ADC-BE: Optimizing Worst-Case Bandwidth in Broadcast Encryption with Boolean Functions

Yadi Zhong

Department of Electrical and Computer Engineering, Auburn University,
Auburn AL 36849, USA
yadi@auburn.edu

Abstract. Recently, Dupin and Abelard [11] proposed a broadcast encryption scheme, outperforming the Complete Subtree-based and Subset Difference broadcast encryption in terms of encryption cost and bandwidth requirement. However, Dupin and Abelard acknowledge that the worst-case bound for bandwidth requirement of Complete Subtree approach can be reached in their scheme as well. In this paper, we answer the call to further reduce this bandwidth bottleneck. We first provide concrete analysis to show how this worst-case upper-bound is reached from particular Boolean functions. Our proof on [11]'s worst-case complexity demonstrates and shows that those Boolean functions, out of all 2^{2^ℓ} possible Boolean functions of n variables, produces the worst-case upper-bound. Then we present two improved broadcast encryption schemes to significantly reduce this worst-case bandwidth consumption for further optimization of Dupin and Abelard's technique. Our proposed ADC-BE suite, composed of two primitives, AD-BE and AC-BE, can significantly optimize this worst-case complexity from $\mathcal{O}(n/2)$ down to $\mathcal{O}(1)$ for a system of n users. This is efficient especially for large number of users in the system. Our schemes combines the algebraic normal form, disjunctive normal form, and conjunctive normal form to optimize a Boolean function to its minimized representation. Furthermore, our approaches support full range of Boolean functions without any restrictions, i.e., general access structures. This is more versatile than monotone access structure, where the group attribute policy is solely restrained to the Boolean clauses without bit-inversion nor negation. Last but not least, both AD-BE and AC-BE can be straightforwardly extended to post-quantum broadcast encryption schemes as the underlying hardness depends on one-way function, e.g., secure hash function.

Keywords: Broadcast encryption · Boolean functions · bandwidth optimization · general access structure · one-way function · post-quantum

1 Introduction

Fiat and Naor [12] first introduced the concept of broadcast encryption in their seminal work of 1993. A broadcaster encrypts a message for a set of users to

decrypt it, called authorized users, but revoked users should obtain no information about message m. It has potentials in applications such as satellite TV services, where different services are provided to various user groups due to their subscriptions and validity period across the time domain. These user groups varies as time progresses, i.e., cancellation or change of service. The revoked user list is not static, but is dynamically updated with recently added and/or removed customers. Broadcast encryption is a special case from ciphertext policy attribute-based encryption (CP-ABE), where the predicate is the authorized user list. Over the year, additional properties for broadcast encryption has been considered. Traitor-tracing [10,14,20] allows authorities to trace to the source (or users) of any compromised cryptographic keys for copyright protections against any potentially unauthorized distributions of contents such as DVDs. Decentralized broadcasting mechanisms [8, 17], preventing single points of failure, have been analyzed extensively in recent years. Broadcast encryption can be built from various assumptions, such as Subset-Cover paradigm [2–4, 20], layered Subset Difference [16], bilinear groups and pairings [6, 9, 17, 18], decisional bilinear Diffie-Hellman assumption [22], learning-with-errors and lattices [1, 26, 27], and k-Lin [13, 25], etc. Quantum resistant broadcast encryptions require large ciphertext size, which are one to two orders of magnitude bigger in size than Dupin and Abelard [11]. However, those based on number theoretic assumptions or structured hardness assumptions involve considerable computation of elliptic curve points, and/or pairings, compared to Dupin and Abelard [11]. In addition, these schemes, as they are built from public-key constructions, are vulnerable to quantum attacks under large-scale cryptanalytic-relevant quantum computers, due to Shor's algorithm [24].

Recently, Dupin and Abelard [11] proposed a broadcast encryption with sum-product decomposition from Boolean functions. It optimizes the number of encryptions needed to broadcast a message m to authorized users only while simultaneously achieving full collusion resistance against any set of revoked users. It relies on Boolean optimizations, e.g., Quine-McCluskey algorithm and Petrick's method [21], to derive the minimized Boolean function, which in turn affects the number of encryptions in this broadcast encryption scheme. It is built on a minimal assumption with one-way function only. Dupin and Abelard's proposal of broadcast encryption from sum-product decomposition (Section 6.2 of [11]) outperforms subset difference-based broadcast encryption by Naor, Naor, and Lotspiech [20] in terms of encryption cost and bandwidth requirement.

1.1 Contributions

In this paper, we provide a theoretical analysis on the worst-case bandwidth complexity of Dupin and Abelard [11], a Boolean function decomposition-based broadcast encryption with n users. We identify from the theoretical standpoint two *hard functions* on which Boolean optimizations cannot be performed. We then propose ADC-BE, Algebraic, Disjunctive, Conjunctive normal forms-based Broadcast Encryptions suite, composed of two broadcast encryption algorithms, AD-BE and AC-BE, where both schemes optimize the worst-case complexity of

$\mathcal{O}(n/2)$ from [11] to constant $\mathcal{O}(1)$. The hardness assumption of AD-BE and AC-BE is based on one-way function, i.e., secure hash function. Both proposed constructions admit arbitrary access structures while achieving full collusion-resistance against any number of revoked users. Our contributions of the paper are as follows:

- We first identify two *hard functions* for any n Boolean variables, where both cannot be simplified by postulates or theorems in Boolean algebra. Both function have the most number of monomials in disjunctive normal form (DNF) representations (or OR clauses in conjunctive normal form (CNF)).
- We show that the worst-case bandwidth complexity of [11] is reached with $r = n/2$ revoked users whose authorized user list is either of the *hard functions* analyzed in Theorem 1. This is due to broadcast encryption applying solely the DNF representation for minimized Boolean function. We also demonstrate that the identical issue occurs for CNF representation.
- We demonstrate how this worst-case complexity can be drastically reduced when introducing a combined representation from DNF or CNF with algebraic normal form. We propose two approaches, AD-BE and AC-BE. Both proposed broadcast encryption schemes can significantly optimize this worst-case complexity from $\mathcal{O}(n/2)$ down to $\mathcal{O}(1)$ for a system of n users (see Table 1 for comparison). In addition, perfect correctness and full collusion-resistance against any number of revoked entities are achieved.
- Furthermore, our construction of ADC-BE suite for broadcast encryption allow general access structures for authorized client list. Our approaches support full range of Boolean functions in the complete Boolean function class without restrictions. Broadcaster can send targeted messages for any arbitrary combination of selected/authorized clients. This is build upon one-way function, e.g., secure hash function.
- Lastly we provide a extension for the proposed AD-BE and AC-BE in order to transform both algorithms to withstand quantum adversaries. This adaption make them possible as post-quantum broadcast encryption.

1.2 Paper Organization

The paper is organized as follows. In Section 2 we provide preliminaries and definitions. Section 3 describes our analysis of the worst-case complexity of broadcast encryption construction proposed by Dupin and Abelard and provide *hard functions* where the bandwidth upper-bound is reached. Our solutions can reduce this bandwidth bottleneck and are presented in Section 4. It also describes the adaption of AD-BE and AC-BE schemes to quantum-resistant broadcast encryptions. We conclude this paper in Section 5.

2 Preliminaries

Notations. We say a function $\epsilon : \mathbb{N} \to \mathbb{R}$ is negligible in the parameter λ if $\epsilon(\lambda) = o(1/p(\lambda))$ for every positive polynomial $p(\cdot)$. We use $e \xleftarrow{\$} \mathcal{S}$ to denote

uniform sampling of an element e from set $\mathcal{S}$. The set of integers $\{a, \ldots, 1\}$ for $a \in \mathbb{N}$ is denoted as $[a]$. Vector $v_{[\ell]}$ denotes all ℓ variables $\{v_\ell, \ldots, v_1\}$. Concatenation of binary strings a, b is represented as $a||b$. Multiple binary strings $s_i \in \mathbb{F}_2^n, i \in [\ell]$ can be combined into one big string $s = \{\|_i s_i\}$ of length ℓn bits. For any Boolean variable v, we use v^{-1} and $\overline{v}$ interchangeably to represent the complement of v.

2.1 Assumptions

Definition 1 (One-Way Function (OWF)). *A function $f : \{0,1\}^* \to \{0,1\}^*$ is a one-way function if:*

1. *(Easy to compute) There exists a polynomial-time algorithm M_f computing f such that $M_f(x) = f(x)$ for all x.*
2. *(Hard to invert) For every probabilistic polynomial-time algorithm $\mathcal{A}$, there is a negligible function $\mathsf{negl}(\cdot)$ such that*

$$\mathsf{Pr}[\mathsf{Invert}_{\mathcal{A},f}(\lambda) = 1] \leq \mathsf{negl}(\lambda),$$

where $\mathsf{Invert}_{\mathcal{A},f}(\lambda)$ is the probabilistic polynomial-time experiment to invert f for security parameter λ.

Definition 2 (Pseudorandom function (PRF)). *An efficiently computable, keyed function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ with key-space $\mathcal{K}$, domain $\mathcal{X}$ and range $\mathcal{Y}$ is a pseudorandom function if for all PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ such that*

$$\Big|\mathsf{Pr}[k \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{F_k(\cdot)}(1^\lambda) = 1] - \mathsf{Pr}[f \xleftarrow{\$} \mathsf{Func}(\mathcal{X}, \mathcal{Y}) : \mathcal{A}^{f(\cdot)}(1^\lambda) = 1]\Big| \leq \mathsf{negl}(\lambda).$$

Our proposed schemes AD-BE and AC-BE depend on OWF, i.e., secure hash function. Both AD-BE and AC-BE achieve beyond the results of the previous work of [11], can be adapted to resist quantum attackers (see Section IV-4.4).

2.2 Broadcast Encryption

We adopt the same notion of broadcast encryption as in Dupin and Abelard [11]. It consists of four algorithms, $\mathsf{Setup}(\cdot), \mathsf{KeyGen}(\cdot), \mathsf{Enc}(\cdot), \mathsf{Dec}(\cdot)$, as follow:

- $\mathsf{Setup}(n, 1^\lambda) \to mk$: With security parameter of unary string 1^λ, and number of users n (or an upper-bound), the broadcaster generates the master key mk.
- $\mathsf{KeyGen}(u, mk) \to k_u$: The sender computes the key material k_u for user u.
- $\mathsf{Enc}(m, \mathcal{AU}, mk) \to (h, c)$: With key mk, the sender encrypts message m where authorized users in list $\mathcal{AU}$ can decrypt but the revoked users cannot. It produces (h, c), a tuple of ciphertext c and header h, containing instructions for decryption for $\mathcal{AU}$.

– $\mathsf{Dec}(h, c, k_u) \rightarrow m$ else $\perp$: Receiver u utilizes header h to check if belongs to list $\mathcal{AU}$ or not. If user $u \in \mathcal{AU}$, it will be able to decrypt c to recover message m. Otherwise, a revoked user cannot decrypt nor determine the original message m, and therefore the decryption algorithm terminates with $\perp$.

It is worth noting that the list of authorized users $\mathcal{AU}$ can vary for one ciphertext to another, accommodating various real-life scenarios, i.e., subscription expires for some member and/or new members join the program. Thus, organizer keeps track of the current authorized users (and/or specific to a particular subscription), and broadcasts ciphertext corresponding to this set of valid users. Broadcast encryption can be implemented with either public key encryption, or symmetric key encryption, which is the approach adopted by Dupin and Abelard [11].

In this paper, we consider the entity which holds the master key mk as broadcaster. It is also interchangeably denoted as emitter, sender, (central) authority, organizer. On the other hand, users belonging to the broadcast group are also called recipients, entities, or receivers in this paper.

2.3 Boolean Function

Boolean function are the core for our proposed broadcast encryptions. We describe its definition and properties, then proceed with various representation forms.

Definition 3 (Boolean Function). *A Boolean function of n Boolean variables $x = [x_n, x_{n-1}, \ldots, x_1]$ is a function $f(x) : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$. The binary vector $v_f = \{f(x)|x \in \mathbb{F}_2^n\}$ of length 2^n is called value vector of $f(x)$.*

Note that inside a Boolean function it is possible to allow *don't care* values, represented as $*$. Depending on the minimization technique targeting on desired reduction format (e.g., CNF or DNF), it can be interpreted as either elements in $\mathbb{F}_2$. If a Boolean function includes *don't cares*, it is also reflected in its value vector v_f.

Definition 4 (Support). *The support of a Boolean function $f(x) : \mathbb{F}_2^n \rightarrow \mathbb{F}_2$ is the support of its v_f, $s_f = \{x \in \mathbb{F}_2^n | f(x) = 1\}$.*

Definition 5 (Weight). *The weight of a Boolean function $f(x)$ is the weight of its value vector, $wt(f) = wt(v_f)$.*

Definition 6 (Balanced). *A Boolean function $f(x)$ is denoted as balanced if its weight $wt(f) = 2^{n-1}$. In other words, function f is uniformly distributed.*

Although any Boolean function can be represented by its truth table, there are several succinct ways to equivalently describe it. Algebraic normal form (ANF) considers a Boolean function f as a multivariate polynomial,

$$f(x) = \sum_{v \in \mathbb{F}_2^n} e_v x^v,$$

where coefficient $e_v \in \mathbb{F}_2$, and x^v denotes the monomial in $\mathbb{F}_2[x_n, \ldots, x_1]/(x_n^2 + x_n, \ldots, x_1^2 + x_1)$. Conjunctive normal form (CNF) and disjunctive normal form (DNF), commonly used in very large scale integration (VLSI) design and test, considers the logical OR and AND operations (together with NOT operator) to represent a Boolean function [28–30]. CNF expression, the ANDing of summation (OR) of input literals, is written as

$$f(x) = \bigwedge_{S_i \in S} \left(\bigvee_{v_j \in S_i} x_j^{v_j} \right);$$

The DNF representation of Boolean function, on the other hand, combines AND product terms through OR function,

$$f(x) = \bigvee_{S_i \in S} \left(\bigwedge_{v_j \in S_i} x_j^{v_j} \right),$$

where $v_j \in \{-1, +1\}$ with $x_j^{+1} = x_j$ and x_j^{-1} denoting the complement of input x_j, $x_j^{-1} = \overline{x_j}$. Literals in set S_i is a subset of all n-bit input $\{x_1, x_2, \ldots, x_n\}$, depending on Boolean optimizations. For CNF, AND clauses $\left(\bigwedge_{v_j \in S_i} x_j^{v_j}\right)$ are logically combined with OR function. Conversely, OR clauses $\left(\bigvee_{v_j \in S_i} x_j^{v_j}\right)$ are ANDed together to construct DNF of Boolean function. Boolean function in DNF format is used in [11].

