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

Yadi Zhong

Auburn University, Auburn AL 36849, USA
yadi@auburn.edu

Abstract. Recently, Dupin and Abelard proposed a broadcast encryption scheme which outperforms 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 concrete Boolean functions. 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 approach ADC-BE, composed of two algorithms, AD-BE and AC-BE, can significantly optimize this worst-case complexity from $n/2$ down to 1 for a system of n users. This is efficient especially for large number of users in the system. Our proposed schemes combines the algebraic normal form, disjunctive normal form, and conjunctive normal form to optimize a Boolean function to its minimized representation. In addition, our approaches can be made secure against quantum adversaries and are therefore post-quantum, where both algorithms AD-BE and AC-BE require minimal assumptions based on existence of one-way function.

1 Introduction

Fiat and Naor [FN94] 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 non-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 predicate is the authorized user list. Over the year, additional properties for broadcast encryption has been considered. Traitor-tracing [NNL01, CFN94, GQWW19] 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 [CW24, KMW23], preventing single point of failure, has been analyzed extensively in recent years. Broadcast encryption can be built from various assumptions, such as Subset-Cover paradigm [NNL01, AI05, AL05, AK08], layered Subset Difference [HS02], bilinear groups and pairings [BGW05, KMW23, CHTV22, MM24], learning-with-errors and lattices [AY20, Wee24, Wee22], and k-Lin [Wee21, GLW23], etc.

Recently, Dupin and Abelard [DA24] proposed a broadcast encryption with sum-product decomposition from Boolean functions. It optimizes number of encryptions needed to broadcast a message m to authorized users only while simultaneously achieving full collusion resistance against any set of revoke users. It relies on Boolean optimizations, e.g.,

Quine-McCluskey algorithm and Petrick's method, to derive the minimized Boolean function, which in turn affects number of encryptions in this broadcast encryption scheme. It is built on minimal assumption with one-way function only. Dupin and Abelard's proposal of broadcast encryption from sum-product decomposition (Section 6.2 of [DA24]) outperforms subset difference-based broadcast encryption by Naor, Naor, and Lotspiech [NNL01] 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 [DA24], a Boolean function decomposition-based broadcast encryption with n users. We identify from the theoretical standpoint two hard functions which Boolean optimizations cannot be performed. We then propose ADC-BE, composed of two algorithms, AD-BE and AC-BE, to optimize this worst-case complexity from $n/2$ down to 1. Our contributions of the paper as follows:

1. We first identifies 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 AND monomials in DNF representations (or OR clauses in CNFs). We formalize them in Theorem 1 with proof.
2. We show that the worst-case bandwidth complexity of [DA24] 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 of [DA24] using the DNF representation for minimized Boolean function. Same issue occurs for CNF representation as well.
3. We demonstrate how this worst-case complexity can be drastically reduced when introducing combined representation from DNF or CNF with algebraic normal form. We propose two approaches, AD-BE and AC-BE. Both proposed broadcast encryption scheme can significantly optimize this worst-case complexity from $n/2$ down to 1 for a system of n users.

1.2 Paper Organization

The paper is organized as follows. In Section 2 we provide preliminaries. 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 with minimal assumptions to reduce this bandwidth bottleneck are presented in Section 4. 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 two binaries a, b is represented as $a||b$. 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.

Assuming the existence of one-way functions, there exists pseudo-random generators, pseudo-random function, and pseudo-random permutations [Hås90, ILL89, NR97].

Definition 2 (Pseudorandom Generator (PRG)). *Let $G : \{0,1\}^* \to \{0,1\}^*$ be a deterministic algorithm, ℓ be a polynomial such that for any input $s \in \{0,1\}^\lambda$, $G(s) \in \{0,1\}^{\ell(\lambda)}$. G is a pseudorandom generator if the following two conditions hold:*

1. *(Expansion) $\ell(\lambda) > \lambda$,*
2. *(Pseudorandomness) For any probabilistic polynomial-time distinguisher D, there exists a negligible function $\mathsf{negl}(\cdot)$ such that, the distributions $\{G(s), s \leftarrow \{0,1\}^\lambda\}$ and $\{r, r \xleftarrow{\$} \{0,1\}^{\ell(\lambda)}\}$ are indistinguishable,*

$$|\mathsf{Pr}[D(G(s)) = 1] - \mathsf{Pr}[D(r) = 1]| \leq \mathsf{negl}(\lambda).$$

Definition 3 (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*

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

With one-way function, one can build secure pseudorandom generators and psedorandom functions, which is the assumption made in [DA24]. In this paper, we follow this assumption of having the one-way function, and implication for psedorandom functions for our proposed broadcast encryption schemes in Section 4.

