)-HSS-CD scheme* $\mathsf{HSS\text{-}CD}$ *for any PPT circuit class $\mathcal{C}$, given that* $\mathsf{HSS\text{-}CD}.\mathcal{Share}(s)$ *outputs shares sh_0, sh_1 that are not entangled with each other.*

Proof. In fact, we will prove that this holds even for a weaker notions of evaluation and deletion correctness, where $\mathcal{Eval}$ and $\mathcal{Del}$ support only a single evaluation. Specifically, for shares (sh_0, sh_1) output by $\mathcal{Share}(s)$, $\mathcal{Eval}(i, C, sh_i)$ outputs a share $\widetilde{sh}_i$, and $\mathcal{Obs}(i, \widetilde{sh}_i)$ outputs a value y_i such that $y_0 \oplus y_1 = C(s)$. Moreover, $\mathcal{Del}(i, \widetilde{sh}_i)$ outputs cert_i such that $\mathsf{Vrfy}(i, \mathsf{vk}_i, \mathsf{cert}_i) = \top$. Furthermore, we will not need to rely on computational secrecy of either share, but only the computational double deletion security. The argument proceeds as follows:

Let $(|\psi_0\rangle, \mathsf{vk}_0), (|\psi_1\rangle, \mathsf{vk}_1)$ be some pure state output by $\mathcal{Share}(s)$, where $|\psi_0\rangle, |\psi_1\rangle$ are not entangled with each other. Let $\left|\widetilde{\psi_i}\right\rangle$ be the state output by $\mathsf{Eval}(i, C, |\psi_i\rangle)$. Wlog, let $\{\Pi_0, \mathbb{I} - \Pi_0\}$ be the projective measurement equivalent of $\mathcal{Obs}(0, \left|\widetilde{\psi_0}\right\rangle)$, i.e., Π_0 corresponds to $y_0 = 0$ and $\mathbb{I} - \Pi_1$ corresponds to $y_0 = 1$. Let Y_0 denote the random variable of the output y_0. Likewise, consider the projective measurement $\{\Pi_1, \mathbb{I} - \Pi_1\}$ equivalent of $\mathcal{Obs}(1, \left|\widetilde{\psi_1}\right\rangle)$ and let Y_1 be the corresponding output random variable. Notice that for every outcome y_0 of Y_0, there is a single outcome y_1 of Y_1 that satisfies $y_1 = C(s) \oplus y_0$. Let $\widetilde{Y_0}$ be the random variable for $\widetilde{y_0}$ sampled as $\widetilde{y_0} = C(s) \oplus y_1 : y_1 \leftarrow Y_1$. By the evaluation correctness requirement, we require that $\Pr\left[Y_0 = \widetilde{Y_0}\right] \geq 1 - \mathsf{negl}(\lambda)$. Since Y_0 and $\widetilde{Y_0}$ are independent random variables, this is only possible if there exists $y_0^\star$ such that $\Pr[Y_0 = y_0^\star] \geq 1 - \mathsf{negl}(\lambda)$ and $\Pr\left[\widetilde{Y_0} = y_0^\star\right] \geq 1 - \mathsf{negl}(\lambda)$. In other words, the measurement $\{\Pi_0, \mathbb{I} - \Pi_0\}$ either accepts the state $|\psi_0\rangle$ with probability $1 - \mathsf{negl}(\lambda)$ or rejects it with probability $1 - \mathsf{negl}(\lambda)$. Consequently, by the gentle measurement lemma, the leftover state is close in trace distance to the state $|\psi_0\rangle$. As a result, it can be certifiably deleted after obtaining y_0. By a similar argument, y_1 can be obtained in the same way. Since this holds for every possible pure state output by $\mathcal{Share}(s)$, it also holds for arbitrary mixed states. As a result, the adversary can efficiently compute $y_0 \oplus y_1 = C(s)$ in the second-stage, breaking the computational double-deletion security. Since this security notion is the weakest one, this also rules out the other notions. □

4 Feasibility Results

4.1 FHE-CD based Construction

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

- Fully Homomorphic Encryption with Certified Deletion (FHE-CD) scheme $\mathsf{FHE\text{-}CD} = \mathsf{FHE\text{-}CD}.(\mathsf{Setup}, \mathcal{Enc}, \mathcal{Dec}, \mathcal{Eval}, \mathcal{Del}, \mathsf{Vrfy})$.
- Secret Sharing with Certified Deletion (SS-CD) scheme $\mathsf{SS\text{-}CD} = \mathsf{SS\text{-}CD}.(\mathcal{Share}, \mathcal{Rec}, \mathcal{Del}, \mathsf{Vrfy})$.