We would like to point out that the representation of a Boolean function f in ANF is fixed, as coefficients a_v of ANF can be uniquely determined by $a_v = \sum_{x \preceq v} f(x)$ of binary Möbius transform from its value vector v_f. However, the representation may not unique in CNF nor DNF constructions. This is due to the possibility of either incomplete minimization under Boolean algebra, or multiple optimal solutions existe from algebraic simplification methods.

2.4 Quine-McCluskey Method

We briefly describe the core concept of minimizing Boolean functions of DNF format with Quine-McCluskey method, where detailed descriptions can be found in [19, 21]. The output from Quine-McCluskey has the minimal number of AND products in the OR expression as well as minimal input literals in each AND function. This same approach can also be applied to determine the reduced Boolean expression in CNF representation, with additional transformation through De-Morgan's theorems. In addition, Quine-McCluskey algorithm supports the minimization of incompletely specified functions, with don't cares included in value vectors.

The overarching goal of Quine-McCluskey is to find the optimized Boolean expression (or one of the optimized expression) of a two-level AND-OR circuit such that the number of logic gates is minimal as well as the number of inputs of

each gate. It is a tabular approach to minimize any Boolean function through: (*i*) an ordered search over minterms of a Boolean function to identify all its prime implicants; (*ii*) a minimal cover of prime implicants is the solution to smallest (minimal) sum-of-products. This process can be summarized in four steps:

1. On initial column, list all minterms (rows in truth table with output 1) and sort it according to the number of input 1s. Two minterms are logically adjacent, and therefore can be further combined together, if they differ in exactly one literal only.
2. Perform a search for groups of adjacent minterms and reduce them to $(n-1)$-variable implicants with a dash on the eliminated variable. The $(n-1)$-variable implicants are recorded on the 2nd column. From these implicants, combine adjacent ones into $(n-2)$-variable implicants and list in 3rd column, and so on. This process is complete until no further implicants can be grouped, and output prime implicants.
3. List minterms (columns) and prime implicants (rows) with a prime implicant chart. An entry is checked-off if the corresponding prime implicant covers a minterm.
4. Choose a minimum number of prime implicants to cover all minterms of the Boolean function. Then use Petrick's method to derive minimal solution [23].

There are other techniques to optimize a Boolean function, especially in integrated circuit design, where efficiencies in power, performance, and area are critical factors leading to the actual fabrication of chips. ESPRESSO algorithm [7] is an alternative to Quine-McCluskey method in finding the minimized Boolean expression. Both optimizations are based on the common postulates and theorems in Boolean algebra. Thus, we omit the details for ESPRESSO in this paper.

3 Complexity Analysis of Dupin and Abelard

In this section, we first describe broadcast encryption paper by Dupin and Abelard [11]. We proceed to discuss its bandwidth complexity by introducing additional definitions on the degree of optimizations of Boolean functions. The worst-case bandwidth complexity of their algorithm is then identified with induction proof.

3.1 Broadcast Encryption Scheme of Dupin and Abelard

Dupin and Abelard proposed a collusion-resistant broadcast encryption scheme requiring only pseudorandom function F. It uses two symmetric encryption schemes $\mathcal{E}$ and $\mathcal{E}_{\mathrm{payload}}$, where $\mathcal{E}_{\mathrm{payload}}$ can be an authenticated encryption. Their corresponding decryption algorithms are $\mathcal{D}, \mathcal{D}_{\mathrm{payload}}$.

In this section, we consider n, the number of users in the broadcast system as a power of 2, with $l = \log_2(n)$. In practice, variable n may be selected to be the next power of 2 such that l is always an integer.

Setup. Emitter first generates master key k_{PRF} from pseudorandom function F. As each user can be uniquely identified with a l-bit binary, emitter generates $2l$ distinct labels, $\{k_i^0\}_{i\in[l]}, \{k_i^1\}_{i\in[l]}$. It can be implemented as concatenation of binary bits to define labels $k_i^j = i||j, i \in [l], j \in \{0,1\}$.

For example [11], emitter will generate 6 labels under $n = 8, l = 3$, $\{k_1^0, k_1^1, k_2^0, k_2^1, k_3^0, k_3^1\}$.

Key Generation. In order to add a user u to the broadcast system, authority provides a set of keys $k_u \leftarrow F_{k_{PRF}}(||_{i\in\mathcal{S}} k_i^{u_i})$ to it based on the binary representation of $u = u_1||u_2||\ldots||u_{l-1}||u_l$ under the ordered lexicographic list $\mathcal{S} \subset [l]$.

User $u = 5 = 101_2$ from $n = 8$ would receive the following key materials in k_u:

$$\begin{array}{lll} F_{k_{PRF}}(k_1^1); & F_{k_{PRF}}(k_2^0); & F_{k_{PRF}}(k_3^1); \\ F_{k_{PRF}}(k_1^1||k_2^0); & F_{k_{PRF}}(k_1^1||k_3^1); & F_{k_{PRF}}(k_2^0||k_3^1); \\ F_{k_{PRF}}(k_1^1||k_2^0||k_3^1); & F_{k_{PRF}}(\varnothing). & \end{array}$$

Encrypt. Central authority (aka the emitter) first identifies the list of valid users before broadcasting message m. To transmit a message m so that the designated users can read but not the unauthorized ones, the emitter encrypts m, a ephemeral key k_e with $\mathcal{E}_{\text{payload}}(k_e, m)$. Key k_e can be derived exclusively by authorized users. This list of authorized user $\mathcal{AU} \subset [n]$ is characterized as a Boolean function y,

$$y(u) = \begin{cases} 1, & \text{if } u \in \mathcal{AU} \\ 0, & \text{if } u \notin \mathcal{AU} \end{cases}$$

With standard minimization techniques in Boolean algebra, i.e., Quine-McCluskey method, central authority obtains the minimal DNF representation of function y. This optimal representation of function y is encoded in header h. Ciphtertext c contains $\mathcal{E}_{\text{payload}}(k_e, m)$ and multiple encryptions of ephemeral key k_e under valid key materials (see details below) for authorized users in $\mathcal{AU}$. If message m is transmitted to its intended user z, the following properties are guaranteed: *(i)* the evaluation of function y on input z is $y(z) = 1$; *(ii)* header h includes at least one binary vector corresponding to the binary pattern of key labels user z received from $\mathsf{KeyGen}(\cdot)$; *(iii)* ciphertext c has at least one encryption(s) of ephemeral key k_e which the user z possesses the corresponding encryption key, and is therefor able to correctly decrypt the key material k_e; and then decrypt $\mathcal{E}_{\text{payload}}(k_e, m)$ with k_e to obtain message m. On the other hand, if user z is revoked, or he/she simply it does not belong to list $\mathcal{AU}$, then *(i)* function y on input z has output $y(z) = 0$ instead; *(ii)* header h does not contain any binary patterns of key labels which user z received from the previous step $\mathsf{KeyGen}(\cdot)$; *(iii)* ciphertext c has no encryption of ephemeral key k_e that user z matching the key materials from $\mathsf{KeyGen}(\cdot)$; therefore z could not decrypt c to get k_e; and *(iv)* therefore, he/she cannot obtain k_e nor message m. Both h and c are sent to all the recipients.

Continuing the example described in Key Generation, suppose broadcaster would like to send a targeted message m to users $1 = 001_2, 2 = 010_2, 5 =$

$101_2, 7 = 111_2$ while users $0, 3, 4, 6$ are unauthorized to view. Equivalently, we need to find the optimal Boolean function y with logic 1 output for users $1, 2, 5, 7$ but 0 output for the rest is expressed as

$$y = \overline{u_1}u_2\overline{u_3} \vee u_1u_3 \vee \overline{u_2}u_3.$$

For any AND product in y, it is encoded in the following manner with a $2l$-bit binary, where a l-bit binary for complemented input variables and another l-bit binary for input variables in their true forms. For example, AND gate of $\overline{u_1}u_2\overline{u_3}$ is encoded as

$$\overline{u_1}u_2\overline{u_3} \rightarrow \overbrace{101}^{\overline{u_1},\overline{u_3}}\overbrace{010}^{u_2}$$

Header h encodes y's three AND products in binary format with an extra l-bit to indicates the number of ANDs in y,

$$\overline{u_1}u_2\overline{u_3} \vee u_1u_3 \vee \overline{u_2}u_3 \rightarrow h = \overbrace{011}^{3}\overbrace{101\ 010}^{\overline{u_1}u_2\overline{u_3}}\overbrace{000\ 101}^{u_1u_3}\overbrace{010\ 001}^{\overline{u_2}u_3}.$$

Ciphertext c is then computed as

$$c = \overbrace{\mathcal{E}_{k_{h_1}}(k_e)||\mathcal{E}_{k_{h_2}}(k_e)||\mathcal{E}_{k_{h_3}}(k_e)}^{3 \text{ encryptions of } k_e}||\mathcal{E}_{\text{payload},k_e}(m)$$

Symmetric keys used for encrypting ephemeral key k_e are $k_{h_1} = F_{k_{PRF}}(k_1^0||k_2^1||k_3^0), k_{h_2} = F_{k_{PRF}}(k_1^1||k_3^1), k_{h_3} = F_{k_{PRF}}(k_2^0||k_3^1)$. Any user can check whether it has one or more, or none of symmetric keys in their storage based on information from the header h. Users who are the intended recipients of this message m should have at least one of the symmetric keys, obtained during $\mathsf{KeyGen}(\cdot)$, valid for decrypting the ciphertext c to get k_e. Therefore, a valid user from set $\{1, 2, 5, 7\}$ is able to successfully decrypt one (or at least one) of the 3 encryptions of k_e and use it to further obtain m from $\mathcal{E}_{\text{payload},k_e}(m)$. It can be verified that none of these symmetric keys are from the key materials of any revoked users. In addition, even when any subset or all of revoked users collude, they do not have the valid key materials for decryption. Therefore, it achieves full collusion-resistance against any number of revoked clients.

Decrypt. When recipient u receives {h,c}, he/she first checks whether $y(u) = 1$ from binary vectors in header h. If $y(u) = 0$ instead, user u does not belong to the authorized user list $\mathcal{AU}$, then decryption process terminates. Otherwise, the intended recipient u finds a symmetric key based of h from stored key materials of $\mathsf{KeyGen}(\cdot)$, where at least one symmetric key must be in possession of such valid users. Suppose that receiver has k_{h_i} in the stored key list, ephemeral key k_e can be then recovered from decryption $\mathcal{D}_{k_{h_i}}(\mathcal{E}_{k_{h_i}}(k_e))$. Finally, message m is extracted from $\mathcal{D}_{\text{payload},k_e}(\mathcal{E}_{\text{payload},k_e}(m))$. For users in list $\mathcal{AU}$, executing this decryption procedure has perfect correctness.

3.2 Worst-Case Bandwidth for Broadcast Encryption using Sum-Product Decomposition

Dupin and Abelard's broadcast encryption from sum-product decomposition (Section 6.2 of [11]) outperforms subset difference-based broadcast encryption by Naor, Naor, and Lotspiech [20] in terms of bandwidth requirement. As Dupin and Abelard pointed out, this bandwidth is dominated by the number of encryptions of ephemeral key k_e inside ciphertext c during $\mathsf{Enc}(\cdot)$ step. We would like to note that the root cause belongs to number of AND clauses within the minimal DNF expression of y, the Boolean function that encodes authorized user list $\mathcal{AU}$, which dictates ciphertext c length. Likewise, we would like to point out that this bandwidth is also affected by the length of header h, the other component from output tuple of $\mathsf{Enc}(\cdot)$ procedure, as its length grows than the number of AND clauses of y. Thus, we are able to conclude that bandwidth complexity of [11] is solely reliant on the optimal representation of Boolean function y. We summarize it in the following remark.

Remark 1. The communication bandwidth of $\mathsf{Enc}(\cdot)$ in Dupin and Abelard's broadcast encryption scheme is determined by number of products in Boolean function y, aka the binary encodings for list $\mathcal{AU}$. As y is the minimal expression of list $\mathcal{AU}$, optimal bandwidth $\mathsf{Enc}(\cdot)$ relies on the optimal representation of Boolean functions.

The sum-product decomposition approach is heavily reliant on the algebraic structure of DNF representation of Boolean functions for its minimization of y. To better understand the worst-case bandwidth of their scheme, we will first discuss theoretical worst-case scenarios in minimization of arbitrary Boolean expressions targeting DNF and CNF (for proposed scheme in Section 4.2) formats, and introduce additional definitions to accurately characterize and compare any optimized function from its original expression.

To better analyze optimized Boolean functions derived from optimization techniques, i.e., Quine-McCluskey and ESPRESSO methods, we introduce the concept of compression ratios, defined as follow, for all minimized expressions.

Definition 7 (Compression Ratio ($\mathbf{CR_{DNF}}$)). *The compression ratio* $\mathsf{cr}_1(f)$ *of a simplified Boolean function* f *is defined as the* $wt(v_f)$ *(number of outputs 1s in the truth table) divided by total number of monomials in* f *in its minimized form.*

Note that this value is $\mathsf{cr}_1(f) \geq 1$ for any Boolean function f. It reaches the lower bound $\mathsf{cr}_1(f) = 1$ when its minimized expression is identical to the un-optimized format – the canonical sum-of-products (CSOP) expression (when optimization techniques cannot reduce f). This is equivalent to having $wt(v_f)$ number of AND clauses in its minimized form. For constant-1 function, we set its compression ratio to $\mathsf{cr}_1 = 2^n$ since its Boolean function has $wt = 2^n$ under n-bit input, which can be considered as a degenerated monomial.