2.2 Broadcast Encryption

We adopt the same notion of broadcast encryption as in Dupin and Abelard [DA24]. It consists of four algorithms, $\mathsf{Setup}(\cdot), \mathsf{KenGen}(\cdot), \mathsf{Enc}(\cdot), \mathsf{Dec}(\cdot)$, as the following:

- $\mathsf{Setup}(n, 1^\lambda) \to mk$: With security parameter of unary string 1^λ, and number of users n (or an upper-bound), the broadcaster generates master key mk.
- $\mathsf{KenGen}(u, mk) \to k_u$: 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 not for the revoked users. It produces (h, c), a tuple of ciphertext c and header h, containing instruction for decryption for $\mathcal{AU}$.
- $\mathsf{Dec}(h, c, k_u) \to m$ else $\perp$: Receiver u utilizes header h to check if belonging 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 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 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 [DA24].

2.3 Boolean Function

Boolean function is the core for broadcast encryption construction by Dupin and Abelard. We first describe its definition and properties, then proceeds with its various representation forms.

Definition 4 (Boolean Function). *A Boolean function of n Boolean variables* $x = \{x_n, x_{n-1}, \ldots, x_1\}$ *is a function* $f(x) : \mathbb{F}_2^n \to \mathbb{F}_2$. *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 a Boolean function can have *don't care* values, represented as $*$. Depending on the minimization technique targeting on desired reduction format (e.g., CNF, 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 5 (Support). *The support of a Boolean function* $f(x) : \mathbb{F}_2^n \to \mathbb{F}_2$ *is the support of its* v_f, $s_f = \{x \in \mathbb{F}_2^n | f(x) = 1\}$.

Definition 6 (Weight). *The weight of a Boolean function* $f(x)$ *is the weight of its value vector,* $wt(f) = wt(v_f)$.

Definition 7 (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 $e_v \in \mathbb{F}_2$ and x^v is 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. CNF expression, the ANDing of summation (OR) of input literals, is generally 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 [DA24].

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 describes the core concept of minimizing Boolean functions of DNF format with Quine-McCluskey method, where detailed descriptions can be found in [NTNCI95, McC56]. 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 as well, with additional transformation through DeMorgan's theorems. In addition, Quine-McCluskey algorithm supports minimization of incompletely specified functions, where don't cares are 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 2^{nd} column. From these implicants of $(n-1)$ variables, combine adjacent ones into $(n-2)$-variable implicants and list in 3^{rd} column, and so on. This process is complete until no further implicants can be grouped, and it outputs 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. Petrick's method to derive minimal solution is often applied here [Pet56].

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 [BHMSV84] 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 [DA24]

In this section, we first describe broadcast encryption paper by Dupin and Abelard [DA24]. We proceed to discuss its bandwidth complexity with 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 the existence of one-way function to build pseudorandom function F with Definition 3. It uses two symmetric encryption schemes $\mathcal{E}$ and $\mathcal{E}_{\text{payload}}$, where $\mathcal{E}_{\text{payload}}$ can be an authenticated encryption. Their corresponding encryption algorithms are $\mathcal{D}, \mathcal{D}_{\text{payload}}$. This broadcast encryption is also post-quantum secure when cryptosystems selected for

$F, \mathcal{E}, \mathcal{E}_{\text{payload}}$ during concrete instantiations are resistant against quantum attackers and large-scale cryptanalytically-relevant quantum computers.

In this section, we consider n, a power of 2, as total number of users in the broadcast system and $l = \log_2(n)$. In practice, n may be selected to be the next power of 2 greater than the current number of users when l is not 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, $\{k_i^0\}_{i\in[l]}, \{k_i^1\}_{i\in[l]}$. Dupin and Abelard uses straightforward concatenation of binary bits to define labels $k_i^j = i||j, i \in [l], j \in \{0, 1\}$.

