Adaptively Secure ABE for DFA from k-Lin and More

Junqing Gong[1,2,⋆] and Hoeteck Wee[2,⋆⋆]

[1] East China Normal University

[2] CNRS, ENS and PSL

{jgong,wee}@di.ens.fr

Abstract. In this work, we present:

- the first adaptively secure ABE for DFA from the k-Lin assumption in prime-order bilinear groups; this resolves one of open problems posed by Waters [CRYPTO'12];
- the first ABE for NFA from the k-Lin assumption, provided the number of accepting paths is smaller than the order of the underlying group; the scheme achieves selective security;
- the first compact adaptively secure ABE (supporting unbounded multi-use of attributes) for branching programs from the k-Lin assumption, which generalizes and simplifies the recent result of Kowalczyk and Wee for boolean formula (NC1) [EUROCRYPT'19].

Our adaptively secure ABE for DFA relies on a new combinatorial mechanism avoiding the exponential security loss in the number of states when naively combining two recent techniques from CRYPTO'19 and EUROCRYPT'19. This requires us to design a selectively secure ABE for NFA; we give a construction which is sufficient for our purpose and of independent interest. Our ABE for branching programs leverages insights from our ABE for DFA.

1 Introduction

Attribute-based encryption (ABE) [19,12] is an advanced form of public-key encryption that supports fine-grained access control for encrypted data. Here, ciphertexts are associated with an attribute x and keys with a policy Γ; decryption is possible only when $\Gamma(x) = 1$. One important class of policies we would like to support are those specified using deterministic finite automata (DFA). Such policies capture many real-world applications involving simple computation on data of unbounded size such as network logging application, tax returns and virus scanners.

Since the seminal work of Waters [21] introducing ABE for DFA and providing the first instantiation from pairings, substantial progress has been made in the design and analysis of ABE schemes for DFA [4,5,1,11,2,3], proving various trade-offs between security assumptions and security guarantees. However, two central problems posed by Waters [21] remain open. The first question pertains to security and assumptions:

⋆ Supported by NSFC-ISF Joint Scientific Research Program (61961146004) and ERC Project aSCEND (H2020 639554).

⋆⋆ Supported by ERC Project aSCEND (H2020 639554).

> ***Q1**: Can we build an ABE for DFA with adaptive security from static assumptions in bilinear groups, notably the k-Lin assumption in prime-order bilinear groups?*

From both a practical and theoretical stand-point, we would like to base cryptography on weaker and better understood assumptions, as is the case with the k-Lin assumption, while also capturing more realistic adversarial models, as is the case with adaptive security. Prior ABE schemes for DFA achieve either adaptive security from less desirable q-type assumptions [21,4,5,1], where the complexity of the assumption grows with the length of the string x, or very recently, selective security from the k-Lin assumption [2,11]. Indeed, this open problem was reiterated again in the latter work [11], emphasizing a security loss that is polynomial (and not exponential) in the size of the DFA.

The next question pertains to expressiveness:

> ***Q2**: Can we build an ABE for nondeterministic finite automata (NFA) with a polynomial dependency on the NFA size?*

The efficiency requirement rules out the naive approach of converting a NFA to a DFA, which incurs an exponential blow-up in size. Here, we do not know any construction even if we only require selective security under q-type assumptions. Partial progress was made very recently by Agrawal *et al.* [3] in the more limited secret-key setting, where encryption requires access to the master secret key. Throughout the rest of this work, we refer only to the standard public-key setting for ABE, and where the adversary can make an a-priori unbounded number of secret key queries.

1.1 Our Results

In this work, we address the afore-mentioned open problems:

- We present an adaptively secure ABE for DFA from the k-Lin assumption in prime-order bilinear groups, which affirmatively answers the first open problem. Our scheme achieves ciphertext and key sizes with linear complexity, as well as security loss that is polynomial in the size of the DFA and the number of key queries. Concretely, over the binary alphabet and under the SXDH (=1-Lin) assumption, our ABE for DFA achieves ciphertext and key sizes 2–3 times that of Waters' scheme (cf. Fig 4), while simultaneously improving on both the assumptions and security guarantees.
- We present a selectively secure ABE for NFA also from the k-Lin assumption, provided the number of accepting paths is smaller than p, where p is the order of the underlying group. We also present a simpler ABE for NFA with the same restriction from the same q-type assumption used in Waters' ABE for DFA. Both ABE schemes for NFA achieve ciphertext and key sizes with linear complexity.
- Finally, we present the first compact adaptively secure ABE for branching programs from the k-Lin assumption, which generalizes and simplifies the recent result of Kowalczyk and Wee [15] for boolean formula (NC1). Here, "compact" is also referred to as "unbounded multi-use of attributes" in [5]; each attribute/input bit can

appear in the formula/program an unbounded number of times. Our construction leverages insights from our ABE for DFA, and works directly with any layered branching program and avoids both the pre-processing step in the latter work for transforming boolean formulas into balanced binary trees of logarithmic depth, as well as the delicate recursive pebbling strategy for binary trees.

We summarize the state of the art of ABE for DFA, NFA and branching programs in Fig 1, 2, 3, respectively.

In the rest of this section, we focus on our three ABE schemes that rely on the k-Lin assumption, all of which follow the high-level proof strategy in [11,15]. We design a series of hybrids that traces through the computation, and the analysis carefully combines (i) a "nested, two-slot" dual system argument [20,16,17,18,13,8], (ii) a new combinatorial mechanism for propagating entropy along the NFA computation path, and (iii) the piecewise guessing framework [14,15] for achieving adaptive security. We proceed to outline and motivate several of our key ideas. From now on, we use GWW to refer to the ABE for DFA by Gong *et al.* [11].

Adaptively secure ABE for DFA. Informally, the piecewise guessing framework [14,15] for ABE adaptive security says that if we have a selectively secure ABE scheme where proving indistinguishability of every pair of adjacent hybrids requires only knowing $\log L$ bits of information about the challenge attribute x, then the same scheme is adaptively secure with a security loss of L. Moreover, when combined with the dual system argument, it suffices to consider selective security when the adversary only gets a single key corresponding to a single DFA.

In the GWW security proof, proving indistinguishability of adjacent hybrids requires knowing the subset of DFA states that are reachable from the accept states by "back-tracking" the computation. This corresponds to $\log L = Q$ —we need Q bits to specify an arbitrary subset of $[Q]$— and a security loss of 2^Q. Our key insight for achieving adaptive security is that via a suitable transformation to the DFA, we can ensure that the subset of reachable states per input are always singleton sets, which corresponds to $\log L = \log Q$ and a security loss of Q. The transformation is very simple: run the DFA "in reverse"! That is, start from the accept states, read the input bits in reverse order and the transitions also in reverse, and accept if we reach the start state. It is easy to see that this actually corresponds to an NFA computation, which means that we still need to design a selectively secure ABE for NFA. Also, back-tracking along this NFA corresponds to normal computation in the original DFA, and therefore always reaches singleton sets of states during any intermediate computation.

ABE for NFA. Next, we sketch our ABE for NFA, which uses an asymmetric bilinear group (G_1, G_2, G_T, e) of prime order p where $e : G_1 \times G_2 \to G_T$. As in Waters' ABE for DFA [21], an encryption of $x = (x_1, \ldots, x_\ell) \in \{0,1\}^\ell$ contains random scalars $s_0, \ldots, s_\ell \leftarrow \mathbb{Z}_p$ in the exponent in G_1. In the secret key, we pick a random scalar $d_u \leftarrow \mathbb{Z}_p$ for each state $u \in [Q]$. We can now describe the invariant used during decryption with g_1, g_2 being respective generators of G_1, G_2:

– In Waters' ABE for DFA, if the computation reaches a state $u_i \in [Q]$ upon reading $x_1, \ldots, x_i$, decryption computes $e(g_1, g_2)^{s_i d_{u_i}}$. In particular, the scheme allows the decryptor to compute the ratios

$$e(g_1, g_2)^{s_j d_v - s_{j-1} d_u}, \ \forall j \in [\ell], u \in [Q], v = \delta(u, x_j) \in [Q] \tag{1}$$

where $\delta : [Q] \times \{0,1\} \to [Q]$ is the DFA transition function.
– The natural way to extend (1) to account for non-deterministic transitions in an NFA is to allow the decryptor to compute

$$e(g_1, g_2)^{s_j d_v - s_{j-1} d_u}, \ \forall j \in [\ell], u \in [Q], v \in \delta(u, x_j) \subseteq [Q] \tag{2}$$

where $\delta : [Q] \times \{0,1\} \to 2^{[Q]}$ is the NFA transition function. As noted by Waters [21], such an ABE scheme for NFA is broken via a so-called "back-tracking attack", which we describe in the full paper.
– In our ABE for NFA, we allow the decryptor to compute

$$e(g_1, g_2)^{s_j (\sum_{v \in \delta(u, x_j)} d_v) - s_{j-1} d_u}, \ \forall j \in [\ell], u \in [Q] \tag{3}$$

A crucial distinction between (3) and (2) is that the decryptor can only compute *one* quantity for each j, u in the former (as is the case also in (1)), and up to Q quantities in the latter. The ability to compute multiple quantities in (2) is exactly what enables the back-tracking attack.

We clarify that our ABE for NFA imposes an extra restriction on the NFA, namely that the total number of accepting paths[3] be non-zero $\bmod p$ for accepting inputs; we use NFA$^{\oplus_p}$ to denote such NFAs. In particular, this is satisfied by standard NFA where the total number of accepting paths is less than p for all inputs. This is in general a non-trivial restriction since the number of accepting paths for an arbitrary NFA can be as large as Q^ℓ. Fortunately, for NFAs obtained by running a DFA "in reverse", the number of accepting paths is always either 0 or 1.

Indeed, the above idea, along with a suitable modification of Waters' proof strategy, already yields our selectively secure ABE for NFA$^{\oplus_p}$ under q-type assumptions in asymmetric bilinear groups of prime order p. We defer the details to the full paper.

– To obtain a selectively secure scheme based on k-Lin, we apply the same modifications as in GWW [11]. For the proof of security, entropy propagation is defined via back-tracking the NFA computation, in a way analogous to that for back-tracking the DFA computation.
– To obtain an adaptively secure scheme based on k-Lin, we adapt the selectively secure scheme to the piecewise guessing framework [15]. One naive approach is to introduce a new semi-functional space. In contrast, we introduce one extra components into master public key, secret key and ciphertext, respectively. With the extra components, we can avoid adding a new semi-functional subspace, by reusing an existing subspace as shown in previous unbounded ABE in [8]. Under

[3] An accepting path on input $x \in \{0,1\}^\ell$ is described by a sequence of states $u_0, \ldots, u_\ell \in [Q]$ where u_0 is the start state, u_ℓ is an accept state and $u_j \in \delta(u_{j-1}, x_j)$ for all $j \in [\ell]$.

reference	assumption	security	$\|\mathsf{sk}\|$	$\|\mathsf{ct}\|$
[21]	q-type	selective	$O(Q)$	$O(\ell)$
[5,4,1]	q-type + k-Lin	adaptive ✓	$O(Q)$	$O(\ell)$
[11]	k-Lin ✓	selective	$O(Q)$	$O(\ell)$
[3]	k-Lin ✓	selective*	$O(Q^2)$	$O(\ell^3)$
ours	k-Lin ✓	adaptive ✓	$O(Q)$	$O(\ell)$

Fig. 1. Summary of ABE schemes for DFA. In the table, Q is the number of states in the DFA associated with sk and ℓ is the length of x associated with ct, and where $|\Sigma| = O(1)$.

reference	$\|\mathsf{sk}\|$	$\|\mathsf{ct}\|$	type of NFA	public key?	assumption
[2]	$\mathsf{poly}(Q)$	$\mathsf{poly}(\ell)$	standard ✓		LWE ✓
ours	$O(Q)$	$O(\ell)$	$\mathsf{NFA}^{\oplus p}$	✓	q-type
	$O(Q)$	$O(\ell)$	$\mathsf{NFA}^{\oplus p}$	✓	k-Lin✓

Fig. 2. Summary of ABE schemes for NFA. In the table, Q is the number of states in the NFA associated with sk and ℓ is the length of x associated with ct.

reference	assumption	compact?
[7]	k-Lin ✓	
[5]	q-type + k-Lin	✓
	k-Lin ✓	
ours	k-Lin ✓	✓

Fig. 3. Summary of adaptively secure ABE schemes for branching programs (BP). Here "compact" is also referred to "unbounded multi-use" in [5].

k-Lin assumption, our technique roughly saves $k \cdot \ell$ elements in the ciphertext and $k \cdot (2|\Sigma| + 2)Q$ elements in the secret key over the general apporach. This way, we obtain ciphertext and key sizes that are almost the same as those in the GWW selectively secure scheme.

ABE for branching programs. We build our compact adaptively secure ABE for branching program (BP) in two steps analogous to our adaptively secure ABE for DFA. In particular, we first show how to transform branching programs to a subclass of non-deterministic branching programs (NBP) and construct adaptively secure ABE for such class of NBP. Note that the latter is sufficient to capture a special BP with permutation transition function (without transforming BP to NBP) and readily simplify the result of Kowalczyk and Wee [15] for boolean formula (NC1).

1.2 Technical Overview

We start by recalling the standard definitions of DFA and NFA using vector-matrix notation: that is, we describe the start and accept states using the character vectors, and specify the transition function via a transition matrix. The use of vector-matrix notation enables a more compact description of our ABE schemes, and also clarifies the connection to branching programs.

reference	$\|\mathsf{ct}\|$	$\|\mathsf{sk}\|$	assumption	security
[21]	$(2\ell+3)\|G_1\|$	$(3\|\Sigma\|Q+4)\|G_2\|$	q-type	selective
[5]	$((2k+2)\ell+6k+6)\|G_1\|$	$((3k+3)\|\Sigma\|Q+5k+5)\|G_2\|$	q-type + k-Lin	adaptive ✓
	$(3\ell+12)\|G_1\|$	$(6\|\Sigma\|Q+10)\|G_2\|$	q-type + SXDH	adaptive ✓
[11]	$((3k+1)\ell+4k+1)\|G_1\|$	$((4k+2)\|\Sigma\|Q+(3k+1)Q+2k+1)\|G_2\|$	k-Lin ✓	selective
	$(4\ell+5)\|G_1\|$	$(6\|\Sigma\|Q+4Q+3)\|G_2\|$	SXDH✓	selective
ours	$((3k+1)\ell+6k+2)\|G_1\|$	$((4k+2)\|\Sigma\|Q+(5k+2)Q+2k+1)\|G_2\|$	k-Lin ✓	adaptive ✓
	$(4\ell+8)\|G_1\|$	$(6\|\Sigma\|Q+7Q+3)\|G_2\|$	SXDH ✓	adaptive ✓

Fig. 4. Concrete parameter sizes of pairing-based ABE schemes for DFA. Note that [21,11] are selectively secure whereas our scheme is adaptively secure; [3] is omitted from the table since the ciphertext and key sizes are asymptotically larger, see Fig 1. In the table, Q is the number of states in the DFA, Σ indicates the alphabet, ℓ is the length of input x. All the schemes work over bilinear groups (G_1, G_2, G_T, e) of prime order p where $e : G_1 \times G_2 \to G_T$. We note that all the schemes shown in the table have mpk of $O(|\Sigma|)$ group elements. In the $|\mathsf{ct}|$-column, we omit one G_T element. In the **assumption** column, SXDH means 1-Lin.

NFA, DFA, NFA$^{\oplus_p}$. An NFA Γ is specified using $(\, Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma\in\Sigma}, \mathbf{u}, \mathbf{f}\,)$ where Σ is the alphabet and

$$Q \in \mathbb{N}; \quad \mathbf{M}_\sigma \in \{0,1\}^{Q\times Q}, \forall \sigma \in \Sigma; \quad \mathbf{u}, \mathbf{f} \in \{0,1\}^{1\times Q}.$$

The NFA Γ accepts an input $x = (x_1, \ldots, x_\ell) \in \Sigma^\ell$, denoted by $\Gamma(x) = 1$, if

$$\mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_2}\mathbf{M}_{x_1}\mathbf{u}^\top > 0 \tag{4}$$

and rejects the input otherwise, denoted by $\Gamma(x) = 0$. We will also refer to the quantity $\mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_2}\mathbf{M}_{x_1}\mathbf{u}^\top$ as the number of accepting paths for x. The above relation (4) is equivalent to

$$\mathbf{u}\mathbf{M}_{x_1}^\top\mathbf{M}_{x_2}^\top \cdots \mathbf{M}_{x_\ell}^\top\mathbf{f}^\top > 0$$

The unusual choice of notation is to simplify the description of our ABE scheme. Let $\mathcal{E}_Q$ be the collection of Q elementary row vectors of dimension Q.