We then extend this definition from optimal representation in DNF format to CNF. This is relevant to our proposed algorithm of Section 4-4.2. Unlike DNF expressions, CNF is distinguished by output 0s instead.

Definition 8 (Compression Ratio ($\mathbf{CR_{CNF}}$)). *Compression ratio* $\mathsf{cr_0}(f)$ *of a simplified Boolean function* f *is defined as the* $\overline{wt} = wt(\overline{v_f})$ *divided by total number of OR clauses in* f *in its minimized form.*

Again, this value is lower-bounded by $\mathsf{cr_0}(f) \geq 1$ for any Boolean function f. It is possible to reach the smallest value $\mathsf{cr_0}(f) = 1$ when its minimized CNF expression is identical to its corresponding un-optimized format – the canonical product-of-sums (CPOS) expression (when optimization techniques cannot reduce f). This is identical to having $wt(\overline{v_f})$ number of OR clauses in its reduced expression. For the constant-zero function, as it has $\overline{wt} = 2^n$ for n inputs, we consider its compression ratio as $\mathsf{cr_0} = 2^n$.

Although minimization techniques like Quine-McCluskey and ESPRESSO allow substantially reduction in the number of AND, OR operations, they all are based on the postulates and theorems of Boolean algebra. If no postulates nor theorems in Boolean algebra can be applied to minimize a particular Boolean function h, then it reaches $\mathsf{cr_1}(f) = 1$ (or $\mathsf{cr_0}(f) = 1$, respectively). This is where the most lengthy and costly CSOP (or CPOS) expression of Boolean function h is also its reduced form and its optimal solution.

Therefore, we are interested to examine the Boolean function subspace of n input variables such that any function inside this subspace has common property of its CSOP (or CPOS) expression being the minimized solution. Then, one of the natural question to ask is: What is the largest weight wt_{max} (or $\overline{wt}_{max}$) of this Boolean function subspace with $cr_1 = 1$ (or $cr_0 = 1$)? What are the characteristic of these Boolean functions with largest weight wt_{max} (or $\overline{wt}_{max}$)? We denote any Boolean function which satisfies $cr_1 = 1$ (or $cr_0 = 1$) with wt_{max} (or $\overline{wt}_{max}$) as hard function in Boolean simplification, which also is accounted for the worst-case bandwidth of [11]. This hard function with worst-case compression ratio of $\mathsf{cr_1} = 1$ (or $\mathsf{cr_0}(f) = 1$, respectively) and wt_{max} (or $\overline{wt}_{max}$) is illustrated in Examples 1-3 below, and is subsequently proved with Theorem 1.

Example 1. For $n = 2$ Boolean variables $[x_2, x_1]$, let us consider Boolean function subspace with $cr_1 = 1$. Any value vector v_h of function h with a single logic 1 as output has $\mathsf{cr_1} = 1$. There are four of those functions, namely $v_h = \{1,0,0,0\}, \{0,1,0,0\}, \{0,0,1,0\}, \{0,0,0,1\}$. For weight $wt = 2$, Boolean functions f_2, g_2 of $\mathsf{cr_1} = 1$ are defined with the following truth tables:

x_2	0	0	1	1
x_1	0	1	0	1
$f_2(x_2, x_1)$	0	1	1	0
$g_2(x_2, x_1)$	1	0	0	1

Under DNF format, $f_2 = \overline{x_2}x_1 \vee x_2\overline{x_1}$ and $g_2 = \overline{x_2}\ \overline{x_1} \vee x_2x_1$. Both functions f_2, g_2 are already in the reduced form where the number of AND clauses is identical to the hamming weight. Thus, compression ratio cr for both functions is $\mathsf{cr_1}(f_2) = \mathsf{cr_1}(g_2) = 1$.

Other functions of $wt = 2$ all have $\mathsf{cr_1} > 1$, whose value vectors are $\{1,1,0,0\}$, $\{1,0,1,0\}$, $\{0,1,0,1\}$, and $\{0,0,1,1\}$. In addition, all possible functions with

$wt = 3$ has to be either of these following four value vectors, $\{1, 1, 1, 0\}$, $\{1, 1, 0, 1\}$, $\{1, 0, 1, 1\}$, or $\{0, 1, 1, 1\}$. All four functions has the minimized DNF structure of 2 AND clauses only, with $\mathsf{cr}_1 = 3/2 > 1$. There exists no Boolean function which simultaneously satisfies $wt = 3$ and $\mathsf{cr}_1 = 1$. Likewise, Boolean function of $wt = 4$ (constant 1 output) cannot be $\mathsf{cr}_1 = 1$ either. Therefore, this Boolean function subspace with $\mathsf{cr}_1 = 1$ of $n = 2$ has functions f_2 and g_2 reaching $wt_{max} = 2$, and they are the *hard function* for $\mathsf{cr}_1 = 1$ under $n = 2$.

The same can be applied to minimized CNF expressions with Boolean function subspace with $\mathsf{cr}_0 = 1$. Function h of $\mathsf{cr}_0 = 1$ with $\overline{wt} = 1$ belongs to either $v_h = \{1, 1, 1, 0\}$, $\{1, 1, 0, 1\}$, $\{1, 0, 1, 1\}$, or $\{0, 1, 1, 1\}$. For weight $\overline{wt} = 2$, only Boolean functions f_2, g_2, as defined above, have $\mathsf{cr}_0 = 1$ with optimal CNF expressions $f_2 = (x_2 \vee x_1)(\overline{x_2} \vee \overline{x_1})$, $g_2 = (x_2 \vee \overline{x_1})(\overline{x_2} \vee x_1)$. Similarly, other functions of $\overline{wt} = 2$ all have $\mathsf{cr}_0 > 1$. Any function with $\overline{wt} = 3$, i.e., whose truth tables are $\{1, 0, 0, 0\}, \{0, 1, 0, 0\}, \{0, 0, 1, 0\}$, or $\{0, 0, 0, 1\}$, has $\mathsf{cr}_1 = 3/2 > 1$; and function of $\overline{wt} = 4$ has $\mathsf{cr}_0 = 4 \neq 1$. Again, functions f_2, g_2 have $\overline{wt}_{max} = 2$ for Boolean function subspace with $\mathsf{cr}_0 = 1$, making them the *hard functions* for $\mathsf{cr}_0 = 1$ under $n = 2$.

Example 2. For $n = 3$ Boolean variables $[x_3, x_2, x_1]$, it can be shown that only two *hard functions* f_3, g_3 achieve $wt_{max} = 4$ and also $\overline{wt}_{max} = 4$ for Boolean function subspace with $\mathsf{cr}_1 = 1$ and $\mathsf{cr}_0 = 1$, respectively. Their truth tables are listed below:

x_3	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_1	0	1	0	1	0	1	0	1
$f_3(x_3, x_2, x_1)$	0	1	1	0	1	0	0	1
$g_3(x_3, x_2, x_1)$	1	0	0	1	0	1	1	0

The corresponding DNF representations for f_3, g_3, which are also the minimized expressions, are

$$\begin{cases} f_3 = \overline{x_3}\ \overline{x_2} x_1 \vee \overline{x_3} x_2 \overline{x_1} \vee x_3 \overline{x_2}\ \overline{x_1} \vee x_3 x_2 x_1 \\ g_3 = \overline{x_3}\ \overline{x_2}\ \overline{x_1} \vee \overline{x_3} x_2 x_1 \vee x_3 \overline{x_2} x_1 \vee x_3 x_2 \overline{x_1}. \end{cases}$$

Similarly, the minimized CNF formulation is expressed as

$$\begin{cases} f_3 = (x_2 \vee x_1 \vee x_0)(x_2 \vee \overline{x_1} \vee \overline{x_0}) \\ \qquad (\overline{x_2} \vee x_1 \vee \overline{x_0})(\overline{x_2} \vee \overline{x_1} \vee x_0) \\ g_3 = (x_2 \vee x_1 \vee \overline{x_0})(x_2 \vee \overline{x_1} \vee x_0) \\ \qquad (\overline{x_2} \vee x_1 \vee x_0)(\overline{x_2} \vee \overline{x_1} \vee \overline{x_0}) \end{cases}.$$

Further, one can demonstrate that Boolean functions with weight $wt > 4$ (or $\overline{wt} > 4$, respectively) have compression ratio $\mathsf{cr}_1 > 1$ (or $\mathsf{cr}_0 > 1$, respectively).

Example 3. For $n = 4$ Boolean variables $[x_4, x_3, x_2, x_1]$, we are focused on the Boolean function subspaces with $\mathsf{cr}_1 = 1$ and $\mathsf{cr}_0 = 1$, the maximum weight wt_{max} and $\overline{wt}_{max}$, and *hard functions* in Boolean simplifications. The same

analysis performed in Examples 1, 2 can be applied for $n = 4$. There exists only two *hard functions* f_4, g_4 in this domain, defined by the following truth tables that satisfies $wt_{max} = 8$ (and $\overline{wt}_{max} = 8$, respectively) for Boolean function subspace of $\mathsf{cr}_1 = 1$ (and $\mathsf{cr}_0 = 1$).

x_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$f_4(x_4, x_3, x_2, x_1)$	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0
$g_4(x_4, x_3, x_2, x_1)$	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1

For functions f_4, g_4, the minimized DNF expressions are

$$\begin{cases} f_4 = \overline{x_3}\ \overline{x_2}\ \overline{x_1}x_0 \vee \overline{x_3}\ \overline{x_2}x_1\overline{x_0} \vee \overline{x_3}x_2\overline{x_1}\ \overline{x_0} \vee \overline{x_3}x_2x_1x_0 \vee x_3\overline{x_2}\ \overline{x_1}\ \overline{x_0} \vee x_3\overline{x_2}x_1x_0 \vee \\ \qquad x_3x_2\overline{x_1}x_0 \vee x_3x_2x_1\overline{x_0} \\ g_4 = \overline{x_3}\ \overline{x_2}\ \overline{x_1}\ \overline{x_0} \vee \overline{x_3}\ \overline{x_2}x_1x_0 \vee \overline{x_3}x_2\overline{x_1}x_0 \vee \overline{x_3}x_2x_1\overline{x_0} \vee x_3\overline{x_2}\ \overline{x_1}x_0 \vee x_3\overline{x_2}x_1\overline{x_0} \vee \\ \qquad x_3x_2\overline{x_1}\ \overline{x_0} \vee x_3x_2x_1x_0 \end{cases}$$

Correspondingly, minimized CNFs for Boolean functions f_4, g_4 are written below.

$$\begin{cases} f_4 = (x_3 \vee x_2 \vee x_1 \vee x_0)(x_3 \vee x_2 \vee \overline{x_1} \vee \overline{x_0})(x_3 \vee \overline{x_2} \vee x_1 \vee \overline{x_0})(x_3 \vee \overline{x_2} \vee \overline{x_1} \vee x_0) \\ \qquad (\overline{x_3} \vee x_2 \vee x_1 \vee \overline{x_0})(\overline{x_3} \vee x_2 \vee \overline{x_1} \vee x_0)(\overline{x_3} \vee \overline{x_2} \vee x_1 \vee x_0)(\overline{x_3} \vee \overline{x_2} \vee \overline{x_1} \vee \overline{x_0}) \\ g_4 = (x_3 \vee x_2 \vee x_1 \vee \overline{x_0})(x_3 \vee x_2 \vee \overline{x_1} \vee x_0)(x_3 \vee \overline{x_2} \vee x_1 \vee x_0)(x_3 \vee \overline{x_2} \vee \overline{x_1} \vee \overline{x_0}) \\ \qquad (\overline{x_3} \vee x_2 \vee x_1 \vee x_0)(\overline{x_3} \vee x_2 \vee \overline{x_1} \vee \overline{x_0})(\overline{x_3} \vee \overline{x_2} \vee x_1 \vee \overline{x_0})(\overline{x_3} \vee \overline{x_2} \vee \overline{x_1} \vee x_0) \end{cases}$$

We summarize in here that these properties for Boolean functions belonging to Boolean function subspaces of $\mathsf{cr}_1 = 1$ and $\mathsf{cr}_0 = 1$ with worst-case compression ratio and maximum weight wt_{max} and $\overline{wt}_{max}$, respectively, as a theorem.

Theorem 1. *Given any n Boolean variables* $[x_n, x_{n-1}, \ldots, x_1]$*, there are only two hard functions* f_n, g_n *with the following characteristics for Boolean functions with maximum weight* wt_{max} *(and* $\overline{wt}_{max}$*) from Boolean function subspaces with* $\mathsf{cr}_1 = 1$ *(and* $\mathsf{cr}_0 = 1$*):*

1. *Functions* f_n, g_n *are balanced Boolean functions having* $\mathsf{cr}_1 = 1$ *and* $\mathsf{cr}_0 = 1$ *with weight* $wt = 2^n/2$ *and* $\overline{wt} = 2^n/2$*.*
2. *Functions* f_n, g_n *are disjoint. No common AND clauses in DNF expressions, nor any shared OR clauses in CNF expressions. (To put it differently, minimization techniques for DNFs and CNFs cannot further reduce the Boolean expressions.)*
3. *Functions* f_n, g_n *partition the* 2^n*-bit value vector such that* $f_n \wedge g_n = 0$ *and* $f_n \vee g_n = 1$*.*
4. *The weights of both functions* f_n, g_n*,* $wt = 2^n/2$*,* $\overline{wt} = 2^n/2$ *are the maximum possible weight* wt_{max} *for* $\mathsf{cr}_1 = 1$ *and* $\overline{wt}_{max}$ *for* $\mathsf{cr}_0 = 1$*).*

Proof of Theorem 1 is shown in Appendix A.