The construction is as follows.

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

1. Generate $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE\text{-}CD.Setup}(1^\lambda)$.
2. Compute $(\mathsf{fhecd}.ct^0, \mathsf{fhecd.vk}) \leftarrow \mathsf{FHE\text{-}CD}.\mathcal{Enc}(s)$.
3. Compute $(\mathsf{sscd}.sh, \mathsf{sscd.csh}), \mathsf{sscd.vk} \leftarrow \mathsf{SS\text{-}CD}.\mathcal{Share}(\mathsf{sk})$.
4. Set $sh_0^0 := (\mathsf{fhecd.pk}, \mathsf{fhecd}.ct^0, \mathsf{sscd.csh})$ and $\mathsf{vk}_0 := \mathsf{fhecd.vk}$.
5. Set $sh_1^0 := \mathsf{sscd}.sh$ and $\mathsf{vk}_1 := \mathsf{sscd.vk}$.
6. Output $(sh_0^0, \mathsf{vk}_0), (sh_1^0, \mathsf{vk}_1)$.

$\mathsf{HSS\text{-}CD}.\mathcal{Eval}(C_j, i, sh_i^{j-1})$: If $i = 1$, set $sh_1^j := sh_1^{j-1}$. Else, execute the following:

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

$\mathsf{HSS\text{-}CD}.\mathcal{Del}(i, sh_i^j)$:

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

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

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

$\mathsf{HSS\text{-}CD}.\mathcal{Rec}(sh_0^q, sh_1^q)$:

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

Theorem 4.1. *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 $\mathsf{HSS\text{-}CD}$ 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 $\mathsf{FHE\text{-}CD}$. Consider a QPT reduction $\mathcal{R}$ that runs as follows in the $\mathsf{FHE\text{-}CD}$ game:

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

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 $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Setup}(1^\lambda)$ and sends pk to $\mathcal{R}$.
3. The challenger encrypts s_b as $ct \leftarrow \mathcal{E}nc(\mathsf{pk}, s_b)$ and sends ct to $\mathcal{R}$.
4. $\mathcal{R}$ computes $sh_0^0 := (\mathsf{pk}, ct, \mathsf{sscd.csh})$, where $\mathsf{sscd.csh} \leftarrow \mathsf{SS\text{-}CD.Sim}(1^\lambda)$.
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 $\mathsf{ans} \leftarrow \mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}_0)$. If $\mathsf{ans} = \top$, it sends sk to $\mathcal{R}$. Else, it outputs $\perp$.
7. $\mathcal{R}$ computes $\mathsf{sscd}.sh$ conditioned on $(\mathsf{sscd}.sh, \mathsf{sscd.sh})$ encoding sk.
8. $\mathcal{R}$ sends $sh_1^0 := \mathsf{sscd}.sh$ 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 $\mathsf{FHE\text{-}CD}$. Observe that the view of $\mathcal{A}_0$ in the reduction is identically distributed to its view in $\mathsf{Expt}^{\mathsf{del}}_{\mathsf{HSS\text{-}CD},(\mathcal{A}_0,\mathcal{A}_1)}(1^\lambda, 0, b)$. Now, notice that if $\mathsf{HSS\text{-}CD.Vrfy}(0, \mathsf{vk}_0, \mathsf{cert}_0)$ passes, then $\mathsf{FHE\text{-}CD.Vrfy}(\mathsf{vk}, \mathsf{cert}_0)$ also passes. Consequently, $\mathcal{R}$ receives the secret key sk. By the information-theoretic secrecy of the scheme $\mathsf{SS\text{-}CD}$, 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 $\mathsf{HSS\text{-}CD}$ 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 $\mathsf{FHE\text{-}CD}$.

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 $\mathsf{SS\text{-}CD}$. Consider a non-local reduction $(\mathcal{R}_0, \mathcal{R}_1)$ that runs as follows:

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

1. $\mathcal{R}_0$ samples $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{FHE\text{-}CD.Setup}(1^\lambda)$. It sets $s_0 := 0^\lambda$ and $s_1 := \mathsf{sk}$ and sends (s_0, s_1) to the challenger.
2. The challenger computes $(sh, \mathsf{csh}, \mathsf{vk}) \leftarrow \mathcal{S}hare(s_b)$. 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 $(\mathsf{pk}, ct, \mathsf{csh})$ to $\mathcal{A}_0$, where $ct \leftarrow \mathsf{FHE\text{-}CD}.\mathcal{E}nc(\mathsf{pk}, s_c')$ 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 $\mathsf{ans} = \mathsf{Vrfy}(\mathsf{vk}, \mathsf{cert}_1)$. If $\mathsf{ans} = \top$, it outputs (R_0, R_1).