As the example in [DA24], emitter has 6 labels, $\{k_1^0, k_1^0, k_2^0, k_2^0, k_3^0, k_3^1\}$, when $n = 8, l = 3$.

Key Generation. 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 $u = u_1||u_2||\ldots||u_{l-1}||u_l$ with ordered (lex) 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. Emitter first identifies the list of valid users before broadcasting message m. To transmit a message m to the designated users only, broadcaster encrypts m with a ephemeral key k_e with $\mathcal{E}_{\text{payload}}(k_e, m)$. Key k_e is available to authorized users exclusively. 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 $\mathcal{AU}$. 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 $\mathsf{KenGen}(\cdot)$; (iii) ciphertext c has at least one encryption(s) of ephemeral key k_e that user z possesses its encryption key and thus able to later decrypt k_e, and then message m through k_e. On the other hand, if user z is revoked, i.e., 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 previous step $\mathsf{KenGen}(\cdot)$; (iii) ciphertext c has no encryption of ephemeral key k_e that user z could decrypt; and (iv) therefore, it cannot obtain k_e nor message m except with negligible probability. Both h and c are sent to recipients.

Continuing the example as in previous step, suppose broadcaster would like to send message m to users $1 = 001_2, 2 = 010_2, 5 = 101_2, 7 = 111_2$ while users $0, 3, 4, 6$ are revoked. The optimal Boolean function y with logic 1 output for users $1, 2, 5, 7$ but 0 output for the rest is expressed as

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

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

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

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

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

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 header h. Users who are the intended recipient of message m should have at least one of the symmetric keys obtained during $\mathsf{KenGen}(\cdot)$. 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. Therefore, it achieves full collusion-resistant against any number of revoke clients.

Decrypt. When recipient u receives {h,c}, he/she first confirms whether $y(u) = 1$ from binary vectors in header h. If $y(u) = 0$ instead, user u does not belong to authorized user list $\mathcal{AU}$ and proceeds to terminates the decryption process. Otherwise, recipient u finds a symmetric key based of h from stored key materials of $\mathsf{KenGen}(\cdot)$, where at least one symmetric key must be in possession of such valid user. 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 decryption with e_k, $\mathcal{D}_{\text{payload},k_e}(\mathcal{E}_{\text{payload},k_e}(m))$.

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

Dupin and Abelard's proposal of broadcast encryption from sum-product decomposition (Section 6.2 of [DA24]) outperforms subset difference-based broadcast encryption by Naor, Naor, and Lotspiech [NNL01] 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. 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. We would like to point out that this bandwidth is likewise affected by the length of header h, the other component from output tuple of $\mathsf{Enc}(\cdot)$ process, which also depends on number of AND clauses of y. Thus, we are able to conclude that bandwidth complexity of [DA24] is solely reliant on the optimal representation of Boolean function y. We summarize it in the following lemma.

Lemma 1. *Bandwidth requirement of* $\mathsf{Enc}(\cdot)$ *in Dupin and Abelard's broadcast encryption scheme is determined by number of products in Boolean function* y*, the binary encoding of 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.*

This sum-product decomposition approach is heavily dependent 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 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 a few concepts to characterize and compare any optimized function from its original expression.

For better analyzing minimal Boolean functions derived from various optimization techniques, such as Quine-McCluskey and ESPRESSO algorithms, we define compression ratio of any minimized expression with the following definitions.

Definition 8 (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 AND clauses in* f *in its minimized form.*

Note that this value is $\mathsf{cr}_1(f) \geq 1$ for any Boolean function f. It reaches $\mathsf{cr}_1(f) = 1$ when its minimized expression is identical to the un-optimized format – the canonical sum-of-products (CSOP) expression. This is equivalent to having $wt(v_f)$ number of AND clauses in its minimal expression. We define the compression ratio $\mathsf{cr}_1 = 2^n$ for a Boolean function of $wt = 2^n$ with n-bit input, which is a degenerated AND clause.

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

Definition 9 (Compression Ratio ($\mathbf{CR_{CNF}}$)). *Compression ratio* $\mathsf{cr}_0(f)$ *of a simplified Boolean function* f *is defined as the* $\overline{wt} = wt(\overline{v_f})$ *(number of outputs 0s in the truth table) divided by total number of OR clauses in* f *in its minimized form.*

Again, this value is $\mathsf{cr}_0(f) \geq 1$ for any Boolean function f. It is possible to have $\mathsf{cr}_0(f) = 1$ when its minimized CNF expression is identical to its corresponding un-optimized format – the canonical product-of-sums (CPOS) expression. This is identical to having $wt(\overline{v_f})$ number of OR clauses in its reduced expression. If a function has $\overline{wt} = 2^n$ for n inputs, we consider its compression ratio $\mathsf{cr}_0 = 2^n$.