4 ADC-BE: Broadcast Encryption from Optimized Boolean Functions with ANF, DNF, CNF

The full collusion-resistant broadcast encryption proposed by Dupin and Abelard [11] achieves better encryption bandwidth requirement than Complete Subtree scheme, and is also more efficient than Subset Difference by Naor, Naor, and Lotspiech [20] in practical settings. On the other hand, the authors of [11] acknowledge that the worst-case bandwidth complexity of Complete Subtree scheme is reached in their broadcast encryption scheme and cited theorem of complexity $\mathcal{O}(r\log(n/r))$ of [20] as a corollary (Section 6.1 of [11]).

From Theorem 1 of Section 3-3.2, we have shown that, for any ℓ Boolean variables, there exist only two *hard functions* f_ℓ and g_ℓ with $wt_{max} = 2^\ell/2$ and also $\overline{wt}_{max} = 2^\ell/2$ while $\mathsf{cr_1} = 1$ and $\mathsf{cr_0} = 1$. These *hard functions* accounts for the worst-case broadcast bandwidth of broadcast encryption scheme [11]. As Dupin and Abelard stated in [11], the worst-case complexity of $\mathcal{O}(r\log(n/r))$ in bandwidth requirement can be reached by their broadcast encryption scheme of sum-product decomposition. If the binary encodings of the authorized user list $\mathcal{AU}$ of n entities matches with either Boolean function f_n or g_n, then this This upper-bound of $\mathcal{O}(r\log(n/r)) = \mathcal{O}(n/2)$ is indeed reached in [11]. See Section IV-4.3 for bandwidth reduction comparison of [11], AD-BE, and AC-BE.

To reduce this worst-case bandwidth requirement of $\mathcal{O}(r\log(n/r)) = \mathcal{O}(n/2)$ from [11], we propose a new broadcast encryption suite for constructing symmetric-key-based broadcast encryptions, denoted as ADC-BE, where ADC stands for ANF, DNF and CNF representations of Boolean functions. ADC-BE consists of two separate broadcast encryption algorithms – AD-BE and AC-BE. Both AD-BE and AC-BE rely on the existence of one-way function as the underlying hardness assumption. We consider number of users n as the power of 2 with $\ell = \log(n)$. We use a secure hash function $\mathcal{H}$, two symmetric encryption schemes $\mathcal{E}^1$ and $\mathcal{E}^2$ with decryption algorithms $\mathcal{D}^1$ and $\mathcal{D}^2$, respectively. If the selected symmetric encryption schemes for implementation are secure and have sufficient security margin against Grover's attack for quantum adversaries, our proposed ADC-BE can be quantum-resistant as long as the selected $\mathcal{H}, \mathcal{E}^1, \mathcal{E}^2$ for concrete instantiations are resistant against large-scale cryptanalytically-relevant quantum computers (See Section IV-4.4 for details).

4.1 AD-BE: Broadcast Encryption from Optimized Boolean Functions with ANF, DNF

The *hard functions* f_n, g_n of n Boolean variables are the bandwidth bottleneck ($\mathcal{O}(r\log(n/r)) = \mathcal{O}(n/2)$) if either of them matches the actual encoding of the authorized user list $\mathcal{AU}$. Regardless of DNF or CNF representations, f_n, g_n cannot be minimized as any two monomials in DNF expression (OR clauses in CNF, respectively) are logically nonadjacent. When n is odd, DNF representations for

f_n and g_n are presented as the following.

$$f_n = \bigvee_{v \in S_n^1} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } S_n^1 = \left\{ \prod_{v_i \in v} v_i = +1 \right\},$$
$$g_n = \bigvee_{v \in T_n^1} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } T_n^1 = \left\{ \prod_{v_i \in v} v_i = -1 \right\}.$$

DNF representations for f_n and g_n under an even number n are defined below.

$$f_n = \bigvee_{v \in S_n^1} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } S_n^1 = \left\{ \prod_{v_i \in v} v_i = -1 \right\},$$
$$g_n = \bigvee_{v \in T_n^1} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } T_n^1 = \left\{ \prod_{v_i \in v} v_i = +1 \right\}.$$

Although f_n, g_n are *hard functions* with the most complex structures in DNF, they have equivalent, and efficient representations in algebraic normal form (ANF). Monomials x^v in f_n are from $\left\{ \prod_{v_i \in v} v_i = +1 \right\}$, which can be translated to ANF with $f_n = \sum_{i \in [n]} x_i$. On the other hand, Monomials x^v in g_n satisfy $\left\{ \prod_{v_i \in v} v_i = -1 \right\}$, which is equivalent to ANF of $g_n = \sum_{i \in [n]} x_i + 1$. Therefore, we include ANF representations of f_n, g_n to improve broadcast encryption of [11]. Our proposed AD-BE includes the following four algorithms, $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$.

AD-BE

$\circ$ $\mathsf{Setup}(n, 1^\lambda) \to mk$: Broadcaster first generates its secret master key mk sampled uniformly at random, $mk = \{mk_1 \| \dots \| mk_{2\ell+2}\} \xleftarrow{\$} \mathcal{U}(\mathbb{F}_2^{2(\ell+1)\times\lambda})$.
$\circ$ $\mathsf{KeyGen}(u, mk) \to k_u$: Broadcaster computes the key material k_u for user u, which is a set of keys,

$$k_u = \left\{ \mathcal{H}(\{\|_{\mathcal{S}}\ mk_{2i+u_i-1}\}) \right\}_{\mathcal{S}} \| \begin{Bmatrix} \mathcal{H}(mk_{2\ell+1}) \\ \mathcal{H}(mk_{2\ell+2}) \end{Bmatrix}$$

with $u = [u_1 || \dots || u_\ell]$ and the ordered lexicographic list $\mathcal{S} \subseteq [\ell]$. If $\sum_{i \in [\ell]} u_i = 1$, then $\mathcal{H}(mk_{2\ell+1})$ is also given; otherwise, k_u includes $\mathcal{H}(mk_{2\ell+2})$ as $\sum_{i \in [\ell]} u_i + 1 = 1$.
$\circ$ $\mathsf{Enc}(m, \mathcal{AU}, mk) \to (h, c)$: Broadcaster first convert the authorized user list $\mathcal{AU}$ to optimized Boolean function $\mathcal{B}_{\mathcal{AU}} = (\bigvee_i \mathcal{B}_{\mathcal{AU}\,i}) + q_n$, where

$$\mathcal{B}_{\mathcal{AU}}(u) = \begin{cases} 1, \text{ if } u \in \mathcal{AU} \\ 0, \text{ if } u \notin \mathcal{AU}, \end{cases} \quad q_n = \begin{cases} f_n, \text{ if } f_n \subseteq \mathcal{AU} \\ g_n, \text{ if } g_n \subseteq \mathcal{AU} \\ 0, \text{ if } f_n, g_n \nsubseteq \mathcal{AU} \end{cases}$$

Each monomial $\mathcal{B}_{\mathcal{AU}\,i}$ is converted to 2ℓ-bit binary $h_{\mathcal{AU}_i}$ with an ℓ-bit counter h_c to record the number of $h_{\mathcal{AU}_i}$s. Together with 1-bit indicators h_f and h_g to denote the Boolean function used in q_n, they form the header h,

$$h = h_c \| \{\|_i\ h_{\mathcal{AU}_i}\} \| h_f \| h_g.$$

Ciphertext c is computed by combining the encryptions of k_e under the corresponding keys indicated from $\mathcal{B}_{\mathcal{AU}}$ and q_n, and the encryption of message

m with k_e,

$$c = \left\{ \|_i \ \mathcal{E}^1_{k_{h_{\mathcal{AU}_i}}}(k_e) \right\} \| \left\{ \begin{array}{l} \mathcal{E}^1_{k_{f_n}}(k_e) \\ \mathcal{E}^1_{k_{g_n}}(k_e) \\ \varnothing \end{array} \right\} \| \mathcal{E}^2_{k_e}(m).$$

$\circ$ $\mathsf{Dec}(h, c, k_u) \to m$ else $\perp$: Receiver u checks header h if he/she belongs to list $\mathcal{AU}$ or not. If user $u \in \mathcal{AU}$, it will be able to decrypt k_e and further recover m. Otherwise, it terminates $\perp$ for a revoked user $u \notin \mathcal{AU}$.

Setup. Broadcaster first constructs the master key $mk \in \mathbb{F}_2^{2(\ell+1)\times\lambda}$ by sampling from the uniform distribution with unary string 1^λ, $mk \xleftarrow{\$} \mathcal{U}(1^\lambda)$. This masker key mk can be considered as $2\ell+2$ λ-bit chunks $mk_i \in \mathbb{F}_2^\lambda$, $i \in [2\ell+2]$ such that $mk = \{mk_1 \| \ldots \| mk_{2\ell} \| mk_{2\ell+1} \| mk_{2\ell+2}\}$. Note that key materials for users in the group are derived from the master key mk with one-way function $\mathcal{H}$. Although such key material can be computed on the fly when a new member joins the group, it can be also prepossessed as the emitter knows n. Once such key materials are generated, the broadcaster can store a copy of them and relieves from repeated computation for subsequent KeyGen and Enc steps.

Key Generation. When user u join this broadcast system, the emitter provides it with a set of key materials, based on $u = [u_1 || \ldots || u_\ell]$ with ordered lexicographic list $\mathcal{S} \subseteq [\ell]$,

$$k_u = \left\{ \mathcal{H}(\{\|_{\mathcal{S}} \ mk_{2i+u_i-1}\}) \right\}_{\mathcal{S}} \| \left\{ \begin{array}{l} \mathcal{H}(mk_{2\ell+1}) \\ \mathcal{H}(mk_{2\ell+2}) \end{array} \right\}.$$

User u is also supplied with $k_{f_n} \leftarrow \mathcal{H}(mk_{2\ell+1})$ if it follows $\sum_{i\in[\ell]} u_i = 1$. If not, then u must satisfy $\sum_{i\in[\ell]} u_i + 1 = 1$ and user is given key $k_{g_n} \leftarrow \mathcal{H}(mk_{2\ell+2})$ instead.

For example, for $n = 8, \ell = 3$, user $u = 5 = 101_2$ receives the following key materials in k_u:

$$\begin{array}{lll} \mathcal{H}(mk_2); & \mathcal{H}(mk_3); & \mathcal{H}(mk_6); \\ \mathcal{H}(mk_2||mk_3); & \mathcal{H}(mk_2||mk_6); & \mathcal{H}(mk_3||mk_6); \\ \mathcal{H}(mk_2||mk_3||mk_6); & k_{g_n} \leftarrow \mathcal{H}(mk_8) & \end{array}$$

Encrypt. Suppose emitter would like to send a message m to its intended recipients, who form the list of authorized users $\mathcal{AU}$. Emitter encrypts m with a ephemeral key k_e with $\mathcal{E}^2_{k_e}(m)$. Key k_e is available to authorized users exclusively. This list of authorized users $\mathcal{AU} \subset [n]$ is converted to optimized Boolean function $\mathcal{B}_{\mathcal{AU}} = (\bigvee_i \mathcal{B}_{\mathcal{AU}\,i}) + q_n$, where

$$\mathcal{B}_{\mathcal{AU}}(u) = \begin{cases} 1, \text{ if } u \in \mathcal{AU} \\ 0, \text{ if } u \notin \mathcal{AU}, \end{cases} \ q_n = \begin{cases} f_n, \text{ if } f_n \subseteq \mathcal{AU} \\ g_n, \text{ if } g_n \subseteq \mathcal{AU} \\ 0, \text{ if } f_n, g_n \nsubseteq \mathcal{AU} \end{cases}$$

Note that AD-BE (and AC-BE in Section 4-4.1) support general access structure for the Boolean representation $\mathcal{B}_{\mathcal{AU}}$ of the authorized list, allowing arbitrary Boolean functions as the encoding of $\mathcal{AU}$. Emitter first checks if $\mathcal{AU}$ contains f_n or g_n as its subset. If it does, emitter then transforms $\mathcal{AU}$ by updating the corresponding monomials of f_n (or g_n) as don't-cares. Then, function $\mathcal{AU}$ is minimized to obtain the optimized DNF representation of $\mathcal{B}_{\mathcal{AU}}$, where f_n (or g_n) is added back to the formula, denoted as q_n. Header h encodes this optimal representation. Each monomial in $\mathcal{B}_{\mathcal{AU}i}$ is converted to a 2ℓ-bit binary $h_{\mathcal{AU}_i}$ with ℓ-bit indicator for true variables and another ℓ-bit indicator for the negation. An extra ℓ-bit counter h_c is required to record the number of $h_{\mathcal{AU}_i}$s. Together with 1-bit indicators h_f and h_g to denote which Boolean function is used in q_n (or neither), header h is constructed as below,

$$h = h_c \| \{\|_i \ h_{\mathcal{AU}_i}\} \| h_f \| h_g.$$

Ciphtertext c contains $\mathcal{E}^2_{k_e}(m)$ and multiple encryptions of ephemeral key k_e under valid key materials indicated in $\mathcal{B}_{\mathcal{AU}}$ and $\{h_{\mathcal{AU}_i}\}_i$. If $\mathcal{B}_{\mathcal{AU}}$ includes either f_n or g_n, then the encryption $\mathcal{E}^1_{k_{f_n}}(k_e)$ or $\mathcal{E}^1_{k_{g_n}}(k_e)$ is also present in c.

$$c = \left\{ \|_i \ \mathcal{E}^1_{k_{h_{\mathcal{AU}_i}}}(k_e) \right\} \| \left\{ \begin{array}{l} \mathcal{E}^1_{k_{f_n}}(k_e) \\ \mathcal{E}^1_{k_{g_n}}(k_e) \\ \varnothing \end{array} \right\} \| \mathcal{E}^2_{k_e}(m).$$

Emitter then broadcasts $\{h, c\}$ to all clients.