- A DFA Γ is a special case of NFA where $\mathbf{u} \in \mathcal{E}_Q$ and each column in every matrix $\mathbf{M}_\sigma$ is an elementary column vector (i.e., contains exactly one 1).
- An NFA$^{\oplus_p}$, parameterized by a prime p, is the same as an NFA except we change the accept criterion in (4) to:

  $$\mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_2}\mathbf{M}_{x_1}\mathbf{u}^\top \neq 0 \bmod p$$

  Note that this coincides with the standard NFA definition whenever the total number of accepting paths for all inputs is less than p.

Throughout the rest of this work, when we refer to NFA, we mean NFA$^{\oplus_p}$ unless stated otherwise.

ABE for NFA$^{\oplus_p}$. Following our overview in Section 1.1, an encryption of $x = (x_1, \ldots, x_\ell) \in \Sigma^\ell$ contains random scalars $s_0, \ldots, s_\ell$ in the exponent, where the plaintext is masked by $e(g_1, g_2)^{s_\ell \alpha}$. To generate a secret key for an NFA$^{\oplus_p}$ Γ, we first pick $\mathbf{d} = (d_1, \ldots, d_Q) \leftarrow \mathbb{Z}_p^Q$ as before. We allow the decryptor to compute the following quantities in the exponent over G_T:

$$\begin{aligned} &\text{(i)} \;\; s_\ell(\alpha \mathbf{f} - \mathbf{d}) \\ &\text{(ii)} \;\; s_j \mathbf{d}\mathbf{M}_{x_j} - s_{j-1}\mathbf{d}, \;\; \forall j \in [\ell] \;\; \text{(corresponds to (3))} \\ &\text{(iii)} \;\; s_0 \mathbf{d}\mathbf{u}^\top \end{aligned} \tag{5}$$

If we write $\mathbf{u}_{j,x}^\top = \mathbf{M}_{x_j} \cdots \mathbf{M}_{x_1} \mathbf{u}^\top$ for all $j \in [\ell]$ and $\mathbf{u}_{0,x} = \mathbf{u}$, then we have

$$s_\ell \alpha \cdot \mathbf{f}\mathbf{u}_{\ell,x}^\top = \overbrace{s_\ell(\alpha \mathbf{f} - \mathbf{d})}^{\text{(i)}} \cdot \mathbf{u}_{\ell,x}^\top + \Big(\sum_{j=1}^{\ell} \overbrace{(s_j \mathbf{d}\mathbf{M}_{x_j} - s_{j-1}\mathbf{d})}^{\text{(ii)}} \cdot \mathbf{u}_{j-1,x}^\top \Big) + \overbrace{s_0 \mathbf{d}\mathbf{u}_{0,x}^\top}^{\text{(iii)}}$$

This means that whenever $\mathbf{f}\mathbf{u}_{\ell,x}^\top \neq 0 \bmod p$, as is the case when $\Gamma(x) = 1$, the decryptor will be able to recover $e(g_1, g_2)^{s_\ell \alpha}$.

Indeed, it is straight-forward to verify that the following ABE scheme satisfies the above requirements, where $[\cdot]_1, [\cdot]_2, [\cdot]_T$ denote component-wise exponentiations in respective groups G_1, G_2, G_T [10].

$$\begin{aligned} \mathsf{msk} &= \big(w_{\text{start}}, w_{\text{end}}, z, \{w_\sigma\}_{\sigma \in \Sigma}, \alpha \big) \\ \mathsf{mpk} &= \big([w_{\text{start}}]_1, [w_{\text{end}}]_1, [z]_1, \{[w_\sigma]_1\}_{\sigma \in \Sigma}, [\alpha]_T \big) \\ \mathsf{ct}_x &= \begin{pmatrix} [s_0]_1, \; [s_0 w_{\text{start}}]_1 \\ \{ [s_j]_1, \; [s_{j-1} z + s_j w_{x_j}]_1 \}_{j \in [\ell]} \\ [s_\ell]_1, \; [s_\ell w_{\text{end}}]_1, \; [s_\ell \alpha]_T \cdot m \end{pmatrix} \\ \mathsf{sk}_\Gamma &= \begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + w_{\text{start}} \mathbf{r}\mathbf{u}^\top]_2, \; [\mathbf{r}\mathbf{u}^\top]_2 \\ \{ [-\mathbf{d} + z\mathbf{r}]_2, \; [\mathbf{d}\mathbf{M}_\sigma + w_\sigma \mathbf{r}]_2, \; [\mathbf{r}]_2 \}_{\sigma \in \Sigma} \\ [\alpha \mathbf{f} - \mathbf{d} + w_{\text{end}} \mathbf{r}]_2, \; [\mathbf{r}]_2 \end{pmatrix}, \quad \mathbf{d}, \mathbf{r} \leftarrow \mathbb{Z}_p^{1 \times Q} \end{aligned} \tag{6}$$

In the full paper, we prove that this scheme is selectively secure under ℓ-EBDHE assumption; this is the assumption underlying Waters' selectively secure ABE for DFA [21].

Selective security from k-Lin. Following the GWW proof strategy which in turn builds on the dual system argument, we design a series of games $\mathsf{G}_0, \ldots, \mathsf{G}_\ell$ such that in G_i, the quantities s_i and $\mathbf{d}$ have some extra entropy in the so-called semi-functional space (which requires first modifying the above scheme). The entropy in $\mathbf{d}$ is propagated from G_0 to G_1, then G_2, and finally to G_ℓ via a combination of a computational and combinatorial arguments. In G_ℓ, we will have sufficient entropy to statistically mask α in the secret key, which allows us to argue that $e(g_1, g_2)^{s_\ell \alpha}$ statistically masks the plaintext. In this overview, we focus on the novel component, namely the combinatorial argument which exploits specific properties of our scheme for NFA$^{\oplus_p}$; the computational steps are completely analogous to those in GWW.

In more detail, we want to replace $\mathbf{d}$ with $\mathbf{d}+\mathbf{d}'_i$ in G_i, where $\mathbf{d}'_i \in \mathbb{Z}_p^Q$ corresponds to the extra entropy we introduce into the secret keys in the semi-functional space. Note that $\mathbf{d}'_i$ will depend on both the challenge attribute x^* as well as the underlying NFA$^{\oplus_p}$. We have the following constraints on $\mathbf{d}'_i$'s, arising from the fact that an adversarial distinguisher for $\mathsf{G}_0, \ldots, \mathsf{G}_\ell$ can always compute what a decryptor can compute in (5):

- to mask α in G_ℓ, we set $\mathbf{d}'_\ell = \Delta\mathbf{f}$ where $\Delta \leftarrow \mathbb{Z}_p$, so that
$$\alpha\mathbf{f} - (\mathbf{d}+\mathbf{d}'_\ell) = (\alpha-\Delta)\mathbf{f} - \mathbf{d}$$
perfectly hides α;
- (ii) implies that
$$\implies \begin{array}{c} \overbrace{s_i\mathbf{d}\mathbf{M}_{x_i^*} - s_{i-1}(\mathbf{d}+\mathbf{d}'_{i-1})}^{\mathsf{G}_{i-1}} \approx_s \overbrace{s_i(\mathbf{d}+\mathbf{d}'_i)\mathbf{M}_{x_i^*} - s_{i-1}\mathbf{d}}^{\mathsf{G}_i} \\ -s_{i-1}\mathbf{d}'_{i-1} \approx_s s_i\mathbf{d}'_i\mathbf{M}_{x_i^*} \end{array}$$
to prevent a distinguishing attack[4] between G_{i-1} and G_i by computing $s_i\mathbf{d}\mathbf{M}_{x_i^*} - s_{i-1}\mathbf{d}$ in both games;
- (iii) implies that $s_0(\mathbf{d}+\mathbf{d}'_0)\mathbf{u}^\top = s_0\mathbf{d}\mathbf{u}^\top$, and therefore, $\mathbf{d}'_0\mathbf{u}^\top = 0 \bmod p$. This is to prevent a distinguishing attack[5] between the real keys and those in G_0.

In particular, we can satisfy the first two constraints by setting[6]

$$\mathbf{d}'_i = \Delta \cdot \mathbf{f}\mathbf{M}_{x_\ell^*} \cdots \mathbf{M}_{x_{i+1}^*} \ \forall i \in [0,\ell]$$

where $\approx_s$ holds over $\Delta \leftarrow \mathbb{Z}_p$, as long as $s_0, \ldots, s_\ell \neq 0$. Whenever $\Gamma(x^*) = 0$, we have

$$\mathbf{f}\mathbf{M}_{x_\ell^*} \cdots \mathbf{M}_{x_1^*}\mathbf{u}^\top = 0 \bmod p$$

and therefore the third constraint is also satisfied.

Two clarifying remarks. First, the quantity

$$\mathbf{f}\mathbf{M}_{x_\ell^*} \cdots \mathbf{M}_{x_{i+1}^*}$$

used in defining $\mathbf{d}'_i$ has a natural combinatorial interpretation: its u'th coordinate corresponds to the number of paths from the accept states to u, while back-tracking along $x_\ell^*, \ldots, x_{i+1}^*$. In the specific case of a DFA, this value is 1 if u is reachable from an accept state, and 0 otherwise. It is then easy to see that our proof strategy generalizes that of GWW for DFA: the latter adds Δ to d_u in G_i whenever u is reachable from accept state while back-tracking along the last $\ell - i$ bits of the challenge attribute (cf. [11, Sec.

[4] Looking ahead to the proof of security in Section 4, this "simplified" attack corresponds roughly to using $\mathsf{ct}_{x^*}^{i-1,i}$ to distinguish $\mathsf{sk}_\Gamma^{i-1,i}$ and sk_Γ^i; this comes up in the proof of $\mathsf{G}_{2.i.2} \approx_c \mathsf{G}_{2.i.3}$ in Lemma 8.

[5] In Section 4, this roughly corresponds to distinguish sk_Γ and sk_Γ^0 with $\mathsf{ct}_{x^*}^0$; this comes up in the proof of $\mathsf{G}_1 \approx_c \mathsf{G}_{2.1.0}$ in Lemma 6.

[6] We adopt the standard convention that the product of an empty sequence of matrices is the identity matrix. This means $\mathbf{d}'_\ell = \Delta \cdot \mathbf{f}$.

3.2]). Second, the "naive" (and insecure) ABE for NFA that captures non-deterministic transitions as in (2) introduces more equations in (ii) in (5); this in turn yields more –and ultimately unsatisfiable– constraints on the $\mathbf{d}'_i$'s.

Finally, we remark that our ABE for NFA$^{\oplus_P}$ (and ABE for DFA from GWW as well) can be proved in the semi-adaptive model [9], which is weaker than adaptive security but stronger than both selective and selective* model used in [3].

Adaptive security for restricted NFA$^{\oplus_P}$ and DFA. Fix a set $\mathcal{F} \subseteq \mathbb{Z}^Q$. We say that an NFA or an NFA$^{\oplus_P}$is $\mathcal{F}$-restricted if

$$\forall\, \ell \in \mathbb{N},\, x \in \Sigma^\ell,\, i \in [0, \ell] \,:\, \mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_{i+1}} \in \mathcal{F}$$

Note that $\mathbf{f}\mathbf{M}_{x^*_\ell} \cdots \mathbf{M}_{x^*_{i+1}}$ corresponding to the challenge attribute x^* is exactly what is used to define $\mathbf{d}'_i$ in the previous paragraph. Moreover, following GWW, knowing this quantity is sufficient to prove indistinguishability of G_{i-1} and G_i. This means that to prove selective security for $\mathcal{F}$-restricted NFAs, it suffices to know $\log |\mathcal{F}|$ bits about the challenge attribute, and via the piecewise guessing framework, this yields adaptive security with a security loss of $|\mathcal{F}|$. Unfortunately, $|\mathcal{F}|$ is in general exponentially large for general NFAs and DFAs. In particular, DFAs are $\{0,1\}^Q$-restricted, and naively applying this argument would yield adaptively secure DFAs with a 2^Q security loss.

Instead, we show how to transform DFAs into $\mathcal{E}_Q$-restricted NFA$^{\oplus_P}$, where $\mathcal{E}_Q \subset \{0,1\}^Q$ is the collection of Q elementary row vectors of dimension Q; this yields adaptively secure ABE for DFAs with a security loss of $|\mathcal{E}_Q| = Q$. Concretely, our adaptively secure ABE for DFA uses an adaptively secure ABE for $\mathcal{E}_Q$-restricted NFA$^{\oplus_P}$, and proceeds

- to encrypt $x = (x_1, \ldots, x_\ell)$, use the ABE for NFA to encrypt $x^\top = (x_\ell, \ldots, x_1)$;[7]
- to generate a secret key for a DFA $\Gamma = (Q, \Sigma, \{\mathbf{M}_\sigma\}, \mathbf{u}, \mathbf{f})$, use the ABE for NFA to generate a key for $\Gamma^\top = (Q, \Sigma, \{\mathbf{M}^\top_\sigma\}, \mathbf{f}, \mathbf{u})$.

Note that we reversed x during encryption, and transposed $\mathbf{M}_\sigma$, and switched $\mathbf{u}, \mathbf{f}$ during key generation. Correctness essentially follows from the equality

$$\overbrace{\mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top}^{\Gamma(x)} = (\mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top)^\top = \overbrace{\mathbf{u}\mathbf{M}^\top_{x_1} \cdots \mathbf{M}^\top_{x_\ell}\mathbf{f}^\top}^{\Gamma^\top(x^\top)}.$$

Furthermore $\Gamma^\top = (Q, \Sigma, \{\mathbf{M}^\top_\sigma\}, \mathbf{f}, \mathbf{u})$ is indeed a $\mathcal{E}_Q$-restricted NFA$^{\oplus_P}$. This follows from the fact that for any DFA Γ:

$$\forall\, \ell \in \mathbb{N},\, x \in \Sigma^\ell,\, i \in [0, \ell] \,:\, (\mathbf{M}_{x_i} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top)^\top \in \mathcal{E}_Q$$

which is implied by the property of DFA: $\mathbf{u} \in \mathcal{E}_Q$ and each column in every matrix $\mathbf{M}_\sigma$ contains exactly one 1. We give an example of reversing DFA in the full paper.

[7] We acknowledge that writing $x^\top$ constitutes an abuse of notation, but nonetheless convenient in analogy with $\mathbf{M}^\top_\sigma$.

	policy	security	decryption		proof	
			direction	information	direction	information
GWW [11]	DFA	selective	forward	reachability	backward	reachability
§ 5	DFA	adaptive	backward	reachability	forward	reachability
Naive	NFA	broken	forward	reachability	-	-
§ 4	NFA	selective	forward	# paths	backward	# paths

Fig. 5. Summary of tracing executions underlying GWW, our adaptively secure ABE for DFA, our selectively secure ABE for $\mathrm{NFA}^{\oplus_P}$ and naive extension of Waters' ABE for DFA.

1.3 Discussion

Tracing executions. Recall that a DFA is specified using a transition function $\delta : [Q] \times \Sigma \to [Q]$. A forward computation upon reading σ goes from a state u to $v = \delta(u, \sigma)$, whereas back-tracking upon reading σ goes from v to u if $v = \delta(u, \sigma)$.

- GWW selective ABE for DFA: Decryption follows normal "forward" computation keeping track of whether a state is reachable from the start state, whereas the security proof introduces entropy based on whether a state is reachable from the accept states via "back-tracking".
- Our adaptive ABE for DFA and branching programs: Decryption uses back-tracking and keeps track of whether a state is reachable from the accept states, whereas the security proof introduces entropy based on whether a state is reachable from the start state via forward computation. To achieve polynomial security loss, we crucially rely on the fact that when reading i input bits, exactly one state is reachable from the start state via forward computation.
- Naive and insecure ABE for $\mathrm{NFA}^{\oplus_P}$: Decryption follows normal forward computation keeping track of whether a state is reachable from the start state.
- Our selective ABE for $\mathrm{NFA}^{\oplus_P}$: Decryption follows normal forward computation keeping track of the number of paths from the start state, whereas the security proof introduces entropy scaled by the number of paths that are reachable from the accept states via back-tracking.

We summarize the discussion in Fig 5.

ABE for DFA vs branching programs. Our work clarifies that the same obstacle (having to guess a large subset of states that are reached upon back-tracking) arose in constructing adaptive ABE for DFA and compact adaptive ABE for branching programs from k-Lin, and presents a new technique that solves both problems simultaneously in the setting of KP-ABE. Furthermore, our results and techniques can carry over to the CP-ABE settings using more-or-less standard (but admittedly non-black-box) arguments, following e.g. [4, Sec.8] and [6, Sec.4]. See the full paper for adaptively secure CP-ABE for DFA and branching programs, respectively.

