A Note on Security Definitions for Secret Sharing with Certified Deletion

Dominique Bazin¹ and Ryo Nishimaki^{2,3}

 ¹ DIENS, ENS, PSL University, Paris, France 45 rue d'Ulm dominique.bazin@ens.psl.eu
² NTT Social Informatics Laboratories, Tokyo, Japan ryo.nishimaki@ntt.com
³ NTT Research Center for Theoretical Quantum Information, Atsugi, Japan

Abstract. Bartusek and Raizes (CRYPTO 2024) proposed two security definitions for secret sharing, *no-signaling certified deletion* and *adaptive certified deletion*. We prove that adaptive certified deletion does not imply no-signaling certified deletion.

1 Introduction

Secret Sharing. Secret sharing allows a dealer to split a secret s into n shares so that any k shares can be used together to recover the secret. However, any set of (k-1) shares gives no information on s. Shamir [4] introduced (k, n) threshold secret sharing in 1979.

Bartusek and Khurana [1] introduced secret sharing with certified deletion, allowing the share dealer to ask the parties for certificates of deletion of the shares. A valid certificate should ensure that any reconstruction power held by the share has been destroyed. Bartusek and Raizes [2] then generalized this notion and presented more general schemes with certified deletion. They proposed two definitions of security that allow a user who suspects a data breach to request and verify that the breached data is deleted: *no-signaling certified deletion* (NSCD) and *adaptive certified deletion* (ACD). They prove that NSCD does not imply ACD and claim these notions to be incomparable but have no proof of the inverse. In this note, we prove that ACD does not imply NSCD.

2 Preliminaries

We use λ to denote security parameters. We write $\operatorname{negl}(.)$ to denote any negligible function, which is a function f such that for every constant $c \in \mathbb{N}$ there exists $N \in \mathbb{N}$ such that for all $n \geq N$, $f(n) \leq n^{-c}$. Part of the following sections are taken from the preliminaries section by Bartusek and Khurana [1] and definitions of Bartusek and Raizes [2].

2.1 Quantum Computation

A register \mathcal{X} is a named Hilbert space \mathbb{C}^{2^n} . A pure quantum state on register \mathcal{X} is a unit vector $|\psi\rangle^{\mathcal{X}} \in \mathbb{C}^{2^n}$, and we say that $|\psi\rangle^{\mathcal{X}}$ consists of n qubits. A mixed state on register \mathcal{X} is described by a density matrix $\rho^{\mathcal{X}} \in \mathbb{C}^{2^n \times 2^n}$, which is a positive semi-definite Hermitian operator with trace 1.

We will make use of the convention that 0 denotes the computational basis $\{|0\rangle, |1\rangle\}$ and 1 denotes the Hadamard basis $\{\frac{|0\rangle+|1\rangle}{2}, \frac{|0\rangle-|1\rangle}{2}\}$. For a bit $r \in \{0, 1\}$, we write $|r\rangle_0$ to denote r encoded in the computational basis, and $|r\rangle_1$ to denote r encoded in the Hadamard basis. For strings $x, \theta \in \{0, 1\}^{\lambda}$, we write $|x\rangle_{\theta}$ to mean $|x_1\rangle_{\theta_1}, \ldots, |x_{\lambda}\rangle_{\theta_{\lambda}}$. This corresponds to what we call a BB84 [3] state.

2.2 State of the Art

We are here going to summarise definitions of Bartusek and Raizes [2].

Secret sharing scheme with certified deletion augments the syntax of secret sharing scheme with additional algorithms to delete shares and verify deletion certificates. We define it for general access structures. An access structure $\mathbb{S} \subseteq \mathcal{P}([n])$ for n parties is a monotonic set of sets, i.e., if $S \in \mathbb{S}$ and $S' \supset S$, then $S' \in \mathbb{S}$. Any set of parties $S \in \mathbb{S}$ is authorized to access the secret. A simple example of an access structure is the threshold structure, where any set of at least k parties is authorized to access the secret. We denote this access structure as (k, n) and call it a threshold access structure.