Decrypt. All users in broadcast system receive $\{h, c\}$ from emitter, where some clients are in the authorized list $\mathcal{AU}$ while others are not. User u checks if any $h_{\mathcal{AU}_i}$ in header h matches with its substrings of $\{u_i\}_i$. If at least one substring coincides with the description in $h_{\mathcal{AU}_i}$, then user possesses the corresponding key $k_{h_{\mathcal{AU}_i}} = \mathcal{H}(\|_{\mathcal{S}} \ mk_{2i+u_i-1})$ to decrypt ephemeral key k_e and apply it to recover message m from $\mathcal{E}^2_{k_e}(m)$. Otherwise, user u is not the intended recipient of message m and proceeds to terminate the decryption procedure (returns $\perp$). Recipient u also examines 1-bit indicators h_f, h_g. If either function f_n, or g_n is used in $\mathcal{B}_{\mathcal{AU}}$, receiver checks whether he/she has corresponding key k_{f_n} or k_{g_n}. If receiver does possess such key, he/she will obtain ephemeral key k_e and decrypt message m. For any intended user, it is guaranteed to correctly decrypt both ephemeral key k_e and message m. If user z is a revoked user (aka not-authorized), he/she does not have the decryption key to obtain k_e, nor subsequently m from k_e. In addition, this broadcast encryption scheme has full collusion resistance as revoked users, combined or individually, do not possess any valid keys for decrypting k_e.

Analysis of AD-BE For symmetric-key-based broadcast encryption, we consider the correctness and security of our scheme. AD-BE has properties of *perfect correctness* and *full collusion-resistance*.

Theorem 2 (Perfect Correctness of AD-BE). *Given AD-BE, an authorized user $u \in \mathcal{AU}$ is able to decrypt the plaintext with probability* 1*.*

$$\Pr\left[u \in \mathcal{AU} : \mathsf{Dec}^{AD\text{-}BE}(h, c, k_u) \to m\right] = 1$$

Proof. The perfect correctness of AD-BE is an extension of the perfect correctness of [11] with additional keys k_{f_n} and k_{g_n}.

(1) Suppose Boolean function $\mathcal{B}_{\mathcal{AU}}$ does not contain f_n nor g_n. Then AD-BE's Enc and Dec algorithms is essentially identical to [11] with extra 1-bit indicators h_f and h_g both being 0. Following the perfect correctness of [11], for AD-BE, any authorized user from the list $\mathcal{AU}$ can recover ephemeral key k_e and decrypt message m.

(2) Suppose Boolean function $\mathcal{B}_{\mathcal{AU}}$ does have f_n or g_n as its subset. Without loss of generality, assume $f_n \subseteq \mathcal{B}_{\mathcal{AU}}$. For Enc algorithm, header h includes $h_f = 1, h_g = 0$, and ciphertext c contains $\mathcal{E}^1_{k_{f_n}}(k_e)$. For an authorized user u, either (i) it was given k_{f_n} during KeyGen if $\sum_{i\in[\ell]} u_i = 1$, or (ii) it possesses at least one (or more) $k_{h_{\mathcal{AU}_j}}$ and k_{g_n}. For client u satisfying condition (i), the message m can be successfully retrieved by first decrypting $\mathcal{E}^1_{k_{f_n}}(k_e)$ to obtain k_e, then applying it on $\mathcal{E}^2_{k_e}(m)$. For client u in condition (ii), message m is also obtained from decryptions of $\mathcal{E}^1_{k_{h_{\mathcal{AU}_j}}}(k_e)$ and $\mathcal{E}^2_{k_e}(m)$. Thus, perfect correctness is achieved.

Theorem 3 (Full Collusion Resistance of AD-BE). *Given AD-BE with secure hash function $\mathcal{H}$, symmetric key encryptions $\mathcal{E}^1, \mathcal{E}^2$ and security parameter 1^λ, let $\mathcal{AU}$ and $\mathcal{RU}$ denote lists of authorized users and revoked users for a particular message m, $\mathcal{RU} = [n]\backslash\mathcal{AU}$. For any adversary $\mathcal{A}$ controlling any subset of revoked users with $\mathcal{RU}^{\mathcal{A}} \subseteq \mathcal{RU}$,*

$$\Pr\left[\forall \mathcal{RU}^{\mathcal{A}} \subseteq \mathcal{RU} : \mathsf{Dec}^{AD\text{-}BE}\left(h, c, \mathcal{RU}^{\mathcal{A}}\right) \to m\right] \leq negl(\lambda).$$

Equivalently, users from the revoked user list $\mathcal{RU}$, individually or combined effort, cannot decrypt the ciphertext c to obtain message m.

Proof. For multiple revoked user to collude (or all of them collude together), it is equivalent to have an adversary $\mathcal{A}$, who controls a subset of revoked users with $\mathcal{RU}^{\mathcal{A}} \subseteq \mathcal{RU}$ and can see all their complete transcripts $\mathcal{T}_{\mathcal{RU}_j^{\mathcal{A}}}$ for protocol execution. The collusion of these revoked users, aka $\mathcal{A}$, is successful if and only if they succeed in decrypting the secret message m, which occurs if they first succeed in decrypting the all the encryptions of ephemeral key k_e with $\mathcal{E}^1$ and $k_{h_{\mathcal{AU}_i}}$. There are three possible scenarios for adversary $\mathcal{A}$ to try to obtain the secret message m with collusion and also for full collusion.

(1) Adversary $\mathcal{A}$ checks if any revoked user $z \in \mathcal{RU}$, or a collection of them, can recover the valid key material(s) $k_{h_{\mathcal{AU}_i}}$, to decrypt $\mathcal{E}^1_{k_{h_{\mathcal{AU}_i}}}(k_e)$. For any intended recipient, it is guaranteed to correctly decrypt both ephemeral key k_e and message m. On the other hand, if user z is not-authorized or revoked, then *(i)* Boolean function outputs $\mathcal{B}_{\mathcal{AU}}(z) = 0$ on input z; *(ii)* header h does not

contain any substring of z, nor the corresponding k_{f_n} or k_{g_n} that z has if f_n or g_n is in $\mathcal{B}_{\mathcal{AU}}$; then *(iii)* ciphertext c has no encryption of k_e that user z could decrypt; and *(iv)* he/she cannot obtain k_e, nor m from k_e. None of the revoked user possesses any valid keys for decrypting k_e. This is true for all revoked users and their combined key materials do not match with any of the $k_{h_{\mathcal{AU}_i}}$ used for encryption of k_e.

(2) Adversary $\mathcal{A}$ tries to brute-force a key material $k_{h_{\mathcal{AU}_i}}$ used in encrypting the ephemeral key k_e, $\mathcal{E}^1_{k_{h_{\mathcal{AU}_i}}}(k_e)$. Note that all key materials, i.e., those received by authorized and unauthorized users, are derived from one-way function $\mathcal{H}$ with segment of secret master key mk_j. Adversary $\mathcal{A}$ holding the invalid key materials $\{\mathcal{H}(\{\|_{\mathcal{S}}\, mk_{2i+z_i-1}\})\}_{\mathcal{S}}$ cannot recover *(i)* mk_{2i+z_i-1}s due to preimage resistance of $\mathcal{H}$; and *(ii)* the master key substring mk_{2i+z_i-1} shares no correlation with the preimage of $k_{h_{\mathcal{AU}_i}}$ due to mk sampled uniformly at random. Therefore, it is impossible for $\mathcal{A}$ to recover any of the valid key material(s) $k_{h_{\mathcal{AU}_i}}$ except with negligible probability of success in breaking preimage resistance of $\mathcal{H}$.

(3) Adversary $\mathcal{A}$ tries to brute-force ephemeral key k_e directly to extract the secret message m from $\mathcal{E}^2_{k_e}(m)$. All revoked users observe $\mathcal{E}^2_{k_e}(m)$ from ciphertext c. However, since they do not have k_e, adversary $\mathcal{A}$ to recover message m without knowing the decryption key k_e has negligible probability of success relative to key space $\mathcal{O}(2^\lambda)$.

4.2 AC-BE: Broadcast Encryption from Optimized Boolean Functions with ANF, CNF

In this section, we consider the dual of the proposed AD-BE broadcast encryption, replacing DNF structures with CNF instead. Note that *hard functions* f_n, g_n have the following CNFs for any positive integer n.

$$f_n = \bigwedge_{v\in S_n^0}\left(\bigvee_{v_j\in v} x_j^{v_j}\right), \text{ with } S_n^0 = \left\{\prod_{v_i\in v} v_i = +1\right\},$$
$$g_n = \bigwedge_{v\in T_n^0}\left(\bigvee_{v_j\in v} x_j^{v_j}\right), \text{ with } T_n^0 = \left\{\prod_{v_i\in v} v_i = -1\right\};$$

Although *hard functions* f_n, g_n have complex representations in CNF, their equivalent representations in ANF is more succinct, $f_n = \sum_{i\in[n]} x_i$ and $g_n = \sum_{i\in[n]} x_i + 1$. We propose AC-BE, a dual of AD-BE, combining both ANFs of f_n, g_n with optimized CNF expressions. It contains the following four algorithms, $\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$.

AC-BE

$\circ$ $\mathsf{Setup}(n, 1^\lambda) \to mk$: Broadcaster first generates its secret master key mk sampled uniformly at random, $mk = \{mk_1 \| \ldots \| mk_{2\ell+2}\} \xleftarrow{\$} \mathcal{U}(\mathbb{F}_2^{2(\ell+1)\times\lambda})$.

$\circ$ $\mathsf{KeyGen}(u, mk) \to k_u$: Broadcaster computes the key material k_u for user u, which is a set of keys,

$$k_u = \left\{\mathcal{H}(\{\|_{\mathcal{S}}\ mk_{2i+u_i-1}\})\right\}_{\mathcal{S}} \| \begin{Bmatrix} \mathcal{H}(mk_{2\ell+1}) \\ \mathcal{H}(mk_{2\ell+2}) \end{Bmatrix}$$

with $u = [u_1 || \ldots || u_\ell]$ and the ordered lexicographic list $\mathcal{S} \subseteq [\ell]$. If $\sum_{i\in[\ell]} u_i = 0$, then $\mathcal{H}(mk_{2\ell+1})$ is also given; otherwise, k_u includes $\mathcal{H}(mk_{2\ell+2})$ as $\sum_{i\in[\ell]} u_i + 1 = 0$.
$\circ$ $\mathsf{Enc}(m, \mathcal{AU}, mk) \to (h, c)$: Broadcaster first convert the authorized user list $\mathcal{AU}$ to optimized Boolean function $\mathcal{B}_{\mathcal{AU}} = (\bigwedge_i \mathcal{B}_{\mathcal{AU}i}) + q_n$, where

$$\mathcal{B}_{\mathcal{AU}}(u) = \begin{cases} 0, \text{ if } u \in \mathcal{AU} \\ 1, \text{ if } u \notin \mathcal{AU}, \end{cases} \quad q_n = \begin{cases} f_n, \text{ if } f_n \subseteq \mathcal{AU} \\ g_n, \text{ if } g_n \subseteq \mathcal{AU} \\ 0, \text{ if } f_n, g_n \nsubseteq \mathcal{AU} \end{cases}$$

Each monomial $\mathcal{B}_{\mathcal{AU}i}$ is converted to 2ℓ-bit binary $h_{\mathcal{AU}_i}$ with an ℓ-bit counter h_c to record the number of $h_{\mathcal{AU}_i}$s. Together with 1-bit indicators h_f and h_g to denote the Boolean function used in q_n, they form the header h,

$$h = h_c \| \{\|_i\ h_{\mathcal{AU}_i}\} \| h_f \| h_g.$$

Ciphertext c is computed by combining the encryptions of k_e under the corresponding keys indicated from $\mathcal{B}_{\mathcal{AU}}$ and q_n, and the encryption of message m with k_e,

$$c = \left\{\|_i\ \mathcal{E}^1_{k_{h_{\mathcal{AU}_i}}}(k_e)\right\} \| \begin{Bmatrix} \mathcal{E}^1_{k_{f_n}}(k_e) \\ \mathcal{E}^1_{k_{g_n}}(k_e) \\ \varnothing \end{Bmatrix} \| \mathcal{E}^2_{k_e}(m).$$

$\circ$ $\mathsf{Dec}(h, c, k_u) \to m$ else $\perp$: Receiver u checks header h if he/she belongs to list $\mathcal{AU}$ or not. If user $u \in \mathcal{AU}$, it will be able to decrypt k_e and further recover m. Otherwise, it terminates $\perp$ for a revoked user $u \notin \mathcal{AU}$.

Setup. Setup procedure is identical to AD-BE. Master key $mk \in \mathbb{F}_2^{2(\ell+1)\times\lambda}$ is sampled from uniform distribution. It can be further partitioned to $2\ell + 2$ λ-bit chunks $mk_i \in \mathbb{F}_2^\lambda$, $i \in [2\ell+2]$ such that $mk = \{mk_1 \| \ldots \| mk_{2\ell} \| mk_{2\ell+1} \| mk_{2\ell+2}\}$.

Key Generation. KeyGen is similar to AD-BE except that user u is given $k_{f_n} \leftarrow \mathcal{H}(mk_{2\ell+1})$ if $\sum_{i\in[\ell]} u_i = 0$, or $k_{g_n} \leftarrow \mathcal{H}(mk_{2\ell+2})$ if $\sum_{i\in[\ell]} u_i + 1 = 0$.

Encrypt. AC-BE represents $\mathcal{AU}$ in Boolean function $\mathcal{B}_{\mathcal{AU}}$ with output 0 for authorized receiver and output 1 for revoked ones.

$$\mathcal{B}_{\mathcal{AU}}(u) = \begin{cases} 0, \text{ if } u \in \mathcal{AU} \\ 1, \text{ if } u \notin \mathcal{AU}, \end{cases} \; q_n = \begin{cases} f_n, \text{ if } f_n \subseteq \mathcal{AU} \\ g_n, \text{ if } g_n \subseteq \mathcal{AU} \\ 0, \text{ if } f_n, g_n \nsubseteq \mathcal{AU} \end{cases}$$

Emitter confirms if $\mathcal{AU}$ contains the CNFs of f_n or g_n as its subset. Emitter first updates $\mathcal{B}_{\mathcal{AU}}$ by replacing the corresponding OR clauses in CNFs of f_n (or g_n) as don't-cares if f_n or g_n is a subset of $\mathcal{AU}$. Boolean function $\mathcal{B}_{\mathcal{AU}}$ is minimized to obtain the optimized CNF representation of $\mathcal{B}_{\mathcal{AU}}$, where f_n (or g_n) is added back to the formula, denoted as q_n. Header h and ciphtertext c follow the same constructions as in AD-BE, which are then sent to all clients.