Consider now the experiment $\mathsf{Exp}^{\mathsf{ss\text{-}cd}}_{\mathsf{SS\text{-}CD},(\mathcal{R}_0,\mathcal{R}_1)}(1^\lambda, 0)$. Notice that if there exists a QPT algorithm $\mathcal{A}$ that obtains the registers (R_0', R_1') and outputs $c' = c$ with probability $\frac{1}{2} + \mathsf{non\text{-}negl}(\lambda)$, then the security of $\mathsf{FHE\text{-}CD}$ is broken. This is because a reduction can obtain an $\mathsf{FHE\text{-}CD}$ ciphertext and simulate the view os $\mathcal{A}_0, \mathcal{A}_1$ 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 $\frac{1}{2} + \mathsf{non\text{-}negl}(\lambda)$ in the experiment $\mathsf{Exp}^{\mathsf{ss\text{-}cd}}_{\mathsf{SS\text{-}CD},(\mathcal{R}_0,\mathcal{R}_1)}(1^\lambda, 1)$.

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 $\frac{1}{2} + \mathsf{non\text{-}negl}(\lambda)$, breaking the security of the scheme $\mathsf{SS\text{-}CD}$. This gives us a contradiction. □

4.2 Spooky-Encryption based Construction with Entangled Shares

5 Additive HSS with Weak Certified-Deletion

5.1 Additive HSS-wCD Syntax

A scheme satisfying the additive HSS-wCD syntax for a PPT circuit family $\mathcal{C}$ is a tuple of 4 algorithms $\mathsf{HSS\text{-}wCD} = \mathsf{HSS\text{-}wCD}.(\mathcal{S}hare, \mathcal{E}val, \mathcal{D}el, \mathsf{Vrfy})$ with the following properties:

Syntax:

$\mathcal{Share}(s) \to (\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1)$: The sharing algorithm outputs quantum (possibly-entangled) secret-shares $\mathcal{sh}_0^0, \mathcal{sh}_1^0$ encoding an input secret s. It also outputs the corresponding classical verification keys $\mathsf{vk}_0, \mathsf{vk}_1$.

$\mathcal{Eval}(C_j, i, \mathcal{sh}_i^{j-1}) \to (y_i^j, \mathcal{sh}_i^j)$: The evaluation algorithm takes the description of a PPT computable circuit C_j, an index $i \in \{0,1\}$ and a share $\mathcal{sh}_i^{j-1}$. It outputs a quantum share $\mathcal{sh}_i^j$ and a classical additive share y_i^j.

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

$\mathsf{Vrfy}(i, \mathsf{vk}_i, \mathsf{cert}_i) \to \top/\bot$: 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 $\bot$.

Evaluation Correctness: The following condition holds for all $q = \text{poly}(\lambda)$ and $(C_1, \ldots, C_q) \in \mathcal{C}^q$:

$$\Pr\left[(y_0^1 \oplus y_1^1, \cdots, y_0^q \oplus y_1^q) = (C_1(s), \cdots, C_q(s)) \;:\; \begin{array}{l} (\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{Share}(s) \\ \forall i, j \in \{0,1\} \times [q] : (y_i^j, \mathcal{sh}_i^j) \leftarrow \mathcal{Eval}(C_j, i, \mathcal{sh}_i^{j-1}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

Deletion Correctness: The following condition holds for all $i \in \{0,1\}$, $q = \text{poly}(\lambda)$ and $(C_1, \ldots, C_q) \in \mathcal{C}^q$:

$$\Pr\left[\mathsf{Vrfy}(i, \mathsf{vk}_i, \mathsf{cert}_i) \to \top \;:\; \begin{array}{l} (\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{Share}(s) \\ \forall j \in [q] : (y_i^j, \mathcal{sh}_i^j) \leftarrow \mathcal{Eval}(C_j, i, \mathcal{sh}_i^{j-1}) \\ \mathsf{cert}_i \leftarrow \mathcal{Del}(i, \mathcal{sh}_i^q) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda)$$