Interestingly, the very recent work of Agarwal *et al.* [3,2] shows a related connection: namely that compact and unbounded adaptive KP and CP-ABE for branching

programs[8] –for which they do not provide any instantiations– yields compact adaptive KP-ABE (as well as CP-ABE) for DFA. In particular, just getting to KP-ABE for DFA already requires both KP and CP-ABE for branching programs and also incurs a larger polynomial blow-up in the parameters compared to our constructions; furthermore, simply getting to compact, unbounded, adaptive KP-ABE for branching programs would also require most of the technical machinery used in this work, notably the "nested, two-slot" dual system argument and the piecewise guessing framework. Nonetheless, there is significant conceptual appeal to having a generic and modular transformation that also yields both KP-ABE and CP-ABE schemes. That said, at the core of our constructions and analysis is a very simple combinatorial object sketched in Section 1.2. We leave the question of properly formalizing this object and building a generic compiler to full-fledged KP-ABE and CP-ABE schemes to further work; in particular, such a compiler should (i) match or improve upon the concrete efficiency of our schemes, as with prior compilers such as [7,5], and (ii) properly decouple the combinatorial arguments that are specific to DFA, NFA and branching programs from the computational arguments that are oblivious to the underlying computational model.

Organization. The next section gives some background knowledge. Section 3 shows the transformation from DFA to $\mathcal{E}$-restricted NFA$^{\oplus_p}$. We show our selectively secure ABE for NFA$^{\oplus_p}$ in Section 4 and upgrade to adaptive security for $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$ in Section 5. The latter implies our adaptively secure ABE for DFA. See the full paper for the concrete description and our basic selectively secure ABE for NFA$^{\oplus_p}$ from q-type assumption. We also defer our compact adaptively secure ABE for branching programs to the full paper.

2 Preliminaries

Notation. We denote by $s \leftarrow S$ the fact that s is picked uniformly at random from a finite set S; by $U(S)$, we indicate uniform distribution over finite set S. We use $\approx_s$ to denote two distributions being statistically indistinguishable, and $\approx_c$ to denote two distributions being computationally indistinguishable. We use $\langle \mathcal{A}, \mathsf{G} \rangle = 1$ to denote that an adversary $\mathcal{A}$ wins in an interactive game G. We use lower case boldface to denote *row* vectors and upper case boldcase to denote matrices. We use $\mathbf{e}_i$ to denote the i'th elementary (row) vector (with 1 at the i'th position and 0 elsewhere) and let $\mathcal{E}_Q$ denote the set of all elementary vectors of dimension Q. For matrix $\mathbf{A}$, we use $\mathsf{span}(\mathbf{A})$ to denote the *row* span of $\mathbf{A}$ and use $\mathsf{basis}(\mathbf{A})$ to denote a basis of *column* span of $\mathbf{A}$. Throughout the paper, we use prime number p to denote the order of underlying groups.

2.1 Attribute-based encryption

Syntax. An attribute-based encryption (ABE) scheme for some class $\mathcal{C}$ consists of four algorithms:

[8] The statement in [3] refers to monotone span programs, which is a more powerful object, but we believe that branching program suffices.

$\mathsf{Setup}(1^\lambda, \mathcal{C}) \to (\mathsf{mpk}, \mathsf{msk})$. The setup algorithm gets as input the security parameter 1^λ and class description $\mathcal{C}$. It outputs the master public key mpk and the master secret key msk. We assume mpk defines the message space $\mathcal{M}$.

$\mathsf{Enc}(\mathsf{mpk}, x, m) \to \mathsf{ct}_x$. The encryption algorithm gets as input mpk, an input x and a message $m \in \mathcal{M}$. It outputs a ciphertext ct_x. Note that x is public given ct_x.

$\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \Gamma) \to \mathsf{sk}_\Gamma$. The key generation algorithm gets as input mpk, msk and $\Gamma \in \mathcal{C}$. It outputs a secret key sk_Γ. Note that Γ is public given sk_Γ.

$\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_\Gamma, \mathsf{ct}_x) \to m$. The decryption algorithm gets as input sk_Γ and ct_x such that $\Gamma(x) = 1$ along with mpk. It outputs a message m.

Correctness. For all input x and Γ with $\Gamma(x) = 1$ and all $m \in \mathcal{M}$, we require

$$\Pr\left[\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_\Gamma, \mathsf{ct}_x) = m : \begin{array}{c}(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{C}) \\ \mathsf{sk}_\Gamma \leftarrow \mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \Gamma) \\ \mathsf{ct}_x \leftarrow \mathsf{Enc}(\mathsf{mpk}, x, m)\end{array}\right] = 1.$$

Security definition. For a stateful adversary $\mathcal{A}$, we define the advantage function

$$\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A}}(\lambda) := \Pr\left[\beta = \beta' : \begin{array}{l}(\mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}(1^\lambda, \mathcal{C}) \\ (x^*, m_0, m_1) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \cdot)}(\mathsf{mpk}) \\ \beta \leftarrow \{0,1\};\ \mathsf{ct}_{x^*} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x^*, m_\beta) \\ \beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \cdot)}(\mathsf{ct}_{x^*})\end{array}\right] - \frac{1}{2}$$

with the restriction that all queries Γ that $\mathcal{A}$ sent to $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \cdot)$ satisfy $\Gamma(x^*) = 0$. An ABE scheme is *adaptively secure* if for all PPT adversaries $\mathcal{A}$, the advantage $\mathsf{Adv}^{\mathrm{ABE}}_{\mathcal{A}}(\lambda)$ is a negligible function in λ. The *selective* security is defined analogously except that the adversary $\mathcal{A}$ selects x^* before seeing mpk. A notion between selective and adaptive is so-called *semi-adaptive security* [9] where the adversary $\mathcal{A}$ is allowed to select x^* after seeing mpk but before making any queries.

2.2 Prime-order Groups

A generator $\mathcal{G}$ takes as input a security parameter 1^λ and outputs a description $\mathbb{G} := (p, G_1, G_2, G_T, e)$, where p is a prime of $\Theta(\lambda)$ bits, G_1, G_2 and G_T are cyclic groups of order p, and $e : G_1 \times G_2 \to G_T$ is a non-degenerate bilinear map. We require that the group operations in G_1, G_2, G_T and the bilinear map e are computable in deterministic polynomial time in λ. Let $g_1 \in G_1$, $g_2 \in G_2$ and $g_T = e(g_1, g_2) \in G_T$ be the respective generators. We employ the *implicit representation* of group elements: for a matrix $\mathbf{M}$ over $\mathbb{Z}_p$, we define $[\mathbf{M}]_1 := g_1^{\mathbf{M}}, [\mathbf{M}]_2 := g_2^{\mathbf{M}}, [\mathbf{M}]_T := g_T^{\mathbf{M}}$, where exponentiation is carried out component-wise. Also, given $[\mathbf{A}]_1, [\mathbf{B}]_2$, we let $e([\mathbf{A}]_1, [\mathbf{B}]_2) = [\mathbf{AB}]_T$. We recall the matrix Diffie-Hellman (MDDH) assumption on G_1 [10]:

Assumption 1 ($\mathrm{MDDH}^d_{k,k'}$ Assumption) *Let $k' > k \geq 1$ and $d \geq 1$. We say that the $\mathrm{MDDH}^d_{k,k'}$ assumption holds if for all PPT adversaries $\mathcal{A}$, the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\mathrm{MDDH}^d_{k,k'}}_{\mathcal{A}}(\lambda) := \left| \Pr[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_1, \boxed{[\mathbf{MS}]_1}) = 1] - \Pr[\mathcal{A}(\mathbb{G}, [\mathbf{M}]_1, \boxed{[\mathbf{U}]_1}) = 1] \right|$$

where $\mathbb{G} := (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{M} \leftarrow \mathbb{Z}_p^{k' \times k}$, $\mathbf{S} \leftarrow \mathbb{Z}_p^{k \times d}$ and $\mathbf{U} \leftarrow \mathbb{Z}_p^{k' \times d}$.

The MDDH assumption on G_2 can be defined in an analogous way. Escala *et al.* [10] showed that

$$k\text{-Lin} \Rightarrow \mathrm{MDDH}^1_{k,k+1} \Rightarrow \mathrm{MDDH}^d_{k,k'} \ \forall k' > k, d \geq 1$$

with a tight security reduction. We will use $\mathsf{Adv}^{k\text{-LIN}}_{\mathcal{A}}(\lambda)$ to denote the advantage function w.r.t. k-Lin assumption.

3 DFA, NFA, and their Relationships

Let p be a global parameter and $\mathcal{E}_Q = \{\mathbf{e}_1, \ldots, \mathbf{e}_Q\}$ be the set of all elementary row vectors of dimension Q. This section describes various notions of DFA and NFA and studies their relationships.

Finite Automata. We use $\Gamma = (Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma \in \Sigma}, \mathbf{u}, \mathbf{f})$ to describe deterministic finite automata (DFA for short), nondeterministic finite automata (NFA for short), p-bounded NFA ($\mathrm{NFA}^{<p}$ for short) and mod-p NFA ($\mathrm{NFA}^{\oplus_p}$ for short), where $Q \in \mathbb{N}$ is the number of states, vectors $\mathbf{u}, \mathbf{f} \in \{0,1\}^{1 \times Q}$ describe the start and accept states, a collection of matrices $\mathbf{M}_\sigma \in \{0,1\}^{Q \times Q}$ describe the transition function. Let $x = (x_1, \ldots, x_\ell)$ denote an input, then,

- for DFA Γ, we have $\mathbf{u} \in \mathcal{E}_Q$, each column in every matrix $\mathbf{M}_\sigma$ is an elementary column vector (i.e., contains exactly one 1) and

$$\Gamma(x) = 1 \iff \mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top = 1;$$

- for NFA Γ, we have

$$\Gamma(x) = 1 \iff \mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top > 0;$$

- for $\mathrm{NFA}^{<p}$ Γ, we have $\mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top < p$ and

$$\Gamma(x) = 1 \iff \mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top > 0;$$

- for $\mathrm{NFA}^{\oplus_p}$ Γ, we have

$$\Gamma(x) = 1 \iff \mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top \neq 0 \bmod p.$$

We immediately have: DFA $\subset$ $\mathrm{NFA}^{<p}$ $\subset$ NFA $\cap$ $\mathrm{NFA}^{\oplus_p}$.

$\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$. We introduce the notion of $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$ which is an NFA$^{\oplus_p}$ $\Gamma = (\, Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma\in\Sigma}, \mathbf{u}, \mathbf{f}\,)$ with an additional property: for all $\ell \in \mathbb{N}$ and all $x \in \Sigma^\ell$, it holds that

$$\mathbf{f}_{i,x} := \mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_{i+1}} \in \mathcal{E}_Q,\ \forall i \in [0, \ell]$$

Here $\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_{i+1}}$ for $i = \ell$ refers to $\mathbf{I}$ of size $Q \times Q$.

Transforming DFA to $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$. In general, a DFA is not necessarily a $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$. The next lemma says that we can nonetheless transform any DFA into a $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$:

Lemma 1 (DFA to $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$). *For each DFA $\Gamma = (\, Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma\in\Sigma}, \mathbf{u}, \mathbf{f}\,)$, we have NFA$^{\oplus_p}$ $\Gamma^\top = (Q, \Sigma, \{\mathbf{M}_\sigma^\top\}_{\sigma\in\Sigma}, \mathbf{f}, \mathbf{u})$ such that*

1. *$\Gamma^\top$ is $\mathcal{E}_Q$-restricted;*
2. *for all $\ell \in \mathbb{N}$ and $x = (x_1, \ldots, x_\ell) \in \Sigma^\ell$, it holds that*

$$\Gamma(x) = 1 \iff \Gamma^\top(x^\top) = 1 \quad \textit{where } x^\top = (x_\ell, \ldots, x_1) \in \Sigma^\ell. \tag{7}$$

Proof. Recall that the definition of DFA implies two properties:

$$\mathbf{f} \in \{0,1\}^Q \tag{8}$$

$$\text{and} \quad (\mathbf{M}_{x_i} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top)^\top \in \mathcal{E}_Q, \quad \forall i \in [0, \ell]. \tag{9}$$

Property (9) comes from the facts that $\mathbf{u} \in \mathcal{E}_Q$ and each column in every matrix $\mathbf{M}_\sigma$ is an elementary column vector.

We parse $x^\top = (x_1^\top, \ldots, x_\ell^\top)$ and prove the two parts of the lemma as below.

1. $\Gamma^\top$ is $\mathcal{E}_Q$-restricted since we have

$$\mathbf{u}\mathbf{M}_{x_\ell^\top}^\top \cdots \mathbf{M}_{x_{i+1}^\top}^\top = (\mathbf{M}_{x_{\ell-i}} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top)^\top \in \mathcal{E}_Q, \quad \forall i \in [0, \ell]$$

   where the equality is implied by the structure of $\Gamma^\top, x^\top$ and we use property (9).
2. To prove (7), we rely on the fact

$$\begin{aligned}
\Gamma(x) = 1 &\iff \mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top = 1 \\
&\iff \mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top \neq 0 \bmod p \\
&\iff \mathbf{u}\mathbf{M}_{x_\ell^\top}^\top \cdots \mathbf{M}_{x_1^\top}^\top \mathbf{f}^\top \neq 0 \bmod p \\
&\iff \Gamma^\top(x^\top) = 1.
\end{aligned}$$

   The second $\iff$ follows from the fact that $\mathbf{f}\mathbf{M}_{x_\ell} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top \in \{0,1\}$ which is implied by property (8) and (9) while the third $\iff$ is implied by the structure of $\Gamma^\top, x^\top$. □

4 Semi-adaptively Secure ABE for NFA$^{\oplus_p}$

In this section, we present our ABE for NFA$^{\oplus_p}$ in prime-order groups. The scheme achieves semi-adaptive security under the k-Lin assumption. Our construction is based on GWW ABE for DFA [11] along with an extension of the key structure and decryption to NFA; the security proof follows that of GWW with our novel combinatorial arguments regarding our NFA extension. (See Section 1.2 for an overview.) We remark that our scheme and proof work well for a more general form of NFA$^{\oplus_p}$ where $\mathbf{u}, \mathbf{f}, \mathbf{M}_\sigma$ are over $\mathbb{Z}_p$ instead of $\{0,1\}$.

4.1 Basis

We will use the same basis as GWW [11]:

$$\mathbf{A}_1 \leftarrow \mathbb{Z}_p^{k\times(2k+1)}, \quad \mathbf{a}_2 \leftarrow \mathbb{Z}_p^{1\times(2k+1)}, \quad \mathbf{A}_3 \leftarrow \mathbb{Z}_p^{k\times(2k+1)} \tag{10}$$

and use $(\mathbf{A}_1^\parallel \mid \mathbf{a}_2^\parallel \mid \mathbf{A}_3^\parallel)$ to denote the dual basis so that $\mathbf{A}_i\mathbf{A}_i^\parallel = \mathbf{I}$ (known as *non-degeneracy*) and $\mathbf{A}_i\mathbf{A}_j^\parallel = \mathbf{0}$ if $i \neq j$ (known as *orthogonality*). For notational convenience, we always consider $\mathbf{a}_2^\parallel$ as a column vector. We review $\mathrm{SD}^{G_1}_{\mathbf{A}_1\mapsto\mathbf{A}_1,\mathbf{A}_3}$ and $\mathrm{DDH}^{G_2}_{d,Q}$ assumption from [8] which are parameterized for basis (10) and tightly implied by k-Lin assumption. By symmetry, we may permute the indices for $\mathbf{A}_1, \mathbf{a}_2, \mathbf{A}_3$.