Although minimization techniques like Quine-McCluskey and ESPRESSO allows to substantially reduce 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 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 functions 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 [DA24]. 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 three examples 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 $v_h = \{1,0,0,0\}, \{0,1,0,0\}, \{0,0,1,0\}$, or $v_h = \{0,0,0,1\}$. 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 in DNF, 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 $\{0, 0, 1, 1\}$. There exists no Boolean function with $wt = 3$ with $\mathsf{cr}_1 = 1$. All possible functions with $wt = 3$ has to be either of these 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 $\mathsf{cr}_1 = 3/2 > 1$. In addition, 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 $\mathsf{cr}_1 = 1$.

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 have $\mathsf{cr}_0 = 1$ (as defined above) 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 functions with $\overline{wt} = 3$, i.e., whose truth table is $\{1, 0, 0, 0\}, \{0, 1, 0, 0\}, \{0, 0, 1, 0\}$, 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 $\mathsf{cr}_0 = 1$.

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 achieving $wt_{max} = 4$ (and $\overline{wt}_{max} = 4$, respectively) for Boolean function subspace with $\mathsf{cr}_1 = 1$ (and $\mathsf{cr}_0 = 1$). Their truth tables are defined as follows:

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 for f_3, g_3, which are also the minimized expressions, are

$$\begin{cases} f_3 = \overline{x_3}\ \overline{x_2}x_1 \vee \overline{x_3}x_2\overline{x_1} \vee x_3\overline{x_2}\ \overline{x_1} \vee x_3x_2x_1 \\ g_3 = \overline{x_3}\ \overline{x_2}\ \overline{x_1} \vee \overline{x_3}x_2x_1 \vee x_3\overline{x_2}x_1 \vee x_3x_2\overline{x_1}. \end{cases}$$

Similarly, the minimized CNF formulation is expressed as below.

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

Further, one can demonstrate that any Boolean function with its weight $wt > 4$ (or $\overline{wt} > 4$, respectively) has compression ratio $\mathsf{cr}_1 > 1$ ($\mathsf{cr}_0 > 1$). This is due to the fact that at least two minterms (or Maxterms with output 0 instead) are logically adjacent to each other and can be reduced.

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.

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

From the above three examples, we summarize 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 $\overline{wt}_{max}$) 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 have $\mathsf{cr}_1 = 1$ (and $\mathsf{cr}_0 = 1$) with weight $wt = 2^n/2$, $\overline{wt} = 2^n/2$, and are balanced.*
2. *Functions f_n, g_n are disjoint. No common AND clauses in DNF expressions, nor any shared OR clauses in CNF 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. *Functions f_n, g_n's weight $wt = 2^n/2$, $\overline{wt} = 2^n/2$ are the maximum weight wt_{max} (and $\overline{wt}_{max}$ for $\mathsf{cr}_1 = 1$ (and $\mathsf{cr}_0 = 1$).*

Proof. We prove Theorem 1 by proof of induction. As we distinguish the inductive step with ℓ as an even or odd number, we have two initial conditions $\ell = 2$ and $\ell = 3$ in this proof by induction.

– 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

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

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

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

where sets S_2^0 and T_2^0 are equivalent to $S_2^0 = \{\prod_{v_i \in v} v_i = +1\}$, $T_2^0 = \{\prod_{v_i \in v} v_i = -1\}$.

– 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.

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

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 $S_3^1 = \{\prod_{v_i \in v} v_i = +1\}$, $T_3^1 = \{\prod_{v_i \in v} v_i = -1\}$. 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, $S_3^0 = S_3^1 = \{\prod_{v_i \in v} v_i = +1\}$, $T_3^0 = T_3^1 = \{\prod_{v_i \in v} v_i = -1\}$.

– 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

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

CNF expressions for hard functions f_ℓ, g_ℓ are

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

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 does 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

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

CNF formulations for hard functions f_{ℓ}, g_{ℓ} are

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

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:

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

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

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 $(\ell+1) = 2i+1$. We omit the proof details here.