Decrypt. Decryption procedure follows AD-BE. The intended recipient(s) of secret message m can decrypt ephemeral key k_e then m. Revoked user z, however, has *(i)* evaluation of $f(z) = 1$ instead; *(ii)* none of binary patterns of key materials which user z received is included in header h, nor keys k_{f_n} or k_{g_n}) if it is included; *(iii)* ciphertext c has no encryption of ephemeral key k_e that user z could decrypt; and *(iv)* he/she cannot obtain k_e nor message m. In like manner, AC-BE is full collusion resistant as all revoked user combined cannot decrypt k_e, m.

Analysis of AC-BE Analysis of perfect correctness and full collusion-resistance for AD-BE can be extended to AC-BE. Proofs for AD-BE can be adapted easily to its dual version - AC-BE.

Theorem 4 (Perfect Correctness of AC-BE). *Given AD-BE, an authorized user $u \in \mathcal{AU}$ is able to decrypt the plaintext with probability* 1.

$$\Pr\left[u \in \mathcal{AU} : \mathsf{Dec}^{AC\text{-}BE}(h, c, k_u) \rightarrow m\right] = 1$$

Theorem 5 (Full Collusion Resistance of AC-BE). *Given AC-BE with secure hash function $\mathcal{H}$, symmetric key encryptions $\mathcal{E}^1, \mathcal{E}^2$ and security parameter 1^λ, let $\mathcal{AU}$ and $\mathcal{RU}$ denote lists of authorized users and revoked users for a particular message m, $\mathcal{RU} = [n] \backslash \mathcal{AU}$. For any adversary $\mathcal{A}$ controlling any subset of revoked users with $\mathcal{RU}^{\mathcal{A}} \subseteq \mathcal{RU}$,*

$$\Pr\left[\forall \mathcal{RU}^{\mathcal{A}} \subseteq \mathcal{RU} : \mathsf{Dec}^{AC\text{-}BE}\left(h, c, \mathcal{RU}^{\mathcal{A}}\right) \rightarrow m\right] \leq negl(\lambda).$$

4.3 Bandwidth Reduction in ADC-BE

The proposed ADC-BE allows significant reduction of bandwidth which could not be achieved in [11]. If $\mathcal{AU} \equiv f_n$ or $\mathcal{AU} \equiv g_n$, we achieve a bandwidth

Table 1: Communication bandwidth comparisons of [11], AD-BE, and AC-BE, as its size depends on the header h length and ciphertext c length, when emitter broadcasts message m under the worst-case bandwidth scenario, i.e., when the binary encoding of the authorized users list $\mathcal{AU}$ coincides with either *hard functions* f_n or g_n.

Schemes	**Header** (bits)	**Ciphertext**		**Total** (bits)
		#Ciphertext	Ciphertext (bits)	
[11]	$(n+1)\cdot\log(n)$	$\frac{n}{2}+1$ $(=\frac{n}{2}\mathcal{E}^1+1\mathcal{E}^2)$	$(n/2+1)\cdot\lambda$	$(n+1)\cdot\log(n)+(n/2+1)\cdot\lambda$
Our AD-BE	2 bits	2 $(=1\mathcal{E}^1+1\mathcal{E}^2)$	2λ	$2\lambda+2$
Our AC-BE	2 bits	2 $(=1\mathcal{E}^1+1\mathcal{E}^2)$	2λ	$2\lambda+2$

of a single encryption $\mathcal{E}^1$ compared to the previous $2^\ell/2 = n/2$ encryptions of ephemeral key k_e. This improvement in fewest number of encryptions is achieved in both proposed AD-BE and AC-BE. Table 1 summarizes the communication bandwidth comparison between [11], AD-BE, and AC-BE, as its depends on the header h length and ciphertext c length, when emitter broadcast message m under the worst-case bandwidth scenario. Whereas the communication bottleneck is quasi-linear in the number of users n in the system for [11], this worst-case bandwidth is dramatically improved by AD-BE and AC-BE to constant size only, irrespective of n. Note we consider $\mathcal{E}^1$ and $\mathcal{E}^2$ of security 1^λ with λ-bit output. As shown in Table 1, the worst-case bandwidth of [11] has quasi-linear header h length of $(n+1)\cdot\log(n)$ bits, along with $n/2$ $\mathcal{E}^1$ encryptions (and 1 in $\mathcal{E}^2$), resulting a total of $(n+1)\cdot\log(n)+(n/2+1)\cdot\lambda$-bit broadcast to the system. In contrast, our proposed AD-BE and AC-BE under the same condition only require 2-bit header h, e.g., 1-bit indicators h_f, h_g since no h_c or $h_{\mathcal{AU}_i}$s are involved, and single encryption in $\mathcal{E}^1$ and $\mathcal{E}^2$, giving a total of $2\lambda+2$-bits in communication.

In addition, if the number of revoked users are fewer than $n/2$, any function $\mathcal{AU}$ with f_n (or g_n) as its subset has $\mathcal{E}^1_{k_{f_n}}(k_e)$ or $\mathcal{E}^1_{k_{g_n}}(k_e)$ to account for the $n/2$ DNF monomials for AD-BE, or $n/2$ CNF clauses for AC-BE. As $n/2$ Boolean terms are now considered as don't cares, it leads to less execution time in Boolean optimizations as much fewer prime implicants are required. Thus, inclusion of f_n and g_n in ANF offers further optimization of the encryption runtime for broadcast encryption schemes.

4.4 On the Post-Quantum Security of Proposed Broadcast Encryption

Post-quantum security refers to the cryptographic primitives withstanding quantum attackers, i.e., adversaries with access to large-scale cryptanalytically-relevant quantum computers [31,32]. Shor's algorithm [24] provides exponential speedup in solving discrete logarithms and integer factorization while Grover's algorithm [15] gives quadratic speedup in quantum search algorithm.

The security of AD-BE and AC-BE relies on the existence of one-way function, e.g., secure hash function. It does not depend on any number-theoretic assumptions or structured hardness assumptions affected by Shor's algorithm. To secure ADC-BE suite against quantum adversaries, we instantiate hash function $\mathcal{H}$, encryptions $\mathcal{E}^1, \mathcal{E}^2$ with at least 256-bit security in the classical setting [5]. Examples include SHA3-256, AES-256, or Rijndael-256-256. With sufficient security margin against Grover's attack, AD-BE and AC-BE can be made resistant to quantum attackers.

5 Conclusion

In this paper, we target the worst-case upper-bound in bandwidth requirement of the recent work of Dupin and Abelard [11], which achieves full collision resistance for any number of revoked r users from the complete set of n users. We first analyze the worst-case complexity of Dupin and Abelard and identify the unique patterns in authorized user list. Then, we provide a new insight to lower complexity cost in broadcast bandwidth bottleneck with different representation of Boolean functions. In particular, the worst-case complexity of broadcast encryption bandwidth can be drastically improved when applying combined forms for Boolean functions while preserving the security guarantees in [11]. Our proposed approaches AD-BE and AC-BE can significantly optimize this worst-case complexity from $\mathcal{O}(n/2)$ down to $\mathcal{O}(1)$ for a system of n users. In addition, we provide equivalent transformation of the broadcast encryption construction of disjunctive formal forms with its conjunctive normal form counterpart. Furthermore, our construction of ADC-BE suite allows general access structures for the attribute policy, supporting arbitrary Boolean functions without any restrictions (i.e., monotone access structure). Lastly, both AD-BE and AC-BE can be made post-quantum as the security depends on one-way function, e.g., secure hash function.

Appendix A

We prove Theorem 1 by proof of induction, which internally contains proof of contradiction in a few sub-cases.

Proof. In proof of induction, we distinguish the inductive step with ℓ when it is an even or odd number. We thus have two initial conditions $\ell = 2$ and $\ell = 3$ to begin with.

- Initial step of $\ell = 2$: With $\ell = 2$ Boolean variables, there are $2^{2^2} = 16$ possible Boolean functions. As we have detailed in Example 1, functions $f_2 = {x_2}^{-1}x_1 \vee x_2{x_1}^{-1}$ and $g_2 = {x_2}^{-1}{x_1}^{-1} \vee x_2x_1$ (and $f_2 = (x_2 \vee x_1)({x_2}^{-1} \vee {x_1}^{-1})$, $g_2 = (x_2 \vee {x_1}^{-1})({x_2}^{-1} \vee x_1)$ respectively in CNF) are *hard functions*. Both f_2, g_2 are balanced functions having $\mathsf{cr_1} = 1$ (and $\mathsf{cr_0} = 1$) with weights $wt = 2^\ell/2 = 2$, and $\overline{wt} = 2^\ell/2 = 2$. Based on truth tables, f_2, g_2 are disjoint, and they partition 2^ℓ-bit value vector such that $f_\ell \wedge g_\ell = 0$ and $f_n \vee g_n = 1$. Weights of f_ℓ, g_ℓ, $wt = 2^\ell/2$ (and $\overline{wt} = 2^\ell/2$), are the maximum for $\mathsf{cr_1} = 1$ (and $\mathsf{cr_0} = 1$) of $\ell = 2$ variables, since functions with $wt > 2^\ell/2$ (and $\overline{wt} > 2^\ell/2$) have $\mathsf{cr_1} > 1$ (and $\mathsf{cr_0} > 1$). Moreover, out of all 6 functions with $wt = 2^\ell/2$ (and $\overline{wt} = 2^\ell/2$), only f_2, g_2 has $\mathsf{cr_1} = 1$ (and $\mathsf{cr_0} = 1$). Therefore, f_2, g_2 are the only two *hard functions* for $\ell = 2$.
  To facilitate discussions in inductive step, we abbreviate the DNF expressions for f_2, g_2 as the following:

$$f_2 = \bigvee\nolimits_{v \in S_2^1} \left(\bigwedge\nolimits_{v_j \in v} x_j^{v_j} \right), \text{ with } S_2^1 = \{[v = v_2v_1 = -1, +1], [v = +1, -1]\},$$
$$g_2 = \bigvee\nolimits_{v \in T_2^1} \left(\bigwedge\nolimits_{v_j \in v} x_j^{v_j} \right), \text{ with } T_2^1 = \{[v = -1, -1], [v = +1, +1]\}.$$

  We can summarize sets S_2^1 and T_2^1 in regard to it individual bits $v_i \in v$ with $S_2^1 = \left\{\prod_{v_i \in v} v_i = -1\right\}$, $T_2^1 = \left\{\prod_{v_i \in v} v_i = +1\right\}$. CNF formulations for f_2, g_2 are denoted as

$$f_2 = \bigwedge\nolimits_{v \in S_2^0} \left(\bigvee\nolimits_{v_j \in v} x_j^{v_j} \right), \text{ with } S_2^0 = \{[+1, +1], [-1, -1]\},$$
$$g_2 = \bigwedge\nolimits_{v \in T_2^0} \left(\bigvee\nolimits_{v_j \in v} x_j^{v_j} \right), \text{ with } T_2^0 = \{[+1, -1], [-1, +1]\},$$

  where sets S_2^0 and T_2^0 are equivalent to the multiplicative representations of $S_2^0 = \left\{\prod_{v_i \in v} v_i = +1\right\}$, $T_2^0 = \left\{\prod_{v_i \in v} v_i = -1\right\}$.
- Initial step of $\ell = 3$: With $\ell = 3$ Boolean variables, there are $2^{2^3} = 64$ possible Boolean functions. As we have detailed in Example 2, functions f_3 and g_3 are *hard functions*, whose DNF representations are shown below.

$$\begin{cases} f_3 = {x_3}^{-1}{x_2}^{-1}x_1 \vee {x_3}^{-1}x_2{x_1}^{-1} \vee x_3{x_2}^{-1}{x_1}^{-1} \vee x_3x_2x_1 \\ g_3 = {x_3}^{-1}{x_2}^{-1}{x_1}^{-1} \vee {x_3}^{-1}x_2x_1 \vee x_3{x_2}^{-1}x_1 \vee x_3x_2{x_1}^{-1}, \end{cases}$$

  CNF representations of f_3 and g_3 are

$$\begin{cases} f_3 = (x_2 \vee x_1 \vee x_0)(x_2 \vee {x_1}^{-1} \vee {x_0}^{-1})({x_2}^{-1} \vee x_1 \vee {x_0}^{-1})({x_2}^{-1} \vee {x_1}^{-1} \vee x_0) \\ g_3 = (x_2 \vee x_1 \vee {x_0}^{-1})(x_2 \vee {x_1}^{-1} \vee x_0)({x_2}^{-1} \vee x_1 \vee x_0)({x_2}^{-1} \vee {x_1}^{-1} \vee {x_0}^{-1}). \end{cases}$$

Both f_3, g_3 are balanced functions having $\mathsf{cr}_1 = 1$ (and $\mathsf{cr}_0 = 1$) with weight $wt = 2^\ell/2 = 4$, $\overline{wt} = 2^\ell/2$. Based on truth tables, f_3, g_3 are disjoint, and they partition 2^ℓ-bit value vector such that $f_\ell \wedge g_\ell = 0$ and $f_n \vee g_n = 1$. Weights of f_ℓ, g_ℓ, $wt = 2^\ell/2$ (and $\overline{wt} = 2^\ell/2$), are the maximum for $\mathsf{cr}_1 = 1$ (and $\mathsf{cr}_0 = 1$) of $\ell = 3$ variables, since functions with $wt > 2^\ell/2$ (and $\overline{wt} > 2^\ell/2$) have $\mathsf{cr}_1 > 1$ (and $\mathsf{cr}_0 > 1$). In addition, of all functions with $wt = 2^\ell/2$ (and $\overline{wt} = 2^\ell/2$), only f_3, g_3 has $\mathsf{cr}_1 = 1$ (and $\mathsf{cr}_0 = 1$). Therefore, f_3, g_3 are the only *hard functions* for $\ell = 3$.
We follow the abbreviation of above for DNF formulations of f_3, g_3

$$f_3 = \bigvee_{v \in S_3^1} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \; S_3^1 = \{[-1,-1,+1],[-1,+1,-1],[+1,-1,-1],[+1,+1,+1]\},$$
$$g_3 = \bigvee_{v \in T_3^1} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \; T_3^1 = \{[-1,-1,-1],[-1,+1,+1],[+1,-1,+1],[+1,+1,-1]\}.$$

Both sets S_3^1 and T_3^1 can be described succinctly as $S_3^1 = \left\{\prod_{v_i \in v} v_i = +1\right\}$, $T_3^1 = \left\{\prod_{v_i \in v} v_i = -1\right\}$. CNFs for f_3, g_3 are represented by

$$f_3 = \bigwedge_{v \in S_3^0} \left(\bigvee_{v_j \in v} x_j^{v_j} \right), \; S_3^0 = \{[+1,+1,+1],[+1,-1,-1],[-1,+1,-1],[-1,-1,+1]\},$$
$$g_3 = \bigwedge_{v \in T_3^0} \left(\bigvee_{v_j \in v} x_j^{v_j} \right), \; T_3^0 = \{[+1,+1,-1],[+1,-1,+1],[-1,+1,+1],[-1,-1,-1]\}.$$

Sets S_3^0 and T_3^0 are identical to S_3^1 and T_3^1, respectively, $S_3^0 = S_3^1 = \left\{\prod_{v_i \in v} v_i = +1\right\}$, $T_3^0 = T_3^1 = \left\{\prod_{v_i \in v} v_i = -1\right\}$.