Lemma 2 ($\mathbf{MDDH}_{k,2k} \Rightarrow \mathbf{SD}^{G_1}_{\mathbf{A}_1\mapsto\mathbf{A}_1,\mathbf{A}_3}$ [8]). *Under the $\mathrm{MDDH}_{k,2k}$ assumption in G_1, there exists an efficient sampler outputting random $([\mathbf{A}_1]_1, [\mathbf{a}_2]_1, [\mathbf{A}_3]_1)$ along with base $\mathsf{basis}(\mathbf{A}_1^\parallel)$, $\mathsf{basis}(\mathbf{a}_2^\parallel)$, $\mathsf{basis}(\mathbf{A}_1^\parallel, \mathbf{A}_3^\parallel)$ (of arbitrary choice) such that the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\mathrm{SD}^{G_1}_{\mathbf{A}_1\mapsto\mathbf{A}_1,\mathbf{A}_3}}_{\mathcal{A}}(\lambda) := \left|\Pr[\mathcal{A}(D, [\mathbf{t}_0]_1) = 1] - \Pr[\mathcal{A}(D, [\mathbf{t}_1]_1) = 1]\right|$$

where

$$D := (\ [\mathbf{A}_1]_1, [\mathbf{a}_2]_1, [\mathbf{A}_3]_1, \mathsf{basis}(\mathbf{A}_1^\parallel), \mathsf{basis}(\mathbf{a}_2^\parallel), \mathsf{basis}(\mathbf{A}_1^\parallel, \mathbf{A}_3^\parallel)\ ),$$
$$\mathbf{t}_0 \leftarrow \boxed{\mathsf{span}(\mathbf{A}_1)},\ \mathbf{t}_1 \leftarrow \mathsf{span}(\mathbf{A}_1, \mathbf{A}_3).$$

More concretely, we have, for all $\mathcal{A}$, there exists $\mathcal{B}$ with $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$ such that $\mathsf{Adv}^{\mathrm{SD}^{G_1}_{\mathbf{A}_1\mapsto\mathbf{A}_1,\mathbf{A}_3}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathrm{MDDH}_{k,2k}}_{\mathcal{A}}(\lambda)$.

Lemma 3 ($\mathbf{MDDH}^d_{k,k+d} \Rightarrow \mathbf{DDH}^{G_2}_{d,Q}$ [8]). *Let $d, Q \in \mathbb{N}$. Under the $\mathrm{MDDH}^d_{k,k+d}$ assumption in G_2, the following advantage function is negligible in λ.*

$$\mathsf{Adv}^{\mathrm{DDH}^{G_2}_{d,Q}}_{\mathcal{A}}(\lambda) := \Big|\Pr[\mathcal{A}([\mathbf{WB}]_2, [\mathbf{B}]_2, \boxed{[\mathbf{WR}]_2}, [\mathbf{R}]_2) = 1] - \Pr[\mathcal{A}([\mathbf{WB}]_2, [\mathbf{B}]_2, \boxed{[\mathbf{WR}+\mathbf{U}]_2}, [\mathbf{R}]_2) = 1]\Big|$$

where $\mathbf{W} \leftarrow \mathbb{Z}_p^{d\times k}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{k\times k}$, $\mathbf{R} \leftarrow \mathbb{Z}_p^{k\times Q}$ and $\mathbf{U} \leftarrow \mathbb{Z}_p^{d\times Q}$. More concretely, we have, for all $\mathcal{A}$, there exists $\mathcal{B}$ with $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$ such that $\mathsf{Adv}^{\mathrm{DDH}^{G_2}_{d,Q}}_{\mathcal{A}}(\lambda) \leq O(1) \cdot \mathsf{Adv}^{\mathrm{MDDH}^d_{k,k+d}}_{\mathcal{A}}(\lambda)$.

Lemma 4 (statistical lemma [8]). *With probability* $1-1/p$ *over* $\mathbf{A}_1, \mathbf{a}_2, \mathbf{A}_3, \mathbf{A}_1^{\shortparallel}, \mathbf{a}_2^{\shortparallel}, \mathbf{A}_3^{\shortparallel}$, *the following two distributions are statistically identical.*

$$\left\{\ \mathbf{A}_1\mathbf{W}, \mathbf{A}_3\mathbf{W}, \boxed{\mathbf{a}_2\mathbf{W}}\ \right\} \quad \textit{and} \quad \left\{\ \mathbf{A}_1\mathbf{W}, \mathbf{A}_3\mathbf{W}, \boxed{\mathbf{w}}\ \right\}$$

where $\mathbf{W} \leftarrow \mathbb{Z}_p^{(2k+1)\times k}$ *and* $\mathbf{w} \leftarrow \mathbb{Z}_p^{1\times k}$.

4.2 Scheme

Our ABE for NFA$^{\oplus_p}$ in prime-order groups is described as follows:

– $\mathsf{Setup}(1^\lambda, \Sigma)$: Run $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample

$$\mathbf{A}_1 \leftarrow \mathbb{Z}_p^{k\times(2k+1)}, \mathbf{k} \leftarrow \mathbb{Z}_p^{1\times(2k+1)}, \mathbf{W}_{\text{start}}, \mathbf{Z}_b, \mathbf{W}_{\sigma,b}, \mathbf{W}_{\text{end}} \leftarrow \mathbb{Z}_p^{(2k+1)\times k}$$

for all $\sigma \in \Sigma$ and $b \in \{0,1\}$. Output

$$\mathsf{mpk} = \left([\,\mathbf{A}_1,\ \mathbf{A}_1\mathbf{W}_{\text{start}},\ \{\,\mathbf{A}_1\mathbf{Z}_b,\ \mathbf{A}_1\mathbf{W}_{\sigma,b}\,\}_{\sigma\in\Sigma,b\in\{0,1\}},\ \mathbf{A}_1\mathbf{W}_{\text{end}}\,]_1,\ [\mathbf{A}_1\mathbf{k}^\top]_T \right)$$
$$\mathsf{msk} = \left(\mathbf{k},\ \mathbf{W}_{\text{start}},\ \{\,\mathbf{Z}_b,\ \mathbf{W}_{\sigma,b}\,\}_{\sigma\in\Sigma,b\in\{0,1\}},\ \mathbf{W}_{\text{end}} \right).$$

– $\mathsf{Enc}(\mathsf{mpk}, x, m)$: Let $x = (x_1, \ldots, x_\ell) \in \Sigma^\ell$ and $m \in G_T$. Pick $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_\ell \leftarrow \mathbb{Z}_p^{1\times k}$ and output

$$\mathsf{ct}_x = \begin{pmatrix} [\mathbf{s}_0\mathbf{A}_1]_1,\ [\mathbf{s}_0\mathbf{A}_1\mathbf{W}_{\text{start}}]_1 \\ \left\{[\mathbf{s}_j\mathbf{A}_1]_1, [\mathbf{s}_{j-1}\mathbf{A}_1\mathbf{Z}_{j \bmod 2} + \mathbf{s}_j\mathbf{A}_1\mathbf{W}_{x_j, j \bmod 2}]_1\right\}_{j\in[\ell]} \\ [\mathbf{s}_\ell\mathbf{A}_1]_1, [\mathbf{s}_\ell\mathbf{A}_1\mathbf{W}_{\text{end}}]_1,\ [\mathbf{s}_\ell\mathbf{A}_1\mathbf{k}^\top]_T \cdot m \end{pmatrix}.$$

– $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \Gamma)$: Let $\Gamma = (\,Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma\in\Sigma}, \mathbf{u}, \mathbf{f}\,)$. Pick $\mathbf{D} \leftarrow \mathbb{Z}_p^{(2k+1)\times Q}$, $\mathbf{R} \leftarrow \mathbb{Z}_p^{k\times Q}$ and output

$$\mathsf{sk}_\Gamma = \begin{pmatrix} [\mathbf{D}\mathbf{u}^\top + \mathbf{W}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{D} + \mathbf{Z}_b\mathbf{R}]_2, [\mathbf{D}\mathbf{M}_\sigma + \mathbf{W}_{\sigma,b}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma,b\in\{0,1\}} \\ [\mathbf{k}^\top\mathbf{f} - \mathbf{D} + \mathbf{W}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}.$$

– $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_\Gamma, \mathsf{ct}_x)$: Parse ciphertext for $x = (x_1, \ldots, x_\ell)$ and key for $\Gamma = (\,Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma\in\Sigma}, \mathbf{u}, \mathbf{f}\,)$ as:

$$\mathsf{ct}_x = \begin{pmatrix} [\mathbf{c}_{0,1}]_1, [\mathbf{c}_{0,2}]_1 \\ \left\{\,[\mathbf{c}_{j,1}]_1, [\mathbf{c}_{j,2}]_1\,\right\}_j \\ [\mathbf{c}_{\ell,1}]_1, [\mathbf{c}_{\text{end}}]_1, C \end{pmatrix} \quad \text{and} \quad \mathsf{sk}_\Gamma = \begin{pmatrix} [\mathbf{k}_0^\top]_2, [\mathbf{r}_0^\top]_2 \\ \left\{\,[\mathbf{K}_b]_2, [\mathbf{K}_{\sigma,b}]_2, [\mathbf{R}]_2\,\right\}_{\sigma,b} \\ [\mathbf{K}_{\text{end}}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

We define

$$\mathbf{u}_{j,x}^\top = \mathbf{M}_{x_j} \cdots \mathbf{M}_{x_1}\mathbf{u}^\top \bmod p,\ \forall j \in [0,\ell] \tag{11}$$

and proceed as follows:

1. Compute

$$B_0 = e([\mathbf{c}_{0,1}]_1, [\mathbf{k}_0^\top]_2) \cdot e([\mathbf{c}_{0,2}]_1, [\mathbf{r}_0^\top]_2)^{-1};$$

2. For all $j \in [\ell]$, compute

$$[\mathbf{b}_j]_T = e([\mathbf{c}_{j-1,1}]_1, [\mathbf{K}_{j \bmod 2}]_2) \cdot e([\mathbf{c}_{j,1}]_1, [\mathbf{K}_{x_j, j \bmod 2}]_2) \cdot e([-\mathbf{c}_{j,2}]_1, [\mathbf{R}]_2)$$
$$\text{and} \quad B_j = [\mathbf{b}_j \mathbf{u}_{j-1,x}^\top]_T;$$

3. Compute

$$[\mathbf{b}_{\text{end}}]_T = e([\mathbf{c}_{\ell,1}]_1, [\mathbf{K}_{\text{end}}]_2) \cdot e([-\mathbf{c}_{\text{end}}]_1, [\mathbf{R}]_2) \quad \text{and} \quad B_{\text{end}} = [\mathbf{b}_{\text{end}} \mathbf{u}_{\ell,x}^\top]_T;$$

4. Compute

$$B_{\text{all}} = B_0 \cdot \textstyle\prod_{j=1}^{\ell} B_j \cdot B_{\text{end}} \quad \text{and} \quad B = B_{\text{all}}^{(\mathbf{f}\mathbf{u}_{\ell,x}^\top)^{-1}}$$

and output the message $m' \leftarrow C \cdot B^{-1}$.

Correctness. For $x = (x_1, \ldots, x_\ell)$ and $\Gamma = (Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma \in \Sigma}, \mathbf{u}, \mathbf{f})$ such that $\Gamma(x) = 1$, we have:

$$B_0 = [\mathbf{s}_0 \mathbf{A}_1 \mathbf{D} \mathbf{u}^\top]_T = [\mathbf{s}_0 \mathbf{A}_1 \mathbf{D} \mathbf{u}_{0,x}^\top]_T \tag{12}$$
$$\mathbf{b}_j = \mathbf{s}_j \mathbf{A}_1 \mathbf{D} \mathbf{M}_{x_j} - \mathbf{s}_{j-1} \mathbf{A}_1 \mathbf{D} \tag{13}$$
$$B_j = [\mathbf{s}_j \mathbf{A}_1 \mathbf{D} \mathbf{u}_{j,x}^\top - \mathbf{s}_{j-1} \mathbf{A}_1 \mathbf{D} \mathbf{u}_{j-1,x}^\top]_T \tag{14}$$
$$\mathbf{b}_{\text{end}} = \mathbf{s}_\ell \mathbf{A}_1 \mathbf{k}^\top \mathbf{f} - \mathbf{s}_\ell \mathbf{A}_1 \mathbf{D} \tag{15}$$
$$B_{\text{end}} = [\mathbf{s}_\ell \mathbf{A}_1 \mathbf{k}^\top \mathbf{f} \mathbf{u}_{\ell,x}^\top - \mathbf{s}_\ell \mathbf{A}_1 \mathbf{D} \mathbf{u}_{\ell,x}^\top]_T \tag{16}$$
$$B_{\text{all}} = [\mathbf{s}_\ell \mathbf{A}_1 \mathbf{k}^\top \mathbf{f} \mathbf{u}_{\ell,x}^\top]_T \tag{17}$$
$$B = [\mathbf{s}_\ell \mathbf{A}_1 \mathbf{k}^\top]_T \tag{18}$$

Here (16) is trivial; (14) and (18) follow from

$$\mathbf{u}_{j,x}^\top = \mathbf{M}_{x_j} \mathbf{u}_{j-1,x}^\top \bmod p, \ \forall j \in [\ell] \quad \text{and} \quad \Gamma(x) = 1 \iff \mathbf{f}\mathbf{u}_{\ell,x}^\top \neq 0 \bmod p \tag{19}$$

by the definition in (11), the remaining equalities follow [7], more detail can be found in the full paper.

Security. We have the following theorem stating that our construction is selectively secure. We remark that our construction achieves semi-adaptive security as is and the proof is almost the same.

Theorem 1 (Selectively secure ABE for NFA$^{\oplus_p}$). *The ABE scheme for NFA$^{\oplus_p}$ in prime-order bilinear groups described above is selectively secure (cf. Section 2.1) under the k-Lin assumption with security loss $O(\ell \cdot |\Sigma|)$. Here ℓ is the length of the challenge input x^*.*

4.3 Game Sequence

The proof is analogous to GWW's proof. We show the proof in the one-key setting where the adversary asks for at most one secret key; this is sufficient to motivate the proof in the next section. As in [11], it is straightforward to handle many keys, see the full paper for more details. Let $x^* \in \Sigma^\ell$ denote the selective challenge and let $\bar{\ell} = \ell \bmod 2$. Without loss of generality, we assume $\ell > 1$. We begin with some auxiliary distributions.

Auxiliary distributions. We describe the auxiliary ciphertext and key distributions that we use in the proof. Throughout, the distributions are the same as the original distributions except for the so-called $\mathbf{a}_2$-components which is defined as below.

$\mathbf{a}_2$-*components.* For a ciphertext in the following form, capturing real and all auxiliary ciphertexts (defined below):

$$\mathsf{ct}_x = \begin{pmatrix} [\mathbf{c}_0]_1,\ [\mathbf{c}_0\mathbf{W}_{\text{start}}]_1 \\ \left\{[\mathbf{c}_j\mathbf{A}_1]_1, [\mathbf{c}_{j-1}\mathbf{Z}_{j \bmod 2} + \mathbf{c}_j\mathbf{W}_{x_j,j \bmod 2}]_1\right\}_j \\ [\mathbf{c}_\ell]_1, [\mathbf{c}_\ell\mathbf{W}_{\text{end}}]_1,\ [\mathbf{c}_\ell\mathbf{k}^\top]_T \cdot m \end{pmatrix} \tag{20}$$

where $\mathbf{c}_j = \mathbf{s}_j\mathbf{A}_1 + s_j\mathbf{a}_2 + \tilde{\mathbf{s}}_j\mathbf{A}_3$ with $\mathbf{s}_j, \tilde{\mathbf{s}}_j \in \mathbb{Z}_p^k$ and $s_j \in \mathbb{Z}_p$, we define its $\mathbf{a}_2$-components, denoted by $\mathsf{ct}_x[2]$, as follows:

$$\mathsf{ct}_x[2] = \begin{pmatrix} [s_0]_1,\ [s_0\mathbf{a}_2\mathbf{W}_{\text{start}}]_1 \\ \left\{[s_j]_1, [s_{j-1}\mathbf{a}_2\mathbf{Z}_{j \bmod 2} + s_j\mathbf{a}_2\mathbf{W}_{x_j,j \bmod 2}]_1\right\}_j \\ [s_\ell]_1, [s_\ell\mathbf{a}_2\mathbf{W}_{\text{end}}]_1,\ [s_\ell\mathbf{a}_2\mathbf{k}^\top]_T \cdot m \end{pmatrix}.$$

For a key in the following form, capturing real and all auxiliary keys (defined below):

$$\mathsf{sk}_\Gamma = \begin{pmatrix} [\mathbf{k}_0^\top]_2, [\mathbf{r}_0^\top]_2 \\ \left\{\ [\mathbf{K}_b]_2, [\mathbf{K}_{\sigma,b}]_2, [\mathbf{R}]_2\ \right\}_{\sigma,b} \\ [\mathbf{K}_{\text{end}}]_2, [\mathbf{R}]_2 \end{pmatrix} \tag{21}$$

where $\mathbf{k}_0 \in \mathbb{Z}_p^{1\times(2k+1)}, \mathbf{K}_b, \mathbf{K}_{\sigma,b}, \mathbf{K}_{\text{end}} \in \mathbb{Z}_p^{(2k+1)\times Q}$ and $\mathbf{r}_0 \in \mathbb{Z}_p^{1\times k}, \mathbf{R} \in \mathbb{Z}_p^{k\times Q}$, we define its $\mathbf{a}_2$-components, denoted by $\mathsf{sk}_\Gamma[2]$, as follows:

$$\mathsf{sk}_\Gamma[2] = \begin{pmatrix} [\mathbf{a}_2\mathbf{k}_0^\top]_2, [\mathbf{r}_0^\top]_2 \\ \left\{\ [\mathbf{a}_2\mathbf{K}_b]_2, [\mathbf{a}_2\mathbf{K}_{\sigma,b}]_2, [\mathbf{R}]_2\ \right\}_{\sigma,b} \\ [\mathbf{a}_2\mathbf{K}_{\text{end}}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