Definition 1 (Secret Sharing with Certified Deletion). A secret sharing scheme with certified deletion is specified by a monotone access structure S over n parties, and consists of four algorithms:

- $\mathsf{Split}_{\mathbb{S}}(1^{\lambda}, s)$ takes in a secret s, and outputs n share registers $\mathcal{S}_1, \ldots, \mathcal{S}_n$ and a verification key vk.
- Reconstruct_S($\{S_i\}_{i \in P}$) takes in a set of share registers for some $P \subseteq [n]$, and outputs either s or \perp .
- $\mathsf{Delete}_{\mathbb{S}}(\mathcal{S}_i)$ takes in a share register and outputs a certificate of deletion cert.
- Verify_S(vk, *i*, cert) takes in the verification key vk, an index *i*, and a certificate of deletion cert, and outputs either \top (indicating accept) or \perp (indicating reject).

Definition 2 (Correctness of Secret Sharing with Certified Deletion).

A secret sharing scheme with certified deletion must satisfy two correctness properties:

- **Reconstruction Correctness.** For all $\lambda \in \mathbb{N}$ and all sets $S \in \mathbb{S}$,

 $\Pr\left[\mathsf{Reconstruct}_{\mathbb{S}}(\{\mathcal{S}_i\}_{i\in S}): (\mathcal{S}_1, \dots, \mathcal{S}_n, \mathsf{vk}) \leftarrow \mathsf{Split}_{\mathbb{S}}(1^{\lambda}, s)\right] = 1.$

- **Deletion Correctness.** For all $\lambda \in \mathbb{N}$ and all $i \in [n]$,

$$\Pr\left[\mathsf{Verify}_{\mathbb{S}}(\mathsf{vk}, i, \mathsf{cert}) = \top : \frac{(\mathcal{S}_1, \dots, \mathcal{S}_n, \mathsf{vk}) \leftarrow \mathsf{Split}_{\mathbb{S}}(1^\lambda, s)}{\mathsf{cert} \leftarrow \mathsf{Delete}_{\mathbb{S}}(\mathcal{S}_i)}\right] = 1.$$

The standard notion of security for secret sharing is called privacy :

Privacy. There exists a randomized algorithm Sim such that for all subsets $P \subseteq [n]$ such that for all $S \in \mathbb{S}$, $P \not\subseteq S$, and any s,

$$\{\{\mathsf{sh}_i\}_{i\in P} : (\mathsf{sh}_1,\ldots,\mathsf{sh}_n) \leftarrow \mathsf{Split}_{\mathbb{S}}(s)\} \equiv \{\{\mathsf{sh}_i\}_{i\in P} : \{\mathsf{sh}_i\}_{i\in P} \leftarrow \mathsf{Sim}(P)\}.$$

Then, Bartusek and Raizes introduced two security notions that differ in how the adversary can access the shares. First, no-signaling where multiple adversaries see small portions of the shares and can join their view once enough deletions have been made:

Definition 3 (No-Signaling Certified Deletion Security for Secret Sharing). Let $P = (P_1, \ldots, P_\ell)$ be a partition of [n], let $|\psi\rangle$ be an ℓ -part state on registers $\mathcal{R}_1, \ldots, \mathcal{R}_\ell$, and let $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_\ell)$ be an ℓ -part adversary. Define the experiment $SS-NSCD_{\mathbb{S}}(1^{\lambda}, P, |\psi\rangle, \mathcal{A}, s)$ as follows:

- 1. Sample $(\mathcal{S}_1, \ldots, \mathcal{S}_n, \mathsf{vk}) \leftarrow \mathsf{Split}_{\mathbb{S}}(1^\lambda, s)$.
- 2. For each $t \in [\ell]$, run ({cert_i}_{i \in P_t}, \mathcal{R}'_t) \leftarrow \mathcal{A}_t(\{\mathcal{S}_i\}_{i \in P_t}, \mathcal{R}_t), where \mathcal{R}'_t is an arbitrary output register.
- 3. If for all $S \in \mathbb{S}$, there exists $i \in S$ such that $\operatorname{Verify}_{\mathbb{S}}(\mathsf{vk}, i, \operatorname{cert}_i) = \top$, then output $(\mathcal{R}'_1, \ldots, \mathcal{R}'_{\ell})$, and otherwise output \bot .