– Inductive step of $\ell \geq 2$: Suppose we have functions f_ℓ, g_ℓ, the *hard functions* for ℓ Boolean variables, satisfying the above four properties. Our goal is to identify and prove *hard functions* $f_{\ell+1}, g_{\ell+1}$ of $(\ell+1)$ variables. We distinguish two cases and prove them separately: (*i*) $\ell = 2i$ is even, or (*ii*) $\ell = 2i+1$ is an odd integer instead.

(i) ($\ell = 2i$ is even.) *Hard functions* f_ℓ, g_ℓ of ℓ Boolean variables are represented in DNF as

$$f_\ell = \bigvee_{v \in S_\ell^1} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } S_\ell^1 = \left\{\prod_{v_i \in v} v_i = -1\right\},$$
$$g_\ell = \bigvee_{v \in T_\ell^1} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } T_\ell^1 = \left\{\prod_{v_i \in v} v_i = +1\right\}.$$

CNF expressions for *hard functions* f_ℓ, g_ℓ are

$$f_\ell = \bigwedge_{v \in S_\ell^0} \left(\bigvee_{v_j \in v} x_j^{v_j} \right), \text{ with } S_\ell^0 = \left\{\prod_{v_i \in v} v_i = +1\right\},$$
$$g_\ell = \bigwedge_{v \in T_\ell^0} \left(\bigvee_{v_j \in v} x_j^{v_j} \right), \text{ with } T_\ell^0 = \left\{\prod_{v_i \in v} v_i = -1\right\}.$$

We first construct $f_{\ell+1}$ and $g_{\ell+1}$ of $(\ell+1)$ Boolean variables $\{v_{\ell+1}, \ldots, v_1\}$ from *hard functions* f_ℓ, g_ℓ of ℓ variables $\{v_\ell, \ldots, v_1\}$, then prove that $f_{\ell+1}$ and $g_{\ell+1}$ satisfy all four properties and are the only *hard functions*. Integer $\ell + 1 = 2i + 1$ is an odd number, and we formulate $f_{\ell+1}$ and $g_{\ell+1}$ in DNF as

$$
\begin{aligned}
f_{\ell+1} &= \left[\bigvee_{v\in S^1_\ell}\left(\bigwedge_{v_j\in v_{[\ell]}} x_j^{v_j}\bigwedge x_{\ell+1}^{-1}\right)\right]\bigvee\left[\bigvee_{v\in T^1_\ell}\left(\bigwedge_{v_j\in v_{[\ell]}} x_j^{v_j}\bigwedge x_{\ell+1}^{+1}\right)\right]\\
&= \left[f_\ell\bigwedge x_{\ell+1}^{-1}\right]\bigvee\left[g_\ell\bigwedge x_{\ell+1}^{+1}\right]\\
&= \bigvee_{v\in S^1_{\ell+1}}\left(\bigwedge_{v_j\in v} x_j^{v_j}\right), \text{ with } S^1_{\ell+1}=\left\{\prod_{v_i\in v_{[\ell+1]}} v_i=+1\right\},
\end{aligned}
$$

$$
\begin{aligned}
g_{\ell+1} &= \left[\bigvee_{v\in S^1_\ell}\left(\bigwedge_{v_j\in v_{[\ell]}} x_j^{v_j}\bigwedge x_{\ell+1}^{+1}\right)\right]\bigvee\left[\bigvee_{v\in T^1_\ell}\left(\bigwedge_{v_j\in v_{[\ell]}} x_j^{v_j}\bigwedge x_{\ell+1}^{-1}\right)\right]\\
&= \left[f_\ell\bigwedge x_{\ell+1}^{+1}\right]\bigvee\left[g_\ell\bigwedge x_{\ell+1}^{-1}\right]\\
&= \bigvee_{v\in T^1_{\ell+1}}\left(\bigwedge_{v_j\in v} x_j^{v_j}\right), \text{ with } T^1_{\ell+1}=\left\{\prod_{v_i\in v_{[\ell+1]}} v_i=-1\right\}.
\end{aligned}
$$

From the above DNF constructions of $f_{\ell+1}$ and $g_{\ell+1}$, both functions do not have any logically adjacent minterms, since at least 2 Boolean variables differ between any two AND clauses in $f_{\ell+1}$, and also for function $g_{\ell+1}$. Therefore, both $f_{\ell+1}$ and $g_{\ell+1}$ have $\mathsf{cr_1}=1$ with 2^ℓ AND monomials, weight $wt=2^\ell=2^{\ell+1}/2$, satisfying Characteristic #1. Functions $f_{\ell+1}$ and $g_{\ell+1}$ are also disjoint, where no common AND monomials (minterms) can be found, which is Characteristic #2. This Characteristic #2 also implies $f_{\ell+1}\wedge g_{\ell+1}=0$. As $f_{\ell+1}$ and $g_{\ell+1}$ are disjoint functions and each having 2^ℓ monomials, the OR evaluation of $f_{\ell+1}\vee g_{\ell+1}$ consists of $2^\ell+2^\ell=2^{\ell+1}$ monomials, which is all possible AND monomials for $(\ell+1)$ Boolean variables, which means that $f_{\ell+1}\vee g_{\ell+1}=1$. Having $f_{\ell+1}\wedge g_{\ell+1}=0$ and $f_{\ell+1}\vee g_{\ell+1}=1$ fulfill the Characteristic #3. We examine Characteristic #4 by proof of contradiction. Suppose in the Boolean function subspace of $\mathsf{cr_1}=1$ with $(\ell+1)$ variables has the maximum weight $wt_{max}>2^\ell$, which means that weight $wt=2^\ell$ of $f_{\ell+1}$ and $g_{\ell+1}$ is not the largest in this subspace. Without loss of generality, let us consider the maximum weight $wt_{max}=2^\ell+1$, as any larger weight can be subsequently build from the following argument. We denote this function of $wt_{max}=2^\ell+1$ as h_1. Its weight means that there exist $(2^\ell+1)$ AND monomials which are logically nonadjacent and cannot be simplified. Thus, we can build it from either $f_{\ell+1}$ or $g_{\ell+1}$ by including an additional AND clause. If we build h_1 from $f_{\ell+1}$, we need to pick one monomial from function $g_{\ell+1}$, as the remaining monomials has to come from $g_{\ell+1}$ due to Characteristic #3. No matter which AND clause to choose from $g_{\ell+1}$, say $v'_{[\ell+1]}=\{v'_\ell,\ldots,v'_1\}$ with $\prod_{v_i\in v} v_i=-1$, there exists a monomial from $f_{\ell+1}$, say $v_{[\ell+1]}=\{v'_\ell,\ldots,v_j,\ldots,v'_1\}$ with $v_j\neq v'_j$ and $\prod_{v_i\in v} v_i=+1$ such

that they differ by one input only. (There are a total of $(\ell+1)$ monomials in $f_{\ell+1}$ being logically adjacent to any AND monomial in $g_{\ell+1}$). This means that two AND clauses are logically adjacent and therefore can be minimized with theorems in Boolean algebra having $\mathsf{cr_1} > 1$, a contradiction from Boolean function subspace of $\mathsf{cr_1} = 1$. The identical analysis can be perform to constructing h_1 from $g_{\ell+1}$ with one additional monomial from $f_{\ell+1}$. However, no matter which monomial to pick from $f_{\ell+1}$, it always is logically adjacent with one (or more) AND clauses in $g_{\ell+1}$, implying $\mathsf{cr_1} > 1$. Again, it is a contraction from $\mathsf{cr_1} = 1$. Thus, no such function h_1 exists and the maximum weight is $wt_{max} = 2^{\ell+1}/2$ for Boolean function subspace of $\mathsf{cr_1} = 1$. Functions $f_{\ell+1}$ and $g_{\ell+1}$ are the *hard functions* in DNF representation. Lastly, we verify that functions $f_{\ell+1}$ and $g_{\ell+1}$ are the only *hard functions* of weight is $wt_{max} = 2^{\ell}$ with $\mathsf{cr_1} = 1$. Any functions h_2 of $wt = 2^{\ell}$ other than $f_{\ell+1}$, $g_{\ell+1}$ can be composed of replacing one or more AND monomials from $f_{\ell+1}$ with monomials of $g_{\ell+1}$ (and vice-versa). Suppose such function h_2 of $wt_{max} = 2^{\ell}$ with $\mathsf{cr_1} = 1$ exists. Without loss of generality, there exists at least one monomial, say $v'_{[\ell+1]} = \{v'_{\ell}, \ldots, v'_1\}$ with $\prod_{v_i \in v} v_i = -1$, included in h_2 from $g_{\ell+1}$, being logically adjacent with the existing monomial(s) in $f_{\ell+1}$, e.g., $v_{[\ell+1]} = \{v'_{\ell}, \ldots, v_j, \ldots, v'_1\}$ with $v_j \neq v'_j$. We again have a contradiction of $\mathsf{cr_1} > 1 \neq 1$. Therefore, Boolean functions $f_{\ell+1}$ and $g_{\ell+1}$ are the only *hard functions* in DNF representation.

Boolean functions $f_{\ell+1}$ and $g_{\ell+1}$ can also be formulated in CNF as

$$
\begin{aligned}
f_{\ell+1} &= \left[\bigwedge_{v \in S^1_{\ell}} \left(\bigvee_{v_j \in v_{[\ell]}} x_j^{v_j} \bigvee x_{\ell+1}^{+1}\right)\right] \bigwedge \left[\bigwedge_{v \in T^1_{\ell}} \left(\bigvee_{v_j \in v_{[\ell]}} x_j^{v_j} \bigvee x_{\ell+1}^{-1}\right)\right] \\
&= \left[f_{\ell} \bigvee x_{\ell+1}^{+1}\right] \bigwedge \left[g_{\ell} \bigvee x_{\ell+1}^{-1}\right] \\
&= \bigwedge_{v \in S^0_{\ell+1}} \left(\bigvee_{v_j \in v} x_j^{v_j}\right), \text{ with } S^0_{\ell+1} = \left\{\prod_{v_i \in v_{[\ell+1]}} v_i = +1\right\},
\end{aligned}
$$

$$
\begin{aligned}
g_{\ell+1} &= \left[\bigwedge_{v \in S^1_{\ell}} \left(\bigvee_{v_j \in v_{[\ell]}} x_j^{v_j} \bigvee x_{\ell+1}^{-1}\right)\right] \bigwedge \left[\bigvee_{v \in T^1_{\ell}} \left(\bigvee_{v_j \in v_{[\ell]}} x_j^{v_j} \bigvee x_{\ell+1}^{+1}\right)\right] \\
&= \left[f_{\ell} \bigvee x_{\ell+1}^{-1}\right] \bigwedge \left[g_{\ell} \bigvee x_{\ell+1}^{+1}\right] \\
&= \bigwedge_{v \in T^0_{\ell+1}} \left(\bigvee_{v_j \in v} x_j^{v_j}\right), \text{ with } T^0_{\ell+1} = \left\{\prod_{v_i \in v_{[\ell+1]}} v_i = -1\right\}.
\end{aligned}
$$

It can be verified that all Characteristics #1,2,3,4 are valid on $f_{\ell+1}$ and $g_{\ell+1}$ and they are the only *hard functions* in CNF representation. This proof is identical to proving the DNF formulations except we now have OR clauses instead AND monomials, which we omit details here. Boolean

functions $f_{\ell+1}$ and $g_{\ell+1}$ are the only *hard functions* with maximum weight $wt_{max} = 2^{\ell+1}/2$ from Boolean function subspaces with $\mathsf{cr}_1 = 1$ and $\mathsf{cr}_0 = 1$.