For notation simplicity of $\mathsf{ct}_x[2]$ and $\mathsf{sk}_\Gamma[2]$ with $\mathbf{k}, \mathbf{D}, \mathbf{W}_{\text{start}}, \mathbf{W}_{\text{end}}, \mathbf{Z}_b, \mathbf{W}_{\sigma,b}$, we write

$$\alpha = \mathbf{a}_2\mathbf{k}^\top,\ \mathbf{d} = \mathbf{a}_2\mathbf{D},\ \mathbf{w}_{\text{start}} = \mathbf{a}_2\mathbf{W}_{\text{start}},\ \mathbf{w}_{\text{end}} = \mathbf{a}_2\mathbf{W}_{\text{end}},\ \mathbf{z}_b = \mathbf{a}_2\mathbf{Z}_b,\ \mathbf{w}_{\sigma,b} = \mathbf{a}_2\mathbf{W}_{\sigma,b}$$

and call them the $\mathbf{a}_2$-components of $\mathbf{k}^\top, \mathbf{D}, \mathbf{W}_{\text{start}}, \mathbf{W}_{\text{end}}, \mathbf{Z}_b, \mathbf{W}_{\sigma,b}$, respectively. We also omit zeroes and adjust the order of terms in $\mathsf{ct}_x[2]$. Furthermore, for all $\mathbf{A}_1, \mathbf{a}_2, \mathbf{A}_3$, mpk and various forms of $\mathsf{ct}_x, \mathsf{sk}_\Gamma$ we will use in the proof, we have

$$\begin{aligned} &\mathsf{ct}_x[2],\ \mathsf{sk}_\Gamma[2],\ \{\ \mathbf{A}_i\mathbf{k}^\top, \mathbf{A}_i\mathbf{D}, \mathbf{A}_i\mathbf{W}_{\text{start}}, \mathbf{A}_i\mathbf{W}_{\text{end}}, \mathbf{A}_i\mathbf{Z}_b, \mathbf{A}_i\mathbf{W}_{\sigma,b}\ \}_{i\in\{1,3\},\sigma\in\Sigma,b\in\{0,1\}} \\ \approx_s\ &\mathsf{ct}_x[2],\ \mathsf{sk}_\Gamma[2],\ \{\ \mathbf{A}_i\widetilde{\mathbf{k}}^\top, \mathbf{A}_i\widetilde{\mathbf{D}}, \mathbf{A}_i\widetilde{\mathbf{W}}_{\text{start}}, \mathbf{A}_i\widetilde{\mathbf{W}}_{\text{end}}, \mathbf{A}_i\widetilde{\mathbf{Z}}_b, \mathbf{A}_i\widetilde{\mathbf{W}}_{\sigma,b}\ \}_{i\in\{1,3\},\sigma\in\Sigma,b\in\{0,1\}} \end{aligned}$$

where $\widetilde{\mathbf{k}} \leftarrow \mathbb{Z}_p^{1\times(2k+1)}, \widetilde{\mathbf{D}} \leftarrow \mathbb{Z}_p^{(2k+1)\times Q}, \widetilde{\mathbf{W}}_{\text{start}}, \widetilde{\mathbf{W}}_{\text{end}}, \widetilde{\mathbf{Z}}_b, \widetilde{\mathbf{W}}_{\sigma,b} \leftarrow \mathbb{Z}_p^{(2k+1)\times k}$ are fresh. This follows from Lemma 4 and the fact that all matrices $\mathbf{W} \in \mathbb{Z}_p^{(2k+1)\times k'}$ with $k' \in \mathbb{N}$ can be decomposed as

$$\mathbf{W} = \mathbf{A}_1^\parallel \cdot \mathbf{A}_1\mathbf{W} + \mathbf{a}_2^\parallel \cdot \mathbf{a}_2\mathbf{W} + \mathbf{A}_3^\parallel \cdot \mathbf{A}_3\mathbf{W}.$$

The property allows us to simulate $\mathsf{mpk}, \mathsf{ct}_x, \mathsf{sk}_\Gamma$ from $\mathsf{ct}_x[2], \mathsf{sk}_\Gamma[2]$ and $\mathbf{A}_1, \mathbf{a}_2, \mathbf{A}_3$ so that we can focus on the crucial argument over $\mathbf{a}_2$-components in the proofs, e.g., those in Section 4.4, 4.5 and 4.6.

Ciphertext distributions. We sample $s_0, s_1, \ldots, s_\ell \leftarrow \mathbb{Z}_p$ and define:

- for $i \in [0, \ell]$: $\mathsf{ct}^{i}_{x^*}$ is the same as ct_{x^*} except we replace $\mathbf{s}_i\mathbf{A}_1$ with $\mathbf{s}_i\mathbf{A}_1 + s_i\mathbf{a}_2$;
- for $i \in [\ell]$: $\mathsf{ct}^{i-1,i}_{x^*}$ is the same as ct_{x^*} except we replace $\mathbf{s}_{i-1}\mathbf{A}_1, \mathbf{s}_i\mathbf{A}_1$ with $\mathbf{s}_{i-1}\mathbf{A}_1 + s_{i-1}\mathbf{a}_2, \mathbf{s}_i\mathbf{A}_1 + s_i\mathbf{a}_2$.

That is, we have: writing $\tau = i \bmod 2$,

$$\mathsf{ct}^{i}_{x^*}[2] = \begin{cases} [s_0\mathbf{w}_{\mathsf{start}}]_1,\ [s_0]_1,\ [s_0\mathbf{z}_1]_1 & \text{if } i = 0 \\ [s_i\mathbf{w}_{x^*_i,\tau}]_1,\ [s_i]_1,\ [s_i\mathbf{z}_{1-\tau}]_1 & \text{if } i \in [\ell-1] \\ [s_\ell\mathbf{w}_{x^*_\ell,\bar{\ell}}]_1,\ [s_\ell]_1,\ [s_\ell\mathbf{w}_{\mathsf{end}}]_1,\ [s_\ell\alpha]_T \cdot m_\beta & \text{if } i = \ell \end{cases}$$

$$\mathsf{ct}^{i-1,i}_{x^*}[2] = \begin{cases} [s_0\mathbf{w}_{\mathsf{start}}]_1,\ [s_0]_1,\ [s_0\mathbf{z}_1 + s_1\mathbf{w}_{x^*_1,1}]_1,\ [s_1]_1,\ [s_1\mathbf{z}_0]_1 & \\ & \text{if } i = 1 \\ [s_{i-1}\mathbf{w}_{x^*_{i-1},1-\tau}]_1,\ [s_{i-1}]_1,\ [s_{i-1}\mathbf{z}_\tau + s_i\mathbf{w}_{x^*_i,\tau}]_1,\ [s_i]_1,\ [s_i\mathbf{z}_{1-\tau}]_1 & \\ & \text{if } i \in [2, \ell-1] \\ [s_{\ell-1}\mathbf{w}_{x^*_{\ell-1},1-\bar{\ell}}]_1,\ [s_{\ell-1}]_1,\ [s_{\ell-1}\mathbf{z}_{\bar{\ell}} + s_\ell\mathbf{w}_{x^*_\ell,\bar{\ell}}]_1,\ [s_\ell]_1,\ [s_\ell\mathbf{w}_{\mathsf{end}}]_1,\ [s_\ell\alpha]_T \cdot m_\beta & \\ & \text{if } i = \ell \end{cases}$$

They are exactly the same as those used in GWW's proof [11].

Secret key distributions. Given $x^* \in \Sigma^\ell$ and $\Gamma = (Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma\in\Sigma}, \mathbf{u}, \mathbf{f})$, we define

$$\mathbf{f}_{i,x^*} = \mathbf{f}\mathbf{M}_{x^*_\ell} \cdots \mathbf{M}_{x^*_{i+1}} \bmod p,\ \forall i \in [0, \ell]. \tag{22}$$

For all $i \in [\ell]$, we sample $\Delta \leftarrow \mathbb{Z}_p$ and define:

- sk^0_Γ is the same as sk_Γ except we replace $\mathbf{D}$ with $\mathbf{D} + \mathbf{a}_2^\parallel \cdot s_0^{-1}\Delta \cdot \mathbf{f}_{0,x^*}$ in the term $[\mathbf{D}\mathbf{u}^\top + \mathbf{W}_{\mathsf{start}}\mathbf{R}\mathbf{u}^\top]_2$;
- sk^i_Γ is the same as sk_Γ except we replace $\mathbf{D}$ with $\mathbf{D} + \mathbf{a}_2^\parallel \cdot s_i^{-1}\Delta \cdot \mathbf{f}_{i,x^*}$ in the term $[\mathbf{D}\mathbf{M}_{x^*_i} + \mathbf{W}_{x^*_i, i \bmod 2}\mathbf{R}]_2$;
- $\mathsf{sk}^{i-1,i}_\Gamma$ is the same as sk_Γ except we replace $-\mathbf{D}$ with $-\mathbf{D} + \mathbf{a}_2^\parallel \cdot s_{i-1}^{-1}\Delta \cdot \mathbf{f}_{i-1,x^*}$ in the term $[-\mathbf{D} + \mathbf{Z}_{i \bmod 2}\mathbf{R}]_2$;
- $\mathsf{sk}^{\ell,*}_\Gamma$ is the same as sk_Γ except we replace $-\mathbf{D}$ with $-\mathbf{D} + \mathbf{a}_2^\parallel \cdot s_\ell^{-1}\Delta \cdot \mathbf{f}_{\ell,x^*}$ in the term $[\mathbf{k}^\top\mathbf{f} - \mathbf{D} + \mathbf{W}_{\mathsf{end}}\mathbf{R}]_2$.

That is, we have: writing $\tau = i \bmod 2$,

$$\mathsf{sk}^0_\Gamma[2] = \begin{pmatrix} [(\mathbf{d} + \boxed{s_0^{-1}\Delta \cdot \mathbf{f}_{0,x^*}})\mathbf{u}^\top + \mathbf{w}_{\mathsf{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{d} + \mathbf{z}_b\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,b}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma, b\in\{0,1\}} \\ [\alpha\mathbf{f} - \mathbf{d} + \mathbf{w}_{\mathsf{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

$$\mathsf{sk}^i_\Gamma[2] = \begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + \mathbf{w}_{\mathsf{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{d} + \mathbf{z}_\tau\mathbf{R}]_2, [(\mathbf{d} + \boxed{s_i^{-1}\Delta \cdot \mathbf{f}_{i,x^*}})\mathbf{M}_{x^*_i} + \mathbf{w}_{x^*_i,\tau}\mathbf{R}]_2, [\mathbf{R}]_2\right\} \\ \left\{[\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,\tau}\mathbf{R}]_2\right\}_{\sigma\neq x^*_i} \\ \left\{[-\mathbf{d} + \mathbf{z}_{1-\tau}\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,1-\tau}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma} \\ [\alpha\mathbf{f} - \mathbf{d} + \mathbf{w}_{\mathsf{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

$$\mathsf{sk}_\Gamma^{i-1,i}[2] = \begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{d} + \boxed{s_{i-1}^{-1}\Delta \cdot \mathbf{f}_{i-1,x^*}} + \mathbf{z}_\tau\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,\tau}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma} \\ \left\{[-\mathbf{d} + \mathbf{z}_{1-\tau}\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,1-\tau}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma} \\ [\alpha\mathbf{f} - \mathbf{d} + \mathbf{w}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

$$\mathsf{sk}_\Gamma^{\ell,*}[2] = \begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{d} + \mathbf{z}_b\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,b}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma, b\in\{0,1\}} \\ [\alpha\mathbf{f} - \mathbf{d} + \boxed{s_\ell^{-1}\Delta \cdot \mathbf{f}_{\ell,x^*}} + \mathbf{w}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

They are analogous to those used in GWW's proof [11] with a novel way to change $\mathbf{a}_2$-components[9]. Following the notations in Section 1.2, we use $\mathbf{d}'_i = s_i^{-1}\Delta \cdot \mathbf{f}_{i,x^*}$ rather than $\mathbf{d}'_i = \Delta \cdot \mathbf{f}_{i,x^*}$. We remark that they are essentially the same but the former helps to simplify the exposition of the proof. Also, we note that s_i is independent of the challenge input x^* which will be crucial for the adaptive security in the next section.

Game sequence. As in GWW's proof, we prove Theorem 1 via a series of games summarized in Fig 6:

- G_0: Identical to the real game.
- G_1: Identical to G_0 except that the challenge ciphertext is $\mathsf{ct}_{x^*}^0$.
- $\mathsf{G}_{2.i.0}$, $i \in [\ell]$: In this game, the challenge ciphertext is $\mathsf{ct}_{x^*}^{i-1}$ and the secret key is sk_Γ^{i-1}.
- $\mathsf{G}_{2.i.1}$, $i \in [\ell]$: Identical to $\mathsf{G}_{2.i.0}$ except that the secret key is $\mathsf{sk}_\Gamma^{i-1,i}$.
- $\mathsf{G}_{2.i.2}$, $i \in [\ell]$: Identical to $\mathsf{G}_{2.i.1}$ except that the challenge ciphertext is $\mathsf{ct}_{x^*}^{i-1,i}$.
- $\mathsf{G}_{2.i.3}$, $i \in [\ell]$: Identical to $\mathsf{G}_{2.i.2}$ except that the secret key is sk_Γ^{i}.
- $\mathsf{G}_{2.i.4}$, $i \in [\ell]$: Identical to $\mathsf{G}_{2.i.3}$ except that the challenge ciphertext is $\mathsf{ct}_{x^*}^{i}$.
- G_3: Identical to $\mathsf{G}_{2.\ell.4}$ except that secret key is $\mathsf{sk}_\Gamma^{\ell,*}$.

Note that $\mathsf{G}_{2.1.0}$ is identical to G_1 except that the secret key is sk_Γ^0 and we have $\mathsf{G}_{2.i.0} = \mathsf{G}_{2.i-1.4}$ for all $i \in [2, \ell]$. The remaining of this section will be devoted to proving the indistinguishability of each pair of adjacent games described above. The proofs will be analogous to those for GWW, however, crucially use the property of $\mathbf{f}_{0,x^*}, \ldots, \mathbf{f}_{\ell,x^*}$. Due to lack of space, we focus on proofs using the properties; other proofs are completely analogous to GWW and can be found in the full paper.

Useful lemmas. Before proceed to the proof, we show the next lemma describing the property of $\mathbf{f}_{0,x^*}, \ldots, \mathbf{f}_{\ell,x^*}$.

Lemma 5 (Property of $\{\mathbf{f}_{i,x^*}\}_{i\in[0,\ell]}$). *For any NFA$^{\oplus_p}$ $\Gamma = (Q, \Sigma, \{\mathbf{M}_\sigma\}, \mathbf{u}, \mathbf{f})$ and input $x^* \in \Sigma^\ell$, we have:*

1. $\Gamma(x^*) = 0 \iff \mathbf{f}_{0,x^*}\mathbf{u}^\top = 0 \bmod p$;
2. $\mathbf{f}_{i-1,x^*} = \mathbf{f}_{i,x^*}\mathbf{M}_{x_i^*} \bmod p$ *for all* $i \in [\ell]$;
3. $\mathbf{f}_{\ell,x^*} = \mathbf{f}$.

Proof. The lemma directly follows from the definitions of NFA$^{\oplus_p}$ in Section 3 and $\mathbf{f}_{0,x^*}, \ldots, \mathbf{f}_{\ell,x^*}$ in (22). □

[9] We also change the definition of sk_Γ^i, $i \in [0, \ell]$, with the goal of improving the exposition.