A secret sharing scheme for access structure S has no-signaling certified deletion security if for any "admissible" partition $P = (P_1, \ldots, P_\ell)$ (i.e. for all $P_t \in P$ and $S \in S$, $P_t \not\subseteq S$), any ℓ -part state $|\psi\rangle$, any (unbounded) ℓ -part adversary A, and any pair of secrets s_0, s_1 ,

$$\mathsf{TD}[\mathsf{SS}\mathsf{-}\mathsf{NSCD}_{\mathbb{S}}(1^{\lambda}, P, \ket{\psi}, \mathcal{A}, s_{0}), \ \mathsf{SS}\mathsf{-}\mathsf{NSCD}_{\mathbb{S}}(1^{\lambda}, P, \ket{\psi}, \mathcal{A}, s_{1})] = \mathsf{negl}(\lambda).$$

Secondly, an adaptive setting where only one adversary will be able to see all the shares one after the other as long as he gives enough valid certificates of deletion before seeing new shares :

Definition 4 (Adaptive Certified Deletion for Secret Sharing). Let \mathcal{A} be an adversary with internal register \mathcal{R} which is initialized to a state $|\psi\rangle$, let \mathbb{S} be an access structure, and let s be a secret. Define the experiment SS-ACD_S $(1^{\lambda}, |\psi\rangle, \mathcal{A}, s)$ as follows:

- 1. Sample $(S_1, \ldots, S_n, \mathsf{vk}) \leftarrow \mathsf{Split}_{\mathbb{S}}(1^\lambda, s)$. Initialize the corruption set $C = \emptyset$ and the deleted set $D = \emptyset$.
- 2. In each round i, the adversary may do one of three things:
 - (a) End the experiment by outputting a register $\mathcal{R} \leftarrow \mathcal{A}(\{\mathcal{S}_j\}_{j \in C}, \mathcal{R})$.
 - (b) Delete a share by outputting a certificate cert_i, an index j_i ∈ [n], and register (cert_i, j_i, R) ← A({S_j}_{j∈C}, R). When the adversary chooses this option, if Verify_S(vk, j_i, cert_i) outputs ⊤, then add j_i to D. Otherwise, immediately abort the experiment and output ⊥.
 - (c) Corrupt a new share by outputting an index $j_i \in [n]$ and register $(j_i, \mathcal{R}) \leftarrow \mathcal{A}(\{\mathcal{S}_j\}_{j \in C}, \mathcal{R})$. When the adversary chooses this option, add j_i to C. If $C \setminus D \in \mathbb{S}$, immediately abort the experiment and output \bot .

3. Output \mathcal{R} , unless the experiment has already aborted.

A secret sharing scheme for access structure \mathbb{S} has adaptive certified deletion security if for any (unbounded) adversary \mathcal{A} , any state $|\psi\rangle$, and any pair of secrets (s_0, s_1) ,

$$\mathsf{TD}[\mathsf{SS}\mathsf{-}\mathsf{ACD}_{\mathbb{S}}(1^{\lambda}, |\psi\rangle, \mathcal{A}, s_0), \ \mathsf{SS}\mathsf{-}\mathsf{ACD}_{\mathbb{S}}(1^{\lambda}, |\psi\rangle, \mathcal{A}, s_1)] = \mathsf{negl}(\lambda)$$

BB84 State and Certified Deletion $\mathbf{2.3}$

Next, we present the original 2-out-of-2 construction of secret sharing by Bartusek and Khurana [1] that uses BB84 states [3]. Their construction is simpler for two reasons: First, only one of the shares can be deleted, and second, there is no notion of no-signaling or adaptive as this is a 2 out of 2 scheme. Therefore, it has its own notion of security that we will present for reduction purposes:

Definition 5 (Certified deletion security).