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

The full collusion-resistant broadcast encryption proposed by Dupin and Abelard [DA24] achieves better encryption bandwidth requirement than Complete Subtree scheme, and

also better than Subset Difference by Naor, Naor, and Lotspiech [NNL01] in practical settings. On the other hand, the authors of [DA24] 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 [NNL01] as a corollary (Section 6.1 of [DA24]).

From Theorem 1 of Section 3.2, we have shown that, for any n Boolean variables, there are two hard functions f_n and g_n with $wt_{max} = 2^n/2$ (and $\overline{wt}_{max} = 2^n/2$) and $\mathsf{cr}_1 = 1$ (and $\mathsf{cr}_0 = 1$). Both such functions in regards to their DNF and CNF representations are already optimal and cannot be simplified. Due to their algebraic structure, hard functions f_n and g_n cannot be minimized by postulates and theorems of Boolean algebra, which are the foundations for any minimization techniques, e.g., Quine-McCluskey and ESPRESSO, etc. These hard functions accounts for the worst-case broadcast bandwidth of broadcast encryption scheme [DA24]. As Dupin and Abelard stated in [DA24], 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. The hard functions proven in Theorem 1 can be interpreted as the binary encoding of authorized user list $\mathcal{AU}$ of $r = 2^l/2 = n/2$ revoked users with worst-case bandwidth complexity of $r \log(n/r) = n/2$ due to maximum weight of $wt_{max} = 2^l/2 = n/2$. This upper-bound of $r \log(n/r) = n/2$ is indeed reached when $n/2$ revoked users and $n/2$ authorized users, respectively, forms the hard functions f_n and g_n, or vice-versa.

To further reduce this worst-case bandwidth requirement of $r \log(n/r) = n/2$ in [DA24], we proposed new constructions for broadcast encryption, denoted as ADC-BE, where ADC stands for ANF, DNF and CNF representations of Boolean functions. ADC-BE consists of two broadcast encryption algorithms – AD-BE and AC-BE. We explain both broadcast encryption schemes in Sections 4.1, 4.2. Both AD-BE and AC-BE rely on the same assumption as in [DA24], the existence of one-way function. We consider number of users n in a broadcast encryption setting be the power of 2, $n = 2^l$, as in Section 3.1. We use the identical pseudorandom function F with Definition 3, two symmetric encryption schemes $\mathcal{E}$ and $\mathcal{E}_{\text{payload}}$ by Dupin and Abelard (with decryption algorithms $\mathcal{D}, \mathcal{D}_{\text{payload}}$), where $\mathcal{E}_{\text{payload}}$ can be an authenticated encryption. 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 is also considered post-quantum whenever $F, \mathcal{E}, \mathcal{E}_{\text{payload}}$ are resistant against large-scale cryptanalytically-relevant quantum computers.

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 ($r \log(n/r) = n/2$) if either of them are the encoding of authorized user list $\mathcal{AU}$. Regardless of DNF or CNF representations, f_n, g_n cannot be minimized as any two AND monomials in DNF expression (OR clauses in CNF, respectively) are logically nonadjacent. When n is odd, DNFs for f_n and g_n are presented as the following.

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

DNFs for f_n and g_n under an even number n is 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 satisfies $\left\{\prod_{v_i \in v} v_i = -1\right\}$, 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 [DA24]. Our proposed AD-BE includes the following four algorithms.

Setup. Broadcaster first generates master key k_{PRF} from pseudorandom function F with security parameter 1^λ. Since pseudorandom function F is a keyed function, key k is sampled uniformly at random with source R, $k \xleftarrow{\$} R$, which can be discarded in a secure manner once generation of k_{PRF} is complete. Each user is uniquely identified by a l-bit binary. Broadcaster produces $2l + 2$ distinct labels with identical $2l$ labels as before, $\{k_i^0\}_{i\in[l]}, \{k_i^1\}_{i\in[l]}$, defined as concatenation of binary bits $k_i^j = i||j, i \in [l], j \in \{0, 1\}$. Two additional labels, a_f, a_g are selected at random and distinct from $\{k_i^0\}_{i\in[l]}, \{k_i^1\}_{i\in[l]}$.