Game	ct_{x^*}	$\mathsf{sk}_\Gamma[2]$							**Remark**
			$? \cdot \mathbf{u}^\top + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top$	$? \cdot \mathbf{M}_{x^*_{i-1}} + \mathbf{w}_{x^*_{i-1},1-\tau}\mathbf{R}$	$? + \mathbf{z}_\tau\mathbf{R}$	$? \cdot \mathbf{M}_{x^*_i} + \mathbf{w}_{x^*_i,\tau}\mathbf{R}$	$\alpha\mathbf{f} + ? + \mathbf{z}_{\text{end}}\mathbf{R}$		
0	ct_{x^*}	sk_Γ	$\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$		real game
1	$\boxed{\mathsf{ct}^0_{x^*}}$	sk_Γ	$\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$		SD
2.1.0	$\mathsf{ct}^0_{x^*}$	$\boxed{\mathsf{sk}^0_\Gamma}$	$\mathbf{d} + \boxed{s_0^{-1}\Delta \cdot \mathbf{f}_{0,x^*}}$	$\mathbf{d}$	$-\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$		Lem 5 - 1
2.i.0	$\mathsf{ct}^{i-1}_{x^*}$	sk^{i-1}_Γ	$\mathbf{d}$	$\mathbf{d} + s_{i-1}^{-1}\Delta \cdot \mathbf{f}_{i-1,x^*}$	$-\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$		$i \in [2, \ell]$
2.i.1	$\mathsf{ct}^{i-1}_{x^*}$	$\boxed{\mathsf{sk}^{i-1,i}_\Gamma}$	$\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d} + \boxed{s_{i-1}^{-1}\Delta \cdot \mathbf{f}_{i-1,x^*}}$	$\mathbf{d}$	$-\mathbf{d}$		change of variables + DDH
2.i.2	$\boxed{\mathsf{ct}^{i-1,i}_{x^*}}$	$\mathsf{sk}^{i-1,i}_\Gamma$	$\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d} + s_{i-1}^{-1}\Delta \cdot \mathbf{f}_{i-1,x^*}$	$\mathbf{d}$	$-\mathbf{d}$		switching lemma
2.i.3	$\mathsf{ct}^{i-1,i}_{x^*}$	$\boxed{\mathsf{sk}^i_\Gamma}$	$\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$	$\mathbf{d} + \boxed{s_i^{-1}\Delta \cdot \mathbf{f}_{i,x^*}}$	$-\mathbf{d}$		transition lemma, Lem 5 - 2
2.i.4	$\boxed{\mathsf{ct}^i_{x^*}}$	sk^i_Γ	$\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$	$\mathbf{d} + s_i^{-1}\Delta \cdot \mathbf{f}_{i,x^*}$	$-\mathbf{d}$		switching lemma
3	$\mathsf{ct}^\ell_{x^*}$	$\boxed{\mathsf{sk}^{\ell,*}_\Gamma}$	$\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d}$	$\mathbf{d}$	$-\mathbf{d} + \boxed{s_\ell^{-1}\Delta \cdot \mathbf{f}_{\ell,x^*}}$		change of variables + DDH

Fig. 6. Game sequence for our selectively secure ABE for $\text{NFA}^{\oplus_p}$ where $i \in [\ell]$. In the table, we only show the $\mathbf{a}_2$-components of secret key. In the **Remark** column, "SD" and "DDH" indicate $\text{SD}^{G_1}_{\mathbf{A}_1 \mapsto \mathbf{A}_1, \mathbf{a}_2}$ and $\text{DDH}^{G_2}_{1,Q}$ assumption, respectively; switching lemma and transition lemma were given in GWW, cf. Lemma 7 and the full paper; "Lem 5 - x" refers to bullet x in Lemma 5.

4.4 Initializing

It is standard to prove $\mathsf{G}_0 \approx_c \mathsf{G}_1$, see the full paper. We only show the proof sketch for $\mathsf{G}_1 \approx_c \mathsf{G}_{2.1.0}$.

Lemma 6 ($\mathsf{G}_1 = \mathsf{G}_{2.1.0}$)**.** *For all $\mathcal{A}$, we have*

$$\Pr[\langle \mathcal{A}, \mathsf{G}_1 \rangle = 1] = \Pr[\langle \mathcal{A}, \mathsf{G}_{2.1.0} \rangle = 1].$$

Proof. Roughly, we will prove that

$$\left(\, \mathsf{mpk},\, \mathsf{ct}^0_{x^*},\, \boxed{\mathsf{sk}_\Gamma} \,\right) = \left(\, \mathsf{mpk},\, \mathsf{ct}^0_{x^*},\, \mathsf{sk}^0_\Gamma \,\right)$$

where we have

$$\mathsf{sk}_\Gamma[2] = \begin{pmatrix} [\boxed{\mathbf{d}\mathbf{u}^\top} + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{d} + \mathbf{z}_b\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,b}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma, b\in\{0,1\}} \\ [\alpha\mathbf{f} - \mathbf{d} + \mathbf{w}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix},$$

$$\mathsf{sk}^0_\Gamma[2] = \begin{pmatrix} [(\mathbf{d} + s_0^{-1}\Delta \cdot \mathbf{f}_{0,x^*})\mathbf{u}^\top + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{d} + \mathbf{z}_b\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,b}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma, b\in\{0,1\}} \\ [\alpha\mathbf{f} - \mathbf{d} + \mathbf{w}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix},$$

and

$$\mathsf{ct}^0_{x^*}[2] = \left(\, [s_0\mathbf{w}_{\text{start}}]_1,\, [s_0]_1,\, [s_0\mathbf{z}_1]_1 \,\right).$$

This follows from the statement:

$$\overbrace{\{\boxed{\mathbf{d}\mathbf{u}^\top} + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top, \mathbf{R}\mathbf{u}^\top\}}^{\mathsf{sk}_\Gamma[2]} = \overbrace{\{(\mathbf{d} + s_0^{-1}\Delta \cdot \mathbf{f}_{0,x^*})\mathbf{u}^\top + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top, \mathbf{R}\mathbf{u}^\top\}}^{\mathsf{sk}^0_\Gamma[2]} \text{ given } \mathbf{d}, \overbrace{\mathbf{w}_{\text{start}}}^{\mathsf{ct}^0_{x^*}[2]}$$

which is implied by the fact $\Gamma(x^*) = 0 \iff \mathbf{f}_{0,x^*}\mathbf{u}^\top = 0 \bmod p$ (see Lemma 5). This is sufficient for the proof. □

4.5 Switching secret keys II

This section proves $\mathsf{G}_{2.i.2} \approx_c \mathsf{G}_{2.i.3}$ for all $i \in [\ell]$ using the the transition lemma from GWW [11].

Lemma 7 ($(\mathbf{z}, \mathbf{w})$**-transition lemma [11]).** *For all $s_{i-1}, s_i \neq 0$ and $\bar{\Delta} \in \mathbb{Z}_p$, we have*

$$\begin{aligned} &\mathsf{aux},\ s_{i-1}\mathbf{z} + s_i\mathbf{w},\ [\boxed{s_{i-1}^{-1}\bar{\Delta}} + \mathbf{z}\mathbf{r}^\top]_2,\ [\mathbf{w}\mathbf{r}^\top]_2,\ [\mathbf{r}^\top]_2 \\ \approx_c\ &\mathsf{aux},\ s_{i-1}\mathbf{z} + s_i\mathbf{w},\ [\mathbf{z}\mathbf{r}^\top]_2,\ [\boxed{s_i^{-1}\bar{\Delta}} + \mathbf{w}\mathbf{r}^\top]_2,\ [\mathbf{r}^\top]_2 \end{aligned}$$

where $\mathsf{aux} = ([\mathbf{z}\mathbf{B}, \mathbf{w}\mathbf{B}, \mathbf{B}]_2)$ *and* $\mathbf{z}, \mathbf{w} \leftarrow \mathbb{Z}_p^{1\times k}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{k\times k}$, $\mathbf{r} \leftarrow \mathbb{Z}_p^{1\times k}$. *Concretely, the advantage function* $\mathsf{Adv}^{\text{TRANS}}_{\mathcal{B}}(\lambda)$ *is bounded by* $O(1)\cdot\mathsf{Adv}^{k\text{-LIN}}_{\mathcal{B}_0}(\lambda)$ *with* $\mathsf{Time}(\mathcal{B}_0) \approx \mathsf{Time}(\mathcal{B})$.

Lemma 8 ($\mathsf{G}_{2.i.2} \approx_c \mathsf{G}_{2.i.3}$)**.** *For all $i \in [\ell]$ and all $\mathcal{A}$, there exists $\mathcal{B}$ with* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$ *such that*

$$\Pr[\langle \mathcal{A}, \mathsf{G}_{2.i.2} \rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{G}_{2.i.3} \rangle = 1] \leq \mathsf{Adv}^{\text{TRANS}}_{\mathcal{B}}(\lambda).$$

Overview. This roughly means

$$\left(\mathsf{mpk},\, \mathsf{ct}_{x^*}^{i-1,i},\, \boxed{\mathsf{sk}_\Gamma^{i-1,i}}\right) \approx_c \left(\mathsf{mpk},\, \mathsf{ct}_{x^*}^{i-1,i},\, \mathsf{sk}_\Gamma^{i}\right);$$

more concretely, we want to prove the following statement over $\mathbf{a}_2$-components:

$$\begin{aligned} &[-\mathbf{d} + \boxed{s_{i-1}^{-1}\Delta \cdot \mathbf{f}_{i-1,x^*} + \mathbf{z}_\tau \mathbf{R}}]_2, && [\mathbf{d}\mathbf{M}_{x_i^*} + \boxed{\mathbf{w}_{x_i^*,\tau}\mathbf{R}}]_2, [\mathbf{R}]_2 \\ \approx_c\ & [-\mathbf{d} + \mathbf{z}_\tau \mathbf{R}]_2, && [\mathbf{d}\mathbf{M}_{x_i^*} + s_i^{-1}\Delta \cdot \mathbf{f}_{i,x^*}\mathbf{M}_{x_i^*} + \mathbf{w}_{x_i^*,\tau}\mathbf{R}]_2, [\mathbf{R}]_2 \end{aligned}$$

given $\mathbf{d}, \Delta, s_{i-1}, s_i, s_{i-1}\mathbf{z}_\tau + s_i \mathbf{w}_{x_i^*,\tau}$ revealed by $\mathsf{ct}_{x^*}^{i-1,i}$. The first row corresponds to $\mathsf{sk}_\Gamma^{i-1,i}[2]$ while the second corresponds to $\mathsf{sk}_\Gamma^{i}[2]$. This can be handled by the $(\mathbf{z}_\tau, \mathbf{w}_{x_i^*,\tau})$-transition lemma and the fact that $\mathbf{f}_{i-1,x^*} = \mathbf{f}_{i,x^*}\mathbf{M}_{x_i^*} \bmod p$ (see Lemma 5).

Proof. Recall that $\tau = i \bmod 2$. By Lemma 4, it suffices to prove the lemma over $\mathbf{a}_2$-components which roughly means:

$$\begin{aligned} \mathsf{sk}_\Gamma^{i-1,i}[2] \;=\; & \begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ [-\mathbf{d} + \boxed{s_{i-1}^{-1}\Delta \cdot \mathbf{f}_{i-1,x^*} + \mathbf{z}_\tau \mathbf{R}}]_2, [\mathbf{d}\mathbf{M}_{x_i^*} + \boxed{\mathbf{w}_{x_i^*,\tau}\mathbf{R}}]_2, [\mathbf{R}]_2 \\ \left\{[\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,\tau}\mathbf{R}]_2\right\}_{\sigma \neq x_i^*} \\ \left\{[-\mathbf{d} + \mathbf{z}_{1-\tau}\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,1-\tau}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma \in \Sigma} \\ [\alpha \mathbf{f} - \mathbf{d} + \mathbf{w}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix} \\ \approx_c & \begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ [-\mathbf{d} + \mathbf{z}_\tau \mathbf{R}]_2, [\mathbf{d}\mathbf{M}_{x_i^*} + s_i^{-1}\Delta \cdot \mathbf{f}_{i,x^*}\mathbf{M}_{x_i^*} + \mathbf{w}_{x_i^*,\tau}\mathbf{R}]_2, [\mathbf{R}]_2 \\ \left\{[\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,\tau}\mathbf{R}]_2\right\}_{\sigma \neq x_i^*} \\ \left\{[-\mathbf{d} + \mathbf{z}_{1-\tau}\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,1-\tau}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma \in \Sigma} \\ [\alpha \mathbf{f} - \mathbf{d} + \mathbf{w}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix} = \mathsf{sk}_\Gamma^{i}[2] \end{aligned}$$

in the presence of

$$\mathsf{ct}_{x^*}^{i-1,i}[2] = \begin{cases} [s_0 \mathbf{w}_{\text{start}}]_1, [s_0]_1, [s_0\mathbf{z}_1 + s_1 \mathbf{w}_{x_1^*,1}]_1, [s_1]_1, [s_1\mathbf{z}_0]_1 & \\ & \text{if } i = 1 \\ [s_{i-1}\mathbf{w}_{x_{i-1}^*,1-\tau}]_1, [s_{i-1}]_1, [s_{i-1}\mathbf{z}_\tau + s_i \mathbf{w}_{x_i^*,\tau}]_1, [s_i]_1, [s_i\mathbf{z}_{1-\tau}]_1 & \\ & \text{if } i \in [2, \ell-1] \\ [s_{\ell-1}\mathbf{w}_{x_{\ell-1}^*,1-\bar{\ell}}]_1, [s_{\ell-1}]_1, [s_{\ell-1}\mathbf{z}_{\bar{\ell}} + s_\ell \mathbf{w}_{x_\ell^*,\bar{\ell}}]_1, [s_\ell]_1, [s_\ell \mathbf{w}_{\text{end}}]_1, [s_\ell \alpha]_T \cdot m_\beta & \\ & \text{if } i = \ell \end{cases}$$

One can sample basis $\mathbf{A}_1, \mathbf{a}_2, \mathbf{A}_3, \mathbf{A}_1^\parallel, \mathbf{a}_2^\parallel, \mathbf{A}_3^\parallel$ and trivially simulate mpk, $\mathsf{ct}_{x^*}^{i-1,i}$ and secret key using terms given out above. Furthermore, we prove this using $(\mathbf{z}_\tau, \mathbf{w}_{x_i^*,\tau})$-transition lemma. On input

$$\mathsf{aux}, [\bar{\Delta}_0 + \mathbf{z}_\tau \mathbf{r}^\top]_2, [\bar{\Delta}_1 + \mathbf{w}_{x_i^*,\tau}\mathbf{r}^\top]_2, [\mathbf{r}^\top]_2$$

where $(\bar{\Delta}_0, \bar{\Delta}_1) \in \left\{\boxed{(s_{i-1}^{-1}\bar{\Delta}, 0)}, \boxed{(0, s_i^{-1}\bar{\Delta})}\right\}$ and

$$\mathsf{aux} = (\bar{\Delta}, s_{i-1}, s_i, s_{i-1}\mathbf{z}_\tau + s_i \mathbf{w}_{x_i^*,\tau}, [\mathbf{z}_\tau \mathbf{B}, \mathbf{w}_{x_i^*,\tau}\mathbf{B}, \mathbf{B}]_2)$$

with $\mathbf{z}_\tau, \mathbf{w}_{x_i^*,\tau} \leftarrow \mathbb{Z}_p^{1\times k}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{k\times k}$, $\mathbf{r} \leftarrow \mathbb{Z}_p^{1\times k}$ and $\bar{\Delta} \leftarrow \mathbb{Z}_p$, we sample $\alpha \leftarrow \mathbb{Z}_p, \mathbf{w}_{\text{start}}, \mathbf{z}_{1-\tau}, \mathbf{w}_{\sigma,1-\tau}, \mathbf{w}_{\text{end}} \leftarrow \mathbb{Z}_p^{1\times k}$ for all $\sigma \in \Sigma$ and $\mathbf{w}_{\sigma,\tau} \leftarrow \mathbb{Z}_p^{1\times k}$ for all $\sigma \neq x_i^*$ and proceed as follows:

(Simulating challenge ciphertext) On input (m_0, m_1), we trivially simulate $\mathsf{ct}_{x^*}^{i-1,i}[2]$ using $s_{i-1}, s_i, s_{i-1}\mathbf{z}_\tau + s_i\mathbf{w}_{x_i^*,\tau}$ in aux and α, $\mathbf{w}_{\text{start}}$, $\mathbf{w}_{\sigma,1-\tau}$, $\mathbf{z}_{1-\tau}$, $\mathbf{w}_{\text{end}}$ as well.