Let $\mathcal{A} = {\mathcal{A}_{\lambda}}_{\lambda \in \mathbb{N}}$ denote an unbounded adversary and b denote a classical bit. Consider experiment $\mathsf{EV}\text{-}\mathsf{EXP}^{\mathcal{A}}_{\lambda}(b)$ which describes everlasting security given a deletion certificate, and is defined as follows.

- Sample $(s_1, s_2, \mathsf{vk}) \leftarrow \mathsf{Split}(b)$.
- Initialize \mathcal{A}_{λ} with s_1 .
- Parse \mathcal{A}_{λ} 's output as a deletion certificate cert and a residual state on register Α'.
- If Verify(vk, cert) = \top then output (A', s_2), and otherwise output \perp .

Then CD-SS = (Split, Reconstruct, Delete, Verify) satisfies certified deletion security if for any unbounded adversary \mathcal{A} , it holds that

$$\mathsf{TD}\left(\mathsf{EV}\text{-}\mathsf{EXP}^{\mathcal{A}}_{\lambda}(0),\mathsf{EV}\text{-}\mathsf{EXP}^{\mathcal{A}}_{\lambda}(1)\right) = \mathsf{negl}(\lambda),$$

Theorem 1. The scheme CD-SS = (Split, Reconstruct, Delete, Verify) defined as follows is a secret sharing scheme with certified deletion.

- Split(m): Sample $x, \theta \leftarrow \{0, 1\}^{\lambda}$ and output

$$s_1 \coloneqq |x\rangle_{\theta}, s_2 \coloneqq \left(\theta, b \oplus \bigoplus_{i:\theta_i=0} x_i\right), \quad \mathsf{vk} \coloneqq (x, \theta).$$

- Reconstruct (s_1, s_2) : Parse $s_1 \coloneqq |x\rangle_{\theta}, s_2 \coloneqq (\theta, b'), \text{ measure } |x\rangle_{\theta}$ in the θ -
- basis to obtain x, and output $b = b' \oplus \bigoplus_{i:\theta_i=0} x_i$. $\mathsf{Delete}(s_1)$: Parse $s_1 \coloneqq |x\rangle_{\theta}$ and measure $|x\rangle_{\theta}$ in the Hadamard basis to obtain a string x', and output cert := x'.
- Verify(vk, cert) : Parse vk as (x, θ) and cert as x' and output \top if and only if $x_i = x'_i$ for all i such that $\theta_i = 1$.

This is the construction for a 1 bit secret, and to share a m bit secret, we apply this construction to each bit.

4

3 ACD Does Not Imply NSCD

In this section, we present a scheme satisfying ACD but not NSCD. As a proof, we present a (3, 4) TSSS, but the same idea can be applied to wider numbers. Let $BK = (Split_2, Reconstruct_2, Delete_2, Verify_2), A_{ss} = (A.Split, A.Reconstruct, A.Del, A.Ver), and <math>C_{ss} = (C.Split, C.Reconstruct)$ be the secret sharing scheme of Bartusek and Khurama that satisfies certified deletion security [1] (Definition 5, where we take the construction for multiple bit secret), an SSS with adaptive certified deletion, and a classical SSS, respectively. We use subscripts such as $Split_{k,n}$ to denote k-out-of-n secret sharing for A_{ss} and C_{ss} .

Theorem 2. The construction given in Fig. 1 satisfies adaptive but not nosignaling certified deletion security (Definitions 3 and 4).