(ii) ($\ell = (2i+1)$ is odd.) The only *hard functions* f_ℓ, g_ℓ of ℓ Boolean variables are expressed in DNF as

$$f_\ell = \bigvee_{v \in S^1_\ell} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } S^1_\ell = \left\{ \prod_{v_i \in v} v_i = +1 \right\},$$
$$g_\ell = \bigvee_{v \in T^1_\ell} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } T^1_\ell = \left\{ \prod_{v_i \in v} v_i = -1 \right\}.$$

CNF formulations for *hard functions* f_ℓ, g_ℓ are

$$f_\ell = \bigwedge_{v \in S^0_\ell} \left(\bigvee_{v_j \in v} x_j^{v_j} \right), \text{ with } S^0_\ell = \left\{ \prod_{v_i \in v} v_i = +1 \right\},$$
$$g_\ell = \bigwedge_{v \in T^0_\ell} \left(\bigvee_{v_j \in v} x_j^{v_j} \right), \text{ with } T^0_\ell = \left\{ \prod_{v_i \in v} v_i = -1 \right\};$$

Boolean functions $f_{\ell+1}$ and $g_{\ell+1}$ of $(\ell+1)$ variables $\{v_{\ell+1}, \ldots, v_1\}$ can be built from the above *hard functions* f_ℓ, g_ℓ of ℓ variables $\{v_\ell, \ldots, v_1\}$. We construct their DNF representations as follows,

$$\begin{aligned} f_{\ell+1} &= \left[\bigvee_{v \in S^1_\ell} \left(\bigwedge_{v_j \in v_{[\ell]}} x_j^{v_j} \bigwedge x_{\ell+1}^{-1} \right) \right] \bigvee \left[\bigvee_{v \in T^1_\ell} \left(\bigwedge_{v_j \in v_{[\ell]}} x_j^{v_j} \bigwedge x_{\ell+1}^{+1} \right) \right] \\ &= \left[f_\ell \bigwedge x_{\ell+1}^{-1} \right] \bigvee \left[g_\ell \bigwedge x_{\ell+1}^{+1} \right] \\ &= \bigvee_{v \in S^1_{\ell+1}} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } S^1_{\ell+1} = \left\{ \prod_{v_i \in v_{[\ell+1]}} v_i = -1 \right\}, \end{aligned}$$

$$\begin{aligned} g_{\ell+1} &= \left[\bigvee_{v \in S^1_\ell} \left(\bigwedge_{v_j \in v_{[\ell]}} x_j^{v_j} \bigwedge x_{\ell+1}^{+1} \right) \right] \bigvee \left[\bigvee_{v \in T^1_\ell} \left(\bigwedge_{v_j \in v_{[\ell]}} x_j^{v_j} \bigwedge x_{\ell+1}^{-1} \right) \right] \\ &= \left[f_\ell \bigwedge x_{\ell+1}^{+1} \right] \bigvee \left[g_\ell \bigwedge x_{\ell+1}^{-1} \right] \\ &= \bigvee_{v \in T^1_{\ell+1}} \left(\bigwedge_{v_j \in v} x_j^{v_j} \right), \text{ with } T^1_{\ell+1} = \left\{ \prod_{v_i \in v_{[\ell+1]}} v_i = +1 \right\}. \end{aligned}$$

Both functions functions $f_{\ell+1}$ and $g_{\ell+1}$ have the corresponding DNF representations:

$$
\begin{aligned}
f_{\ell+1} &= \left[\bigwedge_{v\in S_\ell^1}\left(\bigvee_{v_j\in v_{[\ell]}} x_j^{v_j}\bigvee x_{\ell+1}^{+1}\right)\right]\bigwedge\left[\bigwedge_{v\in T_\ell^1}\left(\bigvee_{v_j\in v_{[\ell]}} x_j^{v_j}\bigvee x_{\ell+1}^{-1}\right)\right] \\
&= \left[f_\ell\bigvee x_{\ell+1}^{+1}\right]\bigwedge\left[g_\ell\bigvee x_{\ell+1}^{-1}\right] \\
&= \bigwedge_{v\in S_{\ell+1}^0}\left(\bigvee_{v_j\in v} x_j^{v_j}\right), \text{ with } S_{\ell+1}^0 = \left\{\prod_{v_i\in v_{[\ell+1]}} v_i = +1\right\},
\end{aligned}
$$

$$
\begin{aligned}
g_{\ell+1} &= \left[\bigwedge_{v\in S_\ell^1}\left(\bigvee_{v_j\in v_{[\ell]}} x_j^{v_j}\bigvee x_{\ell+1}^{-1}\right)\right]\bigwedge\left[\bigvee_{v\in T_\ell^1}\left(\bigvee_{v_j\in v_{[\ell]}} x_j^{v_j}\bigvee x_{\ell+1}^{+1}\right)\right] \\
&= \left[f_\ell\bigvee x_{\ell+1}^{-1}\right]\bigwedge\left[g_\ell\bigvee x_{\ell+1}^{+1}\right] \\
&= \bigwedge_{v\in T_{\ell+1}^0}\left(\bigvee_{v_j\in v} x_j^{v_j}\right), \text{ with } T_{\ell+1}^0 = \left\{\prod_{v_i\in v_{[\ell+1]}} v_i = -1\right\}.
\end{aligned}
$$

The process of proving $f_{\ell+1}$ and $g_{\ell+1}$ of $(\ell+1) = 2i+2$ as the only *hard functions* with Characteristics #1,2,3,4 in both DNF and CNF formats can be borrowed from the proof in step (i) of $(\ell+1) = 2i+1$; thus we omit its proof details.

□

Acknowledgements The author thanks God for this research idea and writing. This document is written without generative AI.

References

1. Agrawal, S., Yamada, S.: Optimal broadcast encryption from pairings and LWE. In: Canteaut, A., Ishai, Y. (eds.) Advances in Cryptology – EUROCRYPT 2020, Part I. Lecture Notes in Computer Science, vol. 12105, pp. 13–43. Springer, Cham (May 2020). https://doi.org/10.1007/978-3-030-45721-1˙2
2. Aravamuthan, S., Lodha, S.: An optimal subset cover for broadcast encryption. In: Maitra, S., Madhavan, C.E.V., Venkatesan, R. (eds.) Progress in Cryptology - INDOCRYPT 2005: 6th International Conference in Cryptology in India. Lecture Notes in Computer Science, vol. 3797, pp. 221–231. Springer, Berlin, Heidelberg (Dec 2005). https://doi.org/10.1007/11596219˙18
3. Attrapadung, N., Imai, H.: Graph-decomposition-based frameworks for subset-cover broadcast encryption and efficient instantiations. In: Roy, B.K. (ed.) Advances in Cryptology – ASIACRYPT 2005. Lecture Notes in Computer Science, vol. 3788, pp. 100–120. Springer, Berlin, Heidelberg (Dec 2005). https://doi.org/10.1007/11593447˙6

4. Austrin, P., Kreitz, G.: Lower bounds for subset cover based broadcast encryption. In: Vaudenay, S. (ed.) AFRICACRYPT 08: 1st International Conference on Cryptology in Africa. Lecture Notes in Computer Science, vol. 5023, pp. 343–356. Springer, Berlin, Heidelberg (Jun 2008). https://doi.org/10.1007/978-3-540-68164-9˙23
5. Bernstein, D.J., Lange, T.: Post-quantum cryptography—dealing with the fallout of physics success. Cryptology ePrint Archive, Report 2017/314 (2017), `https://eprint.iacr.org/2017/314`
6. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) Advances in Cryptology – CRYPTO 2005. Lecture Notes in Computer Science, vol. 3621, pp. 258–275. Springer, Berlin, Heidelberg (Aug 2005). https://doi.org/10.1007/11535218˙16
7. Brayton, R.K., Hachtel, G.D., McMullen, C.T., Sangiovanni-Vincentelli, A.L.: Logic minimization algorithms for vlsi synthesis. The Kluwer International Series in Engineering and Computer Science **2** (1984)
8. Champion, J., Wu, D.J.: Distributed broadcast encryption from lattices. In: Boyle, E., Mahmoody, M. (eds.) TCC 2024: 22nd Theory of Cryptography Conference, Part III. Lecture Notes in Computer Science, vol. 15366, pp. 156–189. Springer, Cham (Dec 2024). https://doi.org/10.1007/978-3-031-78020-2˙6
9. Chhatrapati, A., Hohenberger, S., Trombo, J., Vusirikala, S.: A performance evaluation of pairing-based broadcast encryption systems. In: Ateniese, G., Venturi, D. (eds.) ACNS 2022: 20th International Conference on Applied Cryptography and Network Security. Lecture Notes in Computer Science, vol. 13269, pp. 24–44. Springer, Cham (Jun 2022). https://doi.org/10.1007/978-3-031-09234-3˙2
10. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y. (ed.) Advances in Cryptology – CRYPTO'94. Lecture Notes in Computer Science, vol. 839, pp. 257–270. Springer, Berlin, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48658-5˙25
11. Dupin, A., Abelard, S.: Broadcast encryption using sum-product decomposition of boolean functions. IACR Communications in Cryptology (CiC) **1**(1), 18 (2024). https://doi.org/10.62056/av4fe0iuc
12. Fiat, A., Naor, M.: Broadcast encryption. In: Stinson, D.R. (ed.) Advances in Cryptology – CRYPTO'93. Lecture Notes in Computer Science, vol. 773, pp. 480–491. Springer, Berlin, Heidelberg (Aug 1994). https://doi.org/10.1007/3-540-48329-2˙40
13. Gong, J., Luo, J., Wee, H.: Traitor tracing with $N^{1/3}$-size ciphertexts and $O(1)$-size keys from k-Lin. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023, Part III. Lecture Notes in Computer Science, vol. 14006, pp. 637–668. Springer, Cham (Apr 2023). https://doi.org/10.1007/978-3-031-30620-4˙21
14. Goyal, R., Quach, W., Waters, B., Wichs, D.: Broadcast and trace with N^{ϵ} ciphertext size from standard assumptions. In: Boldyreva, A., Micciancio, D. (eds.) Advances in Cryptology – CRYPTO 2019, Part III. Lecture Notes in Computer Science, vol. 11694, pp. 826–855. Springer, Cham (Aug 2019). https://doi.org/10.1007/978-3-030-26954-8˙27
15. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: 28th Annual ACM Symposium on Theory of Computing. pp. 212–219. ACM Press (May 1996). https://doi.org/10.1145/237814.237866
16. Halevy, D., Shamir, A.: The LSD broadcast encryption scheme. In: Yung, M. (ed.) Advances in Cryptology – CRYPTO 2002. Lecture Notes in Computer Science, vol. 2442, pp. 47–60. Springer, Berlin, Heidelberg (Aug 2002). https://doi.org/10.1007/3-540-45708-9˙4

17. Kolonelos, D., Malavolta, G., Wee, H.: Distributed broadcast encryption from bilinear groups. In: Guo, J., Steinfeld, R. (eds.) Advances in Cryptology – ASIACRYPT 2023, Part V. Lecture Notes in Computer Science, vol. 14442, pp. 407–441. Springer, Singapore (Dec 2023). https://doi.org/10.1007/978-981-99-8733-7˙13
18. Majumder, A., Mukherjee, S.: Reinventing BrED: A practical construction: Formal treatment of broadcast encryption with dealership. IACR Communications in Cryptology (CiC) **1**(3), 47 (2024). https://doi.org/10.62056/ak5txl86bm
19. McCluskey, E.J.: Minimization of boolean functions. The Bell System Technical Journal **35**(6), 1417–1444 (1956)
20. Naor, D., Naor, M., Lotspiech, J.: Revocation and tracing schemes for stateless receivers. In: Kilian, J. (ed.) Advances in Cryptology – CRYPTO 2001. Lecture Notes in Computer Science, vol. 2139, pp. 41–62. Springer, Berlin, Heidelberg (Aug 2001). https://doi.org/10.1007/3-540-44647-8˙3
21. Nelson, V.P., Troy Nagle, H., Carroll, B.D., Irwin, J.D.: Digital logic circuit analysis and design. Prentice Hall (1995)
22. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: Ning, P., De Capitani di Vimercati, S., Syverson, P.F. (eds.) ACM CCS 2007: 14th Conference on Computer and Communications Security. pp. 195–203. ACM Press (Oct 2007). https://doi.org/10.1145/1315245.1315270
23. Petrick, S.R.: A direct determination of the irredundant forms of a boolean function from the set of prime implicants. Air Force Cambridge Res. Center Tech. Report pp. 56–110 (1956)
24. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. IEEE Computer Society Press (Nov 1994). https://doi.org/10.1109/SFCS.1994.365700
25. Wee, H.: Broadcast encryption with size $N^{1/3}$ and more from k-lin. In: Malkin, T., Peikert, C. (eds.) Advances in Cryptology – CRYPTO 2021, Part IV. Lecture Notes in Computer Science, vol. 12828, pp. 155–178. Springer, Cham, Virtual Event (Aug 2021). https://doi.org/10.1007/978-3-030-84259-8˙6
26. Wee, H.: Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In: Dunkelman, O., Dziembowski, S. (eds.) Advances in Cryptology – EUROCRYPT 2022, Part II. Lecture Notes in Computer Science, vol. 13276, pp. 217–241. Springer, Cham (May / Jun 2022). https://doi.org/10.1007/978-3-031-07085-3˙8
27. Wee, H.: Circuit ABE with $\mathsf{poly}(\mathsf{depth}, \lambda)$-sized ciphertexts and keys from lattices. In: Reyzin, L., Stebila, D. (eds.) Advances in Cryptology – CRYPTO 2024, Part III. Lecture Notes in Computer Science, vol. 14922, pp. 178–209. Springer, Cham (Aug 2024). https://doi.org/10.1007/978-3-031-68382-4˙6
28. Zhong, Y.: Test Pattern Compaction with Boolean Satisfiability Attack of Logic Locking (Aug 2024), `https://hal.science/hal-04679534`
29. Zhong, Y.: When Alice meets the "Hardware" Bob — Attacks and Solutions in the Digital World. Auburn University Electronic Theses and Dissertations (ETD) (2024)
30. Zhong, Y.: Optimizing test pattern compaction with boolean satisfiability attack. Chips **4**(4), 51 (2025)
31. Zhong, Y.: Variables for Free: Fault Injection Attack on MAYO via Valid Solutions. Cryptology ePrint Archive, Report 2025/1808 (2025), `https://eprint.iacr.org/2025/1808`

32. Zhong, Y.: Variables for Free: Faulting MAYO via Valid Solutions. In: The 2nd International Conference on Artificial Intelligence Computing and Systems (AICompS) 2025. pp. 1–8. Springer (2026)