As for example in [DA24], emitter has $2l + 2 = 8$ labels, $\{k_1^0, k_1^0, k_2^0, k_2^0, k_3^0, k_3^1, a_f, a_g\}$, when $n = 8, l = 3$.

Key Generation. When user u join this broadcast system, emitter provides user u a set of key materials, $k_u \leftarrow F_{k_{PRF}}(||_{i\in\mathcal{S}} k_i^{u_i})$ based on of $u = u_1||u_2||\ldots||u_{l-1}||u_l$ with ordered (lex) list $\mathcal{S} \subset [l]$. User u is also provided with an additional key $k_{a_f} \leftarrow F_{k_{PRF}}(a_f)$ if $\sum_{i\in[n]} u_i = 1$. If not, it must be true that u satisfies $\sum_{i\in[n]} u_i + 1 = 1$ and user has an additional key of $k_{a_g} \leftarrow F_{k_{PRF}}(a_g)$ instead.

User $u = 5 = 101_2$ from $n = 8$ would receive the following key materials in k_u and k_{a_g}:

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

Encrypt. Emitter possesses a list of valid users before broadcasting a message m. Broadcaster encrypts m with a ephemeral key k_e with $\mathcal{E}_{\text{payload}}(k_e, m)$. Key k_e is available to authorized users exclusively. This list of authorized user $\mathcal{AU} \subset [n]$ is represented by Boolean function y,

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

Emitter first check when y contains f_n or g_n as its subset or not. If it does include f_n (or g_n), emitter first transform y by updating the corresponding AND monomials in DNFs of f_n (or g_n) as don't-cares. Then, function y is reduced with standard minimization techniques in Boolean algebra, i.e., Quine-McCluskey method to obtain the minimal DNF representation of function y. Header h contains this optimal representation, along with two 1-bit indicators for f_n and g_n. Ciphtertext c contains $\mathcal{E}_{\text{payload}}(k_e, m)$ and multiple encryptions of ephemeral key k_e under valid key materials for authorized users $\mathcal{AU}$, including (possibly) an additional encryption of $\mathcal{E}_{k_{a_f}}(k_e)$ if $f_n \subset y$ or $\mathcal{E}_{k_{a_g}}(k_e)$ if $g_n \subset y$.

Decrypt. Decryption process is similar to Dupin and Abelard [DA24]. Recipient u also examines 1-bit indicators for f_n, g_n. If either function f_n, or g_n is used, receiver checks whether he/she has corresponding key k_{a_f} or k_{a_g}. If receiver do possess such key, he/she can also use it to obtain ephemeral key k_e and decrypt message m.

For any intended user, it is guaranteed to correctly decrypt ephemeral key k_e and message m. On the other hand, if user z is revoked, then (i) function y on input z has

output $f(z) = 0$ instead; (ii) header h does not contain any binary patterns of key labels which user z received, nor does it contain the corresponding f_n or g_n if z has its relevant key; (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 except with negligible probability. Therefore, this broadcast encryption scheme has full collusion resistance as none of the revoke user possess any valid keys for decrypting k_e.

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

Hard functions f_n, g_n has 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 has complex representations in CNF, their equivalent representations in ANF is much straightforward, $f_n = \sum_{i \in [n]} x_i$ and $g_n = \sum_{i \in [n]} x_i + 1$. We include ANF representations of f_n, g_n as well as CNF representations to built AC-BE, a complement of AD-BE, where output are focused on 0s instead. It contains the following algorithms.

Setup. Setup procedure is identical to AD-BE. However, the $2l+2$ labels, including a_f, a_g, are selected completely separate (and preferably from random) from those in AD-BE. AC-BE and AD-BE do no share any labels of same value, preventing possible collusions from revoked users.

Key Generation. Setup procedure is identical to AD-BE and we refer Section 4.1 for details. User u also receives k_{a_f} if $\sum_{i \in [n]} u_i = 0$; or k_{a_g} if $\sum_{i \in [n]} u_i + 1 = 0$. Please note that the set of possible keys for AC-BE does not reuse any keys from AD-BE, where labels from the previous step are different from AD-BE.

Encrypt. Encryption step differs from AD-BE by representing $\mathcal{AU}$ with CNF representation of Boolean function y with output 0 for authorized receiver and output 1 for revoked ones.