$\mathsf{Split}(s,\lambda)$:

- 1. Split s through a (3,4) SSS satisfying ACD : $(s_1, s_2, s_3, s_4, \mathsf{vk}_a) \leftarrow \mathsf{A.Split}_{3,4}(s)$
- 2. Split s through a classical (2,2) SSS : $(sh_1, sh_2) \leftarrow \mathsf{C.Split}_{2,2}(s)$.
- 3. For i = 1, 2, split sh_i through $\mathsf{Split}_2 : (|\mathsf{qsh}_i\rangle, \mathsf{csh}_i, \mathsf{vk}_i) \leftarrow \mathsf{Split}_2(sh_i)$.
- 4. Split $t = (\mathsf{csh}_1, \mathsf{csh}_2)$ through a classical (2, 2) SSS: $(t_1, t_2) \leftarrow \mathsf{C.Split}_{2,2}(t)$.
- 5. For $i \in \{1, 2\}$ split t_i through a (2, 2) SSS satisfying ACD: $(t_{i,1}, t_{i,2}, \mathsf{vk}'_i) \leftarrow A.\mathsf{Split}_{2,2}(t_i).$

6. Let $S_1 = (s_1, t_{1,1}, |\mathsf{qsh}_1\rangle); S_2 = (s_2, t_{1,2}, |0\rangle); S_3 = (s_3, t_{2,1}, |\mathsf{qsh}_2\rangle); S_4 = (s_4, t_{2,2}, |0\rangle)$ these are the final shares. We add enough $|0\rangle$ to match the size of $|\mathsf{qsh}_i\rangle$ (i.e 0 written on multiple qbit) for shares 2 and 4.

7. Let $\mathsf{vk} = (\mathsf{vk}_a, \mathsf{vk}_1, \mathsf{vk}_2, \mathsf{vk}_1', \mathsf{vk}_2')$

 $\frac{\text{Reconstruct}(\{S_i\}_{i \in P}): \text{ If } |P| \text{ is less than 3 output } \bot, \text{ else parse the share to get } \\ \hline \text{the } s_i \text{ and use 3 of them to get } s \text{ through A.Reconstruct}_{3,4}.$

 $\frac{\text{Delete}(S_i):}{\text{A.Del}_{3,4} \text{ to } s \text{ and } \text{A.Del}_{2,2} \text{ to } t \text{ to get } cs_i, ct_i. \text{ Apply Delete}_2 \text{ to } |qsh\rangle \text{ to get a certificate } c_i \text{ (if } i \neq 1,3 \text{ then } c_i \text{ meaningless and for size purpose only). Send cert_i = (cs_i, ct_i, c_i).}$

 $\begin{array}{lll} & \underbrace{\mathsf{Verify}(\mathsf{cert}_i,i,\mathsf{vk}):}_{(\mathsf{vk}_a,\mathsf{vk}_1,\mathsf{vk}_2,\mathsf{vk}_1',\mathsf{vk}_2'), \text{ apply } \mathsf{A}.\mathsf{Ver}_{3,4}(\mathsf{cs}_i,i,\mathsf{vk}_a) \text{ and } \mathsf{A}.\mathsf{Ver}_{2,2}(\mathsf{ct}_i,(i\bmod 2)+1,\mathsf{vk}_2'\underbrace{i\pm 1}_2). \text{ If } i \text{ is } 1 \text{ or } 3 \text{ apply } \mathsf{Verify}_2(\mathsf{vk}_{(i+1)/2},\mathsf{c}_i). \text{ If all the former verifications } \\ & \operatorname{are true, output} \top, \text{ else output } \bot. \end{array}$

Fig. 1: Secret Sharing with Adaptive but not No-signaling Certified Deletion security

This scheme's access structure is still (3, 4) as if one has less than 3 shares then t is unknown and so is s. With more than 3 shares, the s_i are sufficient to recreate the secret.

Proof. We will first present a No-signaling attack proving this scheme does not satisfy no-signaling certified deletion before showing why it satisfies adaptive certified deletion.

A No-signaling (Definition 3) attack : Let $P_1 = \{1, 2\}$ and $P_2 = \{3, 4\}$. We describe \mathcal{A}_1 :

- 1. Parse S_1, S_2 to get $(s_1, s_2, t_{1,1}, t_{1,2}, |\mathsf{qsh}_1\rangle)$.
- 2. Apply $\text{Reconstruct}(t_{1,1}, t_{1,2})$ to get t_1 , this does not affect $t_{1,1}, t_{1,2}$.
- 3. S_2 is still intact, and we apply $\mathsf{Delete}(S_2)$ to obtain a certificate that we give.