(Simulating secret key) On input Γ, we want to return a secret key for Γ in the form:

$$\begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + \mathbf{w}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \boxed{[-\mathbf{d} + \Delta_0 \cdot \mathbf{f}_{i-1,x^*} + \mathbf{z}_\tau\mathbf{R}]_2}, [\mathbf{d}\mathbf{M}_{x_i^*} + \Delta_1 \cdot \mathbf{f}_{i-1,x^*} + \mathbf{w}_{x_i^*,\tau}\mathbf{R}]_2, [\mathbf{R}]_2 \\ \left\{[\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,\tau}\mathbf{R}]_2\right\}_{\sigma\neq x_i^*} \\ \left\{[-\mathbf{d} + \mathbf{z}_{1-\tau}\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,1-\tau}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma} \\ [\alpha\mathbf{f} - \mathbf{d} + \mathbf{w}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

where $(\Delta_0, \Delta_1) \in \left\{ \boxed{(s_{i-1}^{-1}\Delta, 0)}, \boxed{(0, s_i^{-1}\Delta)} \right\}$. Observe that

- when $(\Delta_0, \Delta_1) = \boxed{(s_{i-1}^{-1}\Delta, 0)}$, the distribution is identical to $\boxed{\mathsf{sk}_\Gamma^{i-1,i}[2]}$;
- when $(\Delta_0, \Delta_1) = \boxed{(0, s_i^{-1}\Delta)}$, the distribution is identical to $\boxed{\mathsf{sk}_\Gamma^{i}[2]}$ since $\mathbf{f}_{i-1,x^*} = \mathbf{f}_{i,x^*}\mathbf{M}_{x_i^*} \bmod p$ (see Lemma 5).

We sample $\mathbf{d} \leftarrow \mathbb{Z}_p^{1\times Q}$ and $\widetilde{\mathbf{R}} \leftarrow \mathbb{Z}_p^{k\times Q}$ and implicitly set

$$\Delta = \bar{\Delta}, \quad (\Delta_0, \Delta_1) = (\bar{\Delta}_0, \bar{\Delta}_1) \quad \text{and} \quad \mathbf{R} = \mathbf{r}^\top \cdot \mathbf{f}_{i-1,x^*} + \mathbf{B} \cdot \widetilde{\mathbf{R}}.$$

We then generate the key for Γ as follows:

- We simulate $[\mathbf{R}]_2$ from $[\mathbf{r}^\top]_2, [\mathbf{B}]_2$ and $\mathbf{f}_{i-1,x^*}, \widetilde{\mathbf{R}}$.
- We rewrite the terms in the dashed box as follows:

  $$[-\mathbf{d} + (\bar{\Delta}_0 + \mathbf{z}_\tau\mathbf{r}^\top)\cdot\mathbf{f}_{i-1,x^*} + \mathbf{z}_\tau\mathbf{B}\cdot\widetilde{\mathbf{R}}]_2,\ [\mathbf{d}\mathbf{M}_{x_i^*} + (\bar{\Delta}_1 + \mathbf{w}_{x_i^*,\tau}\mathbf{r}^\top)\cdot\mathbf{f}_{i-1,x^*} + \mathbf{w}_{x_i^*,\tau}\mathbf{B}\cdot\widetilde{\mathbf{R}}]_2$$

  and simulate them using $[\bar{\Delta}_0 + \mathbf{z}_\tau\mathbf{r}^\top]_2, [\bar{\Delta}_1 + \mathbf{w}_{x_i^*,\tau}\mathbf{r}^\top]_2, [\mathbf{z}_\tau\mathbf{B}]_2, [\mathbf{w}_{x_i^*,\tau}\mathbf{B}]_2$ and $\mathbf{d}, \mathbf{f}_{i-1,x^*}, \widetilde{\mathbf{R}}$.
- We simulate all remaining terms using $[\mathbf{R}]_2$ and α, $\mathbf{d}$, $\mathbf{w}_{\text{start}}$, $\mathbf{z}_{1-\tau}$, $\{\mathbf{w}_{\sigma,\tau}\}_{\sigma\neq x_i^*}$, $\{\mathbf{w}_{\sigma,1-\tau}\}_{\sigma\in\Sigma}$, $\mathbf{w}_{\text{end}}$.

Observe that, when $(\bar{\Delta}_0, \bar{\Delta}_1) = \boxed{(s_{i-1}^{-1}\bar{\Delta}, 0)}$, we have $(\Delta_0, \Delta_1) = \boxed{(s_{i-1}^{-1}\Delta, 0)}$, then the secret key is $\boxed{\mathsf{sk}_\Gamma^{i-1,i}[2]}$ and the simulation is identical to $\mathsf{G}_{2.i.2}$; when $(\bar{\Delta}_0, \bar{\Delta}_1) = \boxed{(0, s_i^{-1}\bar{\Delta})}$, we have $(\Delta_0, \Delta_1) = \boxed{(0, s_i^{-1}\Delta)}$, then the secret key is $\boxed{\mathsf{sk}_\Gamma^{i}[2]}$ and the simulation is identical to $\mathsf{G}_{2.i.3}$. This completes the proof. □

4.6 Finalize

We finally prove that the adversary wins G_3 with probability 1/2.

Lemma 9. $\Pr[\langle \mathcal{A}, \mathsf{G}_3 \rangle = 1] \approx 1/2$.

Proof. First, we argue that the secret key $\mathsf{sk}_\Gamma^{\ell,*}$ in this game perfectly hides the $\mathbf{a}_2$-component of $\mathbf{k}^\top$, i.e., $\alpha = \mathbf{a}_2\mathbf{k}^\top$. Recall the $\mathbf{a}_2$-components of the secret key:

$$\mathsf{sk}_\Gamma^{\ell,*}[2] = \begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + \mathbf{w}_{\mathsf{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \{[-\mathbf{d} + \mathbf{z}_b\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,b}\mathbf{R}]_2, [\mathbf{R}]_2\}_{\sigma\in\Sigma, b\in\{0,1\}} \\ [\alpha\mathbf{f} - \mathbf{d} + \boxed{s_\ell^{-1}\Delta \cdot \mathbf{f}_{\ell,x^*}} + \mathbf{w}_{\mathsf{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}.$$

By the property $\mathbf{f}_{\ell,x^*} = \mathbf{f}$ (see Lemma 5), we can see that $\mathsf{sk}_\Gamma^{\ell,*}[2]$ can be simulated using $\alpha + s_\ell^{-1}\Delta$, which means the secret key perfectly hides $\alpha = \mathbf{a}_2\mathbf{k}^\top$. Therefore, the unique term involving $\mathbf{k}$ in $\mathsf{ct}_{x^*}^\ell$, i.e., $[\mathbf{s}_\ell\mathbf{A}_1\mathbf{k}^\top + s_\ell\mathbf{a}_2\mathbf{k}^\top]_T$, is independently and uniformly distributed and thus statistically hides message m_β. □

5 Adaptively Secure ABE for $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$ and DFA

In this section, we present our adaptively secure ABE for $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$. By our transformation from DFA to $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$ (cf. Lemma 1), this readily gives us an adaptively secure ABE for DFA. We defer the concrete construction to the full paper.

Overview. Our starting point is the selectively secure ABE scheme in Section 4. To achieve adaptive security, we handle key queries one by one following standard dual system method [20]; for each key, we carry out the one-key selective proof in Section 4 with piecewise guessing framework [15]. However this does not work immediately, we will make some changes to the scheme and proof in Section 4.

Recall that, in the one-key setting, the (selective) proof in Section 4 roughly tells us

$$(\mathsf{mpk}, \mathsf{sk}_\Gamma, \mathsf{ct}_{x^*}) \approx_c (\mathsf{mpk}, \boxed{\mathsf{sk}_\Gamma^{\ell,*}}, \boxed{\mathsf{ct}_{x^*}^\ell}). \tag{23}$$

The two-key setting, for example, is expected to be handled by hybrid arguments:

$$(\mathsf{mpk}, \mathsf{sk}_{\Gamma_1}, \mathsf{sk}_{\Gamma_2}, \mathsf{ct}_{x^*}) \approx_c (\mathsf{mpk}, \boxed{\mathsf{sk}_{\Gamma_1}^{\ell,*}}, \mathsf{sk}_{\Gamma_2}, \boxed{\mathsf{ct}_{x^*}^\ell}) \approx_c (\mathsf{mpk}, \mathsf{sk}_{\Gamma_1}^{\ell,*}, \boxed{\mathsf{sk}_{\Gamma_2}^{\ell,*}}, \mathsf{ct}_{x^*}^\ell)$$

The first step seems to be feasible with some natural extension but the second one is problematic. Since we can not switch the challenge ciphertext back to ct_{x^*} due to the presence of $\mathsf{sk}_{\Gamma_1}^{\ell,*}$, the argument (23) can not be applied to the second key sk_{Γ_2} literally. In more detail, recall that

$$\mathsf{ct}_{x^*}^\ell[2] = \left([s_\ell \mathbf{w}_{x_\ell^*,\bar{\ell}}]_1, [s_\ell]_1, [s_\ell \mathbf{w}_{\mathsf{end}}]_1 \right) \tag{24}$$

leaks information of $\mathbf{w}_{x_\ell^*,\bar{\ell}}$ and $\mathbf{w}_{\mathsf{end}}$ while we need them to be hidden in some steps of the one-key proof; for example, Lemma 4.5 for $\mathsf{G}_{2.i.2} \approx_c \mathsf{G}_{2.i.3}$. We quickly argue

that the natural solution of adding an extra subspace for fresh copies of $\mathbf{w}_{x^*_\ell,\bar{\ell}}$ and $\mathbf{w}_{\mathsf{end}}$ blows up the ciphertext and key sizes (see Section 1.1 for discussion).

Our approach reuses the existing $\mathbf{a}_2$-components as in [8]. Recall that, our one-key proof (23) uses a series of hybrids with random coins $s_0, s_1, \ldots$ and finally stops at a hybrid with s_ℓ (cf. (23) and (24)). Roughly, we change the scheme by adding an extra random coin s into the ciphertext and move one more step in the proof so that we finally stop at a new hybrid with the new s only. This allows us to release s_ℓ and reuse $\mathbf{w}_{x^*_\ell,\bar{\ell}}, \mathbf{w}_{\mathsf{end}}$ for the next key. More concretely, starting with the scheme in Section 4.2, we introduce a new component $[\mathbf{W}]_1 \in G_1^{(2k+1)\times k}$ into mpk:

- during encryption, we pick one more random coin $\mathbf{s} \leftarrow \mathbb{Z}_p^{1\times k}$ and replace the last three components in ct_x with

  $$[\mathbf{s}\mathbf{A}_1]_1, [\mathbf{s}_\ell\mathbf{A}_1\mathbf{W}_{\mathsf{end}} + \mathbf{s}\mathbf{A}_1\mathbf{W}]_1, [\mathbf{s}\mathbf{A}_1\mathbf{k}^\top]_T \cdot m;$$

  this connects the last random coin $\mathbf{s}_\ell$ with the newly introduced $\mathbf{s}$; and $\mathbf{s}$ corresponds to s in the proof;
- during key generation, we replace the last two components in sk_Γ with

  $$[-\mathbf{D} + \mathbf{W}_{\mathsf{end}}\mathbf{R}]_2, [\mathbf{k}^\top\mathbf{f} + \mathbf{W}\mathbf{R}]_2, [\mathbf{R}]_2;$$

  the decryption will recover $[\mathbf{s}\mathbf{A}_1\mathbf{k}^\top\mathbf{f} - \mathbf{s}_\ell\mathbf{A}_1\mathbf{D}]_T$ instead of $[\mathbf{s}_\ell\mathbf{A}_1\mathbf{k}^\top\mathbf{f} - \mathbf{s}_\ell\mathbf{A}_1\mathbf{D}]_T$;
- during the proof, we extend the proof in Section 4.3 by one more step (see the dashed box):

  $$(\mathsf{mpk}, \mathsf{sk}_\Gamma, \mathsf{ct}_{x^*}) \overset{\S 4.3}{\approx_c} (\mathsf{mpk}, \boxed{\mathsf{sk}^{\ell,*}_\Gamma}, \boxed{\mathsf{ct}^{\ell}_{x^*}}) \overset{!}{\approx_c} (\mathsf{mpk}, \boxed{\mathsf{sk}^{*}_\Gamma}, \boxed{\mathsf{ct}^{*}_{x^*}})$$

  so that $\mathsf{ct}^*_{x^*}[2]$ is in the following form:

  $$\mathsf{ct}^*_{x^*}[2] = \big([s\mathbf{w}]_1, [s]_1, [s\alpha]_1 \cdot m_\beta\big)$$

  which leaks $\mathbf{w} = \mathbf{a}_2\mathbf{W}$ instead of $\mathbf{w}_{x^*_\ell,\bar{\ell}}, \mathbf{w}_{\mathsf{end}}$; by this, we can carry out the one-key proof (23) for the next key (with some natural extensions).

Conceptually, we can interpret this as letting the NFA move to a specific dummy state whenever it accepts the input. Such a modification has been mentioned in [4] for simplifying the description rather than improving security and efficiency. In our formal description below, we will rename $\mathbf{W}_{\mathsf{end}}, \mathbf{W}, \mathbf{s}, s$ as $\mathbf{Z}_{\mathsf{end}}, \mathbf{W}_{\mathsf{end}}, \mathbf{s}_{\mathsf{end}}, s_{\mathsf{end}}$, respectively.

5.1 Scheme

Our adaptively secure ABE for $\mathcal{E}_Q$-restricted $\mathsf{NFA}^{\oplus_p}$ in prime-order groups use the same basis as described in Section 4.1 and is described as follows:

- $\mathsf{Setup}(1^\lambda, \Sigma)$: Run $\mathbb{G} = (p, G_1, G_2, G_T, e) \leftarrow \mathcal{G}(1^\lambda)$. Sample

  $$\mathbf{A}_1 \leftarrow \mathbb{Z}_p^{k\times(2k+1)}, \mathbf{k} \leftarrow \mathbb{Z}_p^{1\times(2k+1)}, \mathbf{W}_{\mathsf{start}}, \mathbf{Z}_b, \mathbf{W}_{\sigma,b}, \mathbf{Z}_{\mathsf{end}}, \mathbf{W}_{\mathsf{end}} \leftarrow \mathbb{Z}_p^{(2k+1)\times k}$$

  for all $\sigma \in \Sigma$ and $b \in \{0,1\}$. Output

  $$\mathsf{mpk} = \big([\mathbf{A}_1, \mathbf{A}_1\mathbf{W}_{\mathsf{start}}, \{\mathbf{A}_1\mathbf{Z}_b, \mathbf{A}_1\mathbf{W}_{\sigma,b}\}_{\sigma\in\Sigma,b\in\{0,1\}}, \mathbf{A}_1\mathbf{Z}_{\mathsf{end}}, \mathbf{A}_1\mathbf{W}_{\mathsf{end}}]_1, [\mathbf{A}_1\mathbf{k}^\top]_T\big)$$
  $$\mathsf{msk} = \big(\mathbf{k}, \mathbf{W}_{\mathsf{start}}, \{\mathbf{Z}_b, \mathbf{W}_{\sigma,b}\}_{\sigma\in\Sigma,b\in\{0,1\}}, \mathbf{Z}_{\mathsf{end}}, \mathbf{W}_{\mathsf{end}}\big).$$

– $\mathsf{Enc}(\mathsf{mpk}, x, m)$: Let $x = (x_1, \ldots, x_\ell) \in \Sigma^\ell$ and $m \in G_T$. Pick $\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_\ell, \mathbf{s}_{\text{end}} \leftarrow \mathbb{Z}_p^{1\times k}$ and output

$$\mathsf{ct}_x = \begin{pmatrix} [\mathbf{s}_0\mathbf{A}_1]_1,\ [\mathbf{s}_0\mathbf{A}_1\mathbf{W}_{\text{start}}]_1 \\ \left\{[\mathbf{s}_j\mathbf{A}_1]_1, [\mathbf{s}_{j-1}\mathbf{A}_1\mathbf{Z}_{j \bmod 2} + \mathbf{s}_j\mathbf{A}_1\mathbf{W}_{x_j, j \bmod 2}]_1\right\}_{j\in[\ell]} \\ [\mathbf{s}_{\text{end}}\mathbf{A}_1]_1, [\mathbf{s}_\ell\mathbf{A}_1\mathbf{Z}_{\text{end}} + \mathbf{s}_{\text{end}}\mathbf{A}_1\mathbf{W}_{\text{end}}]_1,\ [\mathbf{s}_{\text{end}}\mathbf{A}_1\mathbf{k}^\top]_T \cdot m \end{pmatrix}.$$