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

Emitter then confirm if y contains the CNFs of f_n or g_n as its subset. If it does have f_n (or g_n), emitter first transform y by updating the corresponding OR monomials in CNFs of f_n (or g_n) as don't-cares. Then, function y is reduced with standard minimization techniques to a minimal CNF representation instead. Header h contains this optimal CNF, along with two 1-bit indicators for f_n and g_n. Ciphtertext c contains $\mathcal{E}_{\text{payload}}(k_e, m)$ and multiple encryptions of ephemeral key k_e under valid key materials for authorized users $\mathcal{AU}$, including (possibly) an additional encryption of $\mathcal{E}_{k_{a_f}}(k_e)$ if $f_n \subset y$ or $\mathcal{E}_{k_{a_g}}(k_e)$ if $g_n \subset y$.

Decrypt. Decryption process is almost identical to AD-BE. Recipient u additionally checks from indicators for f_n, g_n. If either function f_n, or g_n is used, receiver checks whether he/she has corresponding key k_{a_f} or k_{a_g}. If receiver do possess such key, he/she can also use it to obtain ephemeral key k_e and decrypt message m.

Any intended recipient can decrypt ephemeral key k_e and message m. Revoked user z, however, has (i) evaluation of $f(z) = 1$ instead; (ii) none of binary patterns of key labels which user z received is included in header h, which may contain the key (either k_{a_f} or k_{a_g}) that user z does not have; (iii) no encryption of ephemeral key k_e that user z could decrypt; and (iv) he/she cannot obtain k_e nor message m except with negligible probability. In the like manner, this broadcast encryption scheme has full collusion resistance as none of the revoke user a single valid keys for decrypting k_e.

4.3 Bandwidth Reduction in ADC-BE

The proposed ADC-BE allows significant reduction of bandwidth which could not be achieved in [DA24] before. If f_n is the Boolean function representing list $\mathcal{AU}$, we achieve a bandwidth of a single encryption of the ephemeral key compare to the previous $n/2$ encryptions of [DA24]. The identical reduction is true for g_n, where also one encryption of ephemeral key k_e is sufficient in either proposed AD-BE, or AC-BE, in contrast to the $n/2$ encryptions of [DA24]. In addition, if number of revoke users r are fewer than $n/2$, any function y with f_n (or g_n) as its subset can include the encryption of ephemeral key k_e with the corresponding key k_{a_f} or k_{a_g} to account for the $n/2$ monomials of DNF for AD-BE, or OR clauses in CNF for AC-BE, and then solve the minimal expression of y replacing these $n/2$ clauses as don't cares. It should be noted that this optimization approach can lead to less time for Boolean optimizations as fewer minterms (for DNF) or Maxterms (CNF) are included. Thus, inclusion of f_n and g_n in ANF representations offers further optimization of the encryption bandwidth for broadcast encryption schemes.

5 Conclusion

In this paper, we target the worst-case upper-bound in bandwidth requirement of the recent work of Dupin and Abelard [DA24], which achieve 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 [DA24]. Our proposed approaches AD-BE and AC-BE can significantly optimize this worst-case complexity from $n/2$ down to 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.

Acknowledgements The author thanks God for this research idea and writing.

References

[AI05] Nuttapong Attrapadung and Hideki Imai. Graph-decomposition-based frameworks for subset-cover broadcast encryption and efficient instantiations. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 100–120. Springer, Berlin, Heidelberg, December 2005.

[AK08] Per Austrin and Gunnar Kreitz. Lower bounds for subset cover based broadcast encryption. In Serge Vaudenay, editor, *AFRICACRYPT 08*, volume 5023 of *LNCS*, pages 343–356. Springer, Berlin, Heidelberg, June 2008.

[AL05] Sarang Aravamuthan and Sachin Lodha. An optimal subset cover for broadcast encryption. In Subhamoy Maitra, C. E. Veni Madhavan, and Ramarathnam Venkatesan, editors, *INDOCRYPT 2005*, volume 3797 of *LNCS*, pages 221–231. Springer, Berlin, Heidelberg, December 2005.

[AY20] Shweta Agrawal and Shota Yamada. Optimal broadcast encryption from pairings and LWE. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 13–43. Springer, Cham, May 2020.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275. Springer, Berlin, Heidelberg, August 2005.