 \mathcal{A}_2 works the same way. After computing \mathcal{A}_1 and \mathcal{A}_2 , we have deleted two shares and are left with only two non-deleted ones, which is an unauthorized set.

At the end of the experiment, we have $(t_1, t_2, |\mathsf{qsh}_1\rangle, |\mathsf{qsh}_2\rangle)$. We use t_1, t_2 to reconstruct $t = (\mathsf{csh}_1, \mathsf{csh}_2)$. We use $\mathsf{csh}_i, |\mathsf{qsh}_i\rangle$ to get sh_i with Reconstruct₂. We now have sh_1, sh_2 and therefore s which leads to a security break. So this scheme does not satisfy NSCD.

Adaptive Security (Definition 4): We remind (Definition 5) the security of the 2-out-of-2 secret sharing scheme that implies that an adversary given $|qsh_i\rangle$ that produces a certificate for $|qsh_i\rangle$ without knowing csh_i cannot gain knowledge of sh_i even if it were to gain that information afterward. That being said, the adaptive security of the scheme used to create the s_i means that we only have to worry about information being revealed by sh_i and, therefore, by t_i .

Note that the adversary cannot break the (3, 4) SSS with ACD part by the definition. Hence, we can focus on the other parts by (2, 2) SSS, (2, 2) SSS with ACD, and BK. If an adversary does not corrupt all the shares $(C = \{1, 2, 3, 4\}$ at the end of the experiment), then either t_1 or t_2 is fully unknown (because one of $t_{i,j}$ has never been seen) and therefore s is unknown too. Hence, an adversary would have to have $C = \{1, 2, 3, 4\}$ at the end of the experiment. Note that the adversary cannot corrupt three or four shares without outputting valid certificates since we consider a 3-out-of-4 TSSS. Hence, when he corrupts his last share c_4 he has already made two valid deletion certificates d_1, d_2 . We have two cases.

If one of (d_1, d_2) is a certificate for S_1 or S_3 , then at the time of deletion, we had $C \neq \{1, 2, 3, 4\}$. Hence, $t = (\mathsf{csh}_1, \mathsf{csh}_2)$ is fully unknown when the adversary had to produce a deletion for $|\mathsf{qsh}_1\rangle$ or $|\mathsf{qsh}_2\rangle$. Hence, we can apply the security of BK (either $|\mathsf{qsh}_1\rangle$ or $|\mathsf{qsh}_2\rangle$ was deleted) and obtain that sh_1 or sh_2 is fully unknown even if later csh_1 or csh_2 is leaked. Thus, s is safe due to the security of C_{ss} .

Finally, if (d_1, d_2) are certificates for S_2 and S_4 , then c_4 is 1 or 3. We consider the case where c_4 (the fourth corrupted share) is 1. Then, at the time of deletion of $t_{1,2}$, $t_{1,1}$ is unknown. Hence, by the security of A_{ss} , t_1 is also unknown, and

 $\mathbf{6}$

so is $t = (csh_1, csh_2)$ due to the security of C_{ss} . Thus, s is also unknown due the security of BK and C_{ss} . The same applies when c_4 is 3, and t_2 is hidden. Thus, the scheme satisfies ACD.

This works as a testimony to prove that no-signaling adversaries have some power that adaptive adversaries do not have.

References

- Bartusek, J., Khurana, D.: Cryptography with certified deletion. In: Handschuh, H., Lysyanskaya, A. (eds.) CRYPTO 2023, Part V. LNCS, vol. 14085, pp. 192–223. Springer, Cham (Aug 2023). https://doi.org/10.1007/978-3-031-38554-4_7 1, 4, 5
- Bartusek, J., Raizes, J.: Secret sharing with certified deletion. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part VII. LNCS, vol. 14926, pp. 184–214. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68394-7_7 1, 2
- Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: IEEE International Conference on Computers Systems and Signal Processing. pp. 175–179. IEEE (1984) 2, 4
- Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery 22(11), 612–613 (Nov 1979). https://doi.org/10.1145/359168. 359176 1