5.2 Security Definitions

Deletion Security wrt Share j, Circuit Class $\mathcal{C}$, and Distribution $\mathcal{D}$: The following security notion is defined wrt a non-local quantum adversary $(\mathcal{A}_0, \mathcal{A}_1)$:

$\mathsf{Exp}^{\mathsf{del\text{-}weak}}_{\mathsf{HSS\text{-}wCD},(\mathcal{A}_0,\mathcal{A}_1)}(1^\lambda, j, \mathcal{C}, \mathcal{D}, b)$**:**

1. The challenger samples $(s_0, s_1) \leftarrow \mathcal{D}$ and sends (s_0, s_1) to both $\mathcal{A}_0, \mathcal{A}_1$.
2. The challenger runs $(\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{Share}(s_b)$ and sends each $\mathcal{sh}_i^0$ to $\mathcal{A}_i$.
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 $\mathsf{Vrfy}(j, \mathsf{vk}_j, \mathsf{cert}_j) = \top$, then output (R_0, R_1).

Need to Formalize this: Let $\mathcal{D}$ be a distribution such that for (s_0, s_1) drawn from $\mathcal{D}$, any QPT oracle algorithm $\mathcal{B}^{\mathcal{C}(\cdot)}$ cannot distinguish between (s_0, s_1).

Statistical (*likewise,* Computational) Deletion Security holds if the following holds for all *hard-given-$\mathcal{C}$* distributions $\mathcal{D}$ and unbounded (*likewise*, QPT) algorithms $\mathcal{A}$:

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

Double-Deletion Security wrt Circuit Class $\mathcal{C}$, and Distribution $\mathcal{D}$: The following security notion is defined wrt a non-local quantum adversary $(\mathcal{A}_0, \mathcal{A}_1)$:

$\mathsf{Exp}^{\mathsf{del\text{-}weak\text{-}2}}_{\mathsf{HSS\text{-}wCD},(\mathcal{A}_0,\mathcal{A}_1)}(1^\lambda, \mathcal{C}, \mathcal{D}, b)$**:**

1. The challenger samples $(s_0, s_1) \leftarrow \mathcal{D}$ and sends (s_0, s_1) to both $\mathcal{A}_0, \mathcal{A}_1$.
2. The challenger runs $(\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{Share}(s_b)$ and sends each $\mathcal{sh}_i^0$ to $\mathcal{A}_i$.
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 $\mathsf{Vrfy}(0, \mathsf{vk}_0, \mathsf{cert}_0) = \mathsf{Vrfy}(1, \mathsf{vk}_1, \mathsf{cert}_1) = \top$, then output (R_0, R_1).

Statistical (*likewise*, Computational) Double-Deletion Security holds if the following holds for all *hard-given-$\mathcal{C}$* distributions $\mathcal{D}$ and unbounded (*likewise*, QPT) algorithms $\mathcal{A}$:

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

Computational Secrecy wrt Share j: The following security notion is defined wrt a QPT adversary $\mathcal{A}$:

$\mathsf{Expt}^{\mathsf{ind}}_{\mathsf{HSS\text{-}wCD},\mathcal{A}}(1^\lambda, j, b)$**:**

1. $\mathcal{A}$ sends $(s_0, s_1) \in \{0,1\}^\lambda$ to the challenger. The challenger runs $(\mathcal{sh}_0^0, \mathsf{vk}_0), (\mathcal{sh}_1^0, \mathsf{vk}_1) \leftarrow \mathcal{Share}(s_b)$ and sends $\mathcal{sh}_i^0$ to $\mathcal{A}$.
2. $\mathcal{A}$ sends b' to the challenger. The challenger outputs b'.

$$\left| \Pr\left[\mathsf{Expt}^{\mathsf{ind}}_{\mathsf{HSS\text{-}wCD},\mathcal{A}}(1^\lambda, j, 0) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{ind}}_{\mathsf{HSS\text{-}wCD},\mathcal{A}}(1^\lambda, j, 1) = 1\right] \right| \leq \mathsf{negl}(\lambda)$$

References

[BGI+18] Elette Boyle, Niv Gilboa, Yuval Ishai, Huijia Lin, and Stefano Tessaro. Foundations of homomorphic secret sharing. In Anna R. Karlin, editor, *ITCS 2018*, volume 94, pages 21:1–21:21. LIPIcs, January 2018. (Cited on page 6.)