[BHMSV84] Robert K Brayton, Gary D Hachtel, Curtis T McMullen, and Alberto L Sangiovanni-Vincentelli. Logic minimization algorithms for vlsi synthesis. *The Kluwer International Series in Engineering and Computer Science*, 2, 1984.

[CFN94] Benny Chor, Amos Fiat, and Moni Naor. Tracing traitors. In Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 257–270. Springer, Berlin, Heidelberg, August 1994.

[CHTV22] Arush Chhatrapati, Susan Hohenberger, James Trombo, and Satyanarayana Vusirikala. A performance evaluation of pairing-based broadcast encryption systems. In Giuseppe Ateniese and Daniele Venturi, editors, *ACNS 22International Conference on Applied Cryptography and Network Security*, volume 13269 of *LNCS*, pages 24–44. Springer, Cham, June 2022.

[CW24] Jeffrey Champion and David J. Wu. Distributed broadcast encryption from lattices. In *TCC 2024, Part III*, LNCS, pages 156–189. Springer, Cham, November 2024.

[DA24] Aurélien Dupin and Simon Abelard. Broadcast encryption using sum-product decomposition of boolean functions. *CiC*, 1(1):18, 2024.

[FN94] Amos Fiat and Moni Naor. Broadcast encryption. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 480–491. Springer, Berlin, Heidelberg, August 1994.

[GLW23] Junqing Gong, Ji Luo, and Hoeteck Wee. Traitor tracing with $N^{1/3}$-size ciphertexts and $O(1)$-size keys from k-Lin. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part III*, volume 14006 of *LNCS*, pages 637–668. Springer, Cham, April 2023.

[GQWW19] Rishab Goyal, Willy Quach, Brent Waters, and Daniel Wichs. Broadcast and trace with N^{ϵ} ciphertext size from standard assumptions. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 826–855. Springer, Cham, August 2019.

[Hås90] Johan Håstad. Pseudo-random generators under uniform assumptions. In *22nd ACM STOC*, pages 395–404. ACM Press, May 1990.

[HS02] Dani Halevy and Adi Shamir. The LSD broadcast encryption scheme. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 47–60. Springer, Berlin, Heidelberg, August 2002.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *21st ACM STOC*, pages 12–24. ACM Press, May 1989.

[KMW23] Dimitris Kolonelos, Giulio Malavolta, and Hoeteck Wee. Distributed broadcast encryption from bilinear groups. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part V*, volume 14442 of *LNCS*, pages 407–441. Springer, Singapore, December 2023.

[McC56] Edward J McCluskey. Minimization of boolean functions. *The Bell System Technical Journal*, 35(6):1417–1444, 1956.

[MM24] Avishek Majumder and Sayantan Mukherjee. Reinventing BrED: A practical construction: Formal treatment of broadcast encryption with dealership. *CiC*, 1(3):47, 2024.

[NNL01] Dalit Naor, Moni Naor, and Jeffery Lotspiech. Revocation and tracing schemes for stateless receivers. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 41–62. Springer, Berlin, Heidelberg, August 2001.

[NR97] Moni Naor and Omer Reingold. On the construction of pseudo-random permutations: Luby-Rackoff revisited (extended abstract). In *29th ACM STOC*, pages 189–199. ACM Press, May 1997.

[NTNCI95] Victor P Nelson, H Troy Nagle, Bill D Carroll, and J David Irwin. *Digital logic circuit analysis and design*. Englewood Cliffs: Prentice Hall, 1995.

[Pet56] Stanley R Petrick. A direct determination of the irredundant forms of a boolean function from the set of prime implicants. *Air Force Cambridge Res. Center Tech. Report*, pages 56–110, 1956.

[Wee21] Hoeteck Wee. Broadcast encryption with size $N^{1/3}$ and more from k-lin. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 155–178, Virtual Event, August 2021. Springer, Cham.

[Wee22] Hoeteck Wee. Optimal broadcast encryption and CP-ABE from evasive lattice assumptions. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part II*, volume 13276 of *LNCS*, pages 217–241. Springer, Cham, May / June 2022.

[Wee24] Hoeteck Wee. Circuit ABE with $\mathsf{poly}(\mathrm{depth}, \lambda)$-sized ciphertexts and keys from lattices. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part III*, volume 14922 of *LNCS*, pages 178–209. Springer, Cham, August 2024.