– $\mathsf{KeyGen}(\mathsf{mpk}, \mathsf{msk}, \Gamma)$: Let $\Gamma = (Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma\in\Sigma}, \mathbf{u}, \mathbf{f})$. Pick $\mathbf{D} \leftarrow \mathbb{Z}_p^{(2k+1)\times Q}$, $\mathbf{R} \leftarrow \mathbb{Z}_p^{k\times Q}$ and output

$$\mathsf{sk}_\Gamma = \begin{pmatrix} [\mathbf{D}\mathbf{u}^\top + \mathbf{W}_{\text{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{D} + \mathbf{Z}_b\mathbf{R}]_2, [\mathbf{D}\mathbf{M}_\sigma + \mathbf{W}_{\sigma,b}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma, b\in\{0,1\}} \\ [-\mathbf{D} + \mathbf{Z}_{\text{end}}\mathbf{R}]_2, [\mathbf{k}^\top\mathbf{f} + \mathbf{W}_{\text{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}.$$

– $\mathsf{Dec}(\mathsf{mpk}, \mathsf{sk}_\Gamma, \mathsf{ct}_x)$: Parse ciphertext for $x = (x_1, \ldots, x_\ell)$ and key for $\Gamma = (Q, \Sigma, \{\mathbf{M}_\sigma\}_{\sigma\in\Sigma}, \mathbf{u}, \mathbf{f})$ as

$$\mathsf{ct}_x = \begin{pmatrix} [\mathbf{c}_{0,1}]_1, [\mathbf{c}_{0,2}]_1 \\ \left\{[\mathbf{c}_{j,1}]_1, [\mathbf{c}_{j,2}]_1\right\}_j \\ [\mathbf{c}_{\text{end},1}]_1, [\mathbf{c}_{\text{end},2}]_1, C \end{pmatrix} \quad \text{and} \quad \mathsf{sk}_\Gamma = \begin{pmatrix} [\mathbf{k}_0^\top]_2, [\mathbf{r}_0^\top]_2 \\ \left\{[\mathbf{K}_b]_2, [\mathbf{K}_{\sigma,b}]_2, [\mathbf{R}]_2\right\}_{\sigma,b} \\ [\mathbf{K}_{\text{end},1}]_2, [\mathbf{K}_{\text{end},2}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

We define $\mathbf{u}_{j,x}^\top$ for all $j \in [0, \ell]$ as (11) in Section 4.2 and proceed as follows:

1. Compute
$$B_0 = e([\mathbf{c}_{0,1}]_1, [\mathbf{k}_0^\top]_2) \cdot e([\mathbf{c}_{0,2}]_1, [\mathbf{r}_0^\top]_2)^{-1};$$
2. For all $j \in [\ell]$, compute
$$[\mathbf{b}_j]_T = e([\mathbf{c}_{j-1,1}]_1, [\mathbf{K}_{j \bmod 2}]_2) \cdot e([\mathbf{c}_{j,1}]_1, [\mathbf{K}_{x_j, j \bmod 2}]_2) \cdot e([-\mathbf{c}_{j,2}]_1, [\mathbf{R}]_2)$$
$$\text{and} \quad B_j = [\mathbf{b}_j\mathbf{u}_{j-1,x}^\top]_T;$$
3. Compute
$$[\mathbf{b}_{\text{end}}]_T = e([\mathbf{c}_{\ell,1}]_1, [\mathbf{K}_{\text{end},1}]_2) \cdot e([\mathbf{c}_{\text{end},1}]_1, [\mathbf{K}_{\text{end},2}]_2) \cdot e([-\mathbf{c}_{\text{end},2}]_1, [\mathbf{R}]_2)$$
$$\text{and} \quad B_{\text{end}} = [\mathbf{b}_{\text{end}}\mathbf{u}_{\ell,x}^\top]_T;$$
4. Compute
$$B_{\text{all}} = B_0 \cdot \textstyle\prod_{j=1}^{\ell} B_j \cdot B_{\text{end}} \quad \text{and} \quad B = B_{\text{all}}^{(\mathbf{f}\mathbf{u}_{\ell,x}^\top)^{-1}}$$
and output the message $m' \leftarrow C \cdot B^{-1}$.

It is direct to verify the correctness as in Section 4.2. See the full paper for more details.

Security. We prove the following theorem stating the adaptive security of the above ABE for $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$. This readily implies our adaptively secure ABE for DFA thanks to Lemma 1.

Theorem 2 (Adaptively seucre ABE for $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$). *The ABE scheme for $\mathcal{E}_Q$-restricted NFA$^{\oplus_p}$ in prime-order bilinear groups described above is adaptively secure (cf. Section 2.1) under the k-Lin assumption with security loss $O(q \cdot \ell \cdot |\Sigma|^3 \cdot Q^2)$. Here ℓ is the length of the challenge input x^* and q is the number of key queries.*

5.2 Proof of Main Theorem

From a high level, we employ the standard dual system proof switching the challenge ciphertext and keys into semi-functional forms in a one-by-one manner. To switch a secret key, we employ the proof technique for one-key selective setting in Section 4 in the piecewise guessing framework [15,14]. We will capture this by a core lemma. Let $x^* \in \Sigma^\ell$ denote the adaptive challenge. We begin with auxiliary distributions and use the notation for $\mathbf{a}_2$-components in Section 4.3.

Auxiliary distributions. We sample $s_{\mathrm{end}} \leftarrow \mathbb{Z}_p$, $\Delta \leftarrow \mathbb{Z}_p$ and define semi-functional ciphertext and key:

- $\mathsf{ct}^*_{x^*}$ is the same as ct_{x^*} except we replace $\mathbf{s}_{\mathrm{end}}\mathbf{A}_1$ with $\mathbf{s}_{\mathrm{end}}\mathbf{A}_1 + s_{\mathrm{end}}\mathbf{a}_2$;
- sk^*_Γ is the same as sk_Γ except we replace $\mathbf{k}^\top$ with $\mathbf{k}^\top + \mathbf{a}_2^\parallel \cdot s_{\mathrm{end}}^{-1}\Delta$ in the term $[\mathbf{k}^\top \mathbf{f} + \mathbf{W}_{\mathrm{end}}\mathbf{R}]_2$.

That is, we have:

$$\mathsf{ct}^*_{x^*}[2] = \left([s_{\mathrm{end}}\mathbf{w}_{\mathrm{end}}]_1,\ [s_{\mathrm{end}}]_1,\ [s_{\mathrm{end}}\alpha]_T \cdot m_\beta \right)$$

$$\mathsf{sk}^*_\Gamma[2] = \begin{pmatrix} [\mathbf{d}\mathbf{u}^\top + \mathbf{w}_{\mathrm{start}}\mathbf{R}\mathbf{u}^\top]_2, [\mathbf{R}\mathbf{u}^\top]_2 \\ \left\{[-\mathbf{d} + \mathbf{z}_b\mathbf{R}]_2, [\mathbf{d}\mathbf{M}_\sigma + \mathbf{w}_{\sigma,b}\mathbf{R}]_2, [\mathbf{R}]_2\right\}_{\sigma\in\Sigma, b\in\{0,1\}} \\ [-\mathbf{d} + \mathbf{z}_{\mathrm{end}}\mathbf{R}]_2, [\alpha\mathbf{f} + \boxed{s_{\mathrm{end}}^{-1}\Delta \cdot \mathbf{f}} + \mathbf{w}_{\mathrm{end}}\mathbf{R}]_2, [\mathbf{R}]_2 \end{pmatrix}$$

Game sequence and core lemma. We prove Theorem 2 via a series of games following standard dual system method [20]:

- G_0: Identical to the real game.
- G_1: Identical to G_0 except that the challenge ciphertext is semi-functional, i.e., $\mathsf{ct}^*_{x^*}$.
- $\mathsf{G}_{2.\kappa}$ for $\kappa \in [0, q]$: Identical to G_1 except that the first κ secret keys are semi-functional, i.e., sk^*_Γ.
- G_3: Identical to $\mathsf{G}_{2.q}$ except that the challenge ciphertext is an encryption of a random message.

Here we have $\mathsf{G}_{2.0} = \mathsf{G}_1$. It is standard to prove $\mathsf{G}_0 \approx_c \mathsf{G}_1$, $\mathsf{G}_{2.q} \approx_s \mathsf{G}_3$ and show that adversary in G_3 has no advantage. We sketch the proofs in the full paper. To prove $\mathsf{G}_{2.\kappa-1} \approx_c \mathsf{G}_{2.\kappa}$ for all $\kappa \in [q]$, we use core lemma:

Lemma 10 (Core lemma). *For all $\mathcal{A}$, there exists $\mathcal{B}$ with* $\mathsf{Time}(\mathcal{B}) \approx \mathsf{Time}(\mathcal{A})$ *and*

$$\mathsf{Adv}^{\text{CORE}}_{\mathcal{A}}(\lambda) = \Pr[\langle \mathcal{A}, \mathsf{H}_0 \rangle = 1] - \Pr[\langle \mathcal{A}, \mathsf{H}_1 \rangle = 1] \leq O(\ell \cdot |\Sigma|^3 \cdot Q^2) \cdot \mathsf{Adv}^{k\text{-LIN}}_{\mathcal{B}}(\lambda)$$

where, for all $b \in \{0,1\}$, we define:

$$\langle \mathcal{A}, \mathsf{H}_b \rangle := \left\{ b' \leftarrow \mathcal{A}^{\mathsf{OEnc}(\cdot),\mathsf{OKey}(\cdot)}(\mathsf{mpk}, \mathsf{aux}_1, \mathsf{aux}_2) \right\}$$

where

$$\begin{aligned}
\mathsf{mpk} &= \left([\mathbf{A}_1, \mathbf{A}_1\mathbf{W}_{start}, \{\mathbf{A}_1\mathbf{Z}_b, \mathbf{A}_1\mathbf{W}_{\sigma,b}\}_{\sigma\in\Sigma,b\in\{0,1\}}, \mathbf{A}_1\mathbf{Z}_{end}, \mathbf{A}_1\mathbf{W}_{end}]_1, [\mathbf{A}_1\mathbf{k}^\top]_T\right) \\
\mathsf{aux}_1 &= \left([\mathbf{k}, \mathbf{B}, \mathbf{W}_{start}\mathbf{B}, \{\mathbf{Z}_b\mathbf{B}, \mathbf{W}_{\sigma,b}\mathbf{B}\}_{\sigma\in\Sigma,b\in\{0,1\}}, \mathbf{Z}_{end}\mathbf{B}, \mathbf{W}_{end}\mathbf{B}]_2\right) \\
\mathsf{aux}_2 &= \left([\mathbf{r}^\top, \mathbf{W}_{start}\mathbf{r}^\top, \{\mathbf{Z}_b\mathbf{r}^\top, \mathbf{W}_{\sigma,b}\mathbf{r}^\top\}_{\sigma\in\Sigma,b\in\{0,1\}}, \mathbf{Z}_{end}\mathbf{r}^\top, \mathbf{a}_2^\parallel \cdot s_{end}^{-1}\Delta + \mathbf{W}_{end}\mathbf{r}^\top]_2\right)
\end{aligned}$$

with $\mathbf{W}_{start}, \mathbf{Z}_0, \mathbf{Z}_1, \mathbf{W}_{\sigma,0}, \mathbf{W}_{\sigma,1}, \mathbf{Z}_{end}, \mathbf{W}_{end} \leftarrow \mathbb{Z}_p^{(2k+1)\times k}$, $\mathbf{B} \leftarrow \mathbb{Z}_p^{k\times k}$, $\mathbf{r} \leftarrow \mathbb{Z}_p^{1\times k}$, $s_{end}, \Delta \leftarrow \mathbb{Z}_p$ and the two oracles work as follows:

- $\mathsf{OEnc}(x^*, m)$*: output* $\mathsf{ct}^*_{x^*}$ *using* s_{end} *in* aux_2*;*
- $\mathsf{OKey}(\Gamma)$*: output* $\boxed{\mathsf{sk}_\Gamma}$ *if* $b = 0$*; output* sk^*_Γ *using* Δ *and* s_{end} *in* aux_2 *if* $b = 1$*;*

with the restrictions that (1) $\mathcal{A}$ makes only one query to each oracle; (2) queries Γ and x^* satisfy $\Gamma(x^*) = 0$.

It is direct to see that the core lemma implies $\mathsf{G}_{2.\kappa-1} \approx_c \mathsf{G}_{2.\kappa}$; here aux_1 and aux_2 are sufficient to simulate other $q-1$ keys which are either sk_Γ or sk^*_Γ, see the full paper for more details.

Acknowledgments. We thank Brent Waters for insightful discussions on adaptive security, as well as the anonymous reviewers for constructive feedback on our write-up.

References

1. S. Agrawal and M. Chase. Simplifying design and analysis of complex predicate encryption schemes. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part I*, volume 10210 of *LNCS*, pages 627–656. Springer, Heidelberg, Apr. / May 2017.
2. S. Agrawal, M. Maitra, and S. Yamada. Attribute based encryption (and more) for non-deterministic finite automata from LWE. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 765–797. Springer, Heidelberg, Aug. 2019.
3. S. Agrawal, M. Maitra, and S. Yamada. Attribute based encryption for deterministic finite automata from DLIN. In *TCC*, 2019.
4. N. Attrapadung. Dual system encryption via doubly selective security: Framework, fully secure functional encryption for regular languages, and more. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 557–577. Springer, Heidelberg, May 2014.
5. N. Attrapadung. Dual system encryption framework in prime-order groups via computational pair encodings. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 591–623. Springer, Heidelberg, Dec. 2016.

6. N. Attrapadung and S. Yamada. Duality in ABE: Converting attribute based encryption for dual predicate and dual policy via computational encodings. In K. Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 87–105. Springer, Heidelberg, Apr. 2015.
7. J. Chen, R. Gay, and H. Wee. Improved dual system ABE in prime-order groups via predicate encodings. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 595–624. Springer, Heidelberg, Apr. 2015.
8. J. Chen, J. Gong, L. Kowalczyk, and H. Wee. Unbounded ABE via bilinear entropy expansion, revisited. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part I*, volume 10820 of *LNCS*, pages 503–534. Springer, Heidelberg, Apr. / May 2018.
9. J. Chen and H. Wee. Semi-adaptive attribute-based encryption and improved delegation for Boolean formula. In M. Abdalla and R. D. Prisco, editors, *SCN 14*, volume 8642 of *LNCS*, pages 277–297. Springer, Heidelberg, Sept. 2014.
10. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. Villar. An algebraic framework for Diffie-Hellman assumptions. In R. Canetti and J. A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, Aug. 2013.
11. J. Gong, B. Waters, and H. Wee. ABE for DFA from k-lin. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part II*, volume 11693 of *LNCS*, pages 732–764. Springer, Heidelberg, Aug. 2019.
12. V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In A. Juels, R. N. Wright, and S. Vimercati, editors, *ACM CCS 06*, pages 89–98. ACM Press, Oct. / Nov. 2006. Available as Cryptology ePrint Archive Report 2006/309.
13. D. Hofheinz, J. Koch, and C. Striecks. Identity-based encryption with (almost) tight security in the multi-instance, multi-ciphertext setting. In J. Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 799–822. Springer, Heidelberg, Mar. / Apr. 2015.
14. Z. Jafargholi, C. Kamath, K. Klein, I. Komargodski, K. Pietrzak, and D. Wichs. Be adaptive, avoid overcommitting. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 133–163. Springer, Heidelberg, Aug. 2017.
15. L. Kowalczyk and H. Wee. Compact adaptively secure ABE for $\mathsf{NC}s^1$ from k-lin. In V. Rijmen and Y. Ishai, editors, *EUROCRYPT 2019, Part I*, LNCS, pages 3–33. Springer, Heidelberg, May 2019.
16. A. B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In D. Micciancio, editor, *TCC 2010*, volume 5978 of *LNCS*, pages 455–479. Springer, Heidelberg, Feb. 2010.
17. A. B. Lewko and B. Waters. Unbounded HIBE and attribute-based encryption. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 547–567. Springer, Heidelberg, May 2011.
18. T. Okamoto and K. Takashima. Fully secure unbounded inner-product and attribute-based encryption. In X. Wang and K. Sako, editors, *ASIACRYPT 2012*, volume 7658 of *LNCS*, pages 349–366. Springer, Heidelberg, Dec. 2012.
19. A. Sahai and B. R. Waters. Fuzzy identity-based encryption. In R. Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 457–473. Springer, Heidelberg, May 2005.
20. B. Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 619–636. Springer, Heidelberg, Aug. 2009.
21. B. Waters. Functional encryption for regular languages. In R. Safavi-Naini and R. Canetti, editors, *CRYPTO 2012*, volume 7417 of *LNCS*, pages 218–235. Springer, Heidelberg, Aug. 2012.