

Document Change Log

No.	Date	Author	Document Name	Changes Made / Description
1.	28 Aug 2025	Z. Mahad	1. KAZ SIGN/3. Source Code 2. KAZ KA/3. Source Code 3. KAZ KEM/3. Source Code	Folder names change from Optimized_Implementation to References_Implementation
2.	28 Aug 2025	Z. Mahad	KAZ SIGN/3. Source Code/ References_Implementation/KAZ Security Level 128/ • kaz_api.h	#define KAZ_DS_SP_qQ "274740016173379194546381236446787565723556071979741740" #define KAZ_DS_SP_G1qQ "109791784035469620464528703986642036404462199422783032019481545979393622480000" #define KAZ_DS_SP_PHIqQ "35111136773400413456929878039434623864922544537600000" #define KAZ_DS_SP_G1qQRHO "63997580811605089901461886588211664277541743781178507127907222480599391279443741157968408096194052214975438487920000"
3.	28 Aug 2025	Z. Mahad	KAZ KA/3. Source Code/ References_Implementation/KAZ Security Level 128/ • kaz_api.h	#define KAZ_KA_SP_g1 "65563" #define KAZ_KA_SP_g2 "65617" #define KAZ_KA_SP_Og1N "277591565561588714832683121976613472000" #define KAZ_KA_SP_LOg1N 128 // in binary (17 bytes) #define KAZ_KA_SP_Og2N "832774696684766144498049365929840416000" #define KAZ_KA_SP_LOg2N 130 // in binary (17 bytes)

				extern void init_random(); extern void clear_random();
4.	28 Aug 2025	Z. Mahad	KAZ KA/3. Source Code/ References_Implementation/KAZ Security Level 192/ <ul style="list-style-type: none"> kaz_api.h 	#define KAZ_KA_SP_g1 "65563" #define KAZ_KA_SP_g2 "65617" #define KAZ_KA_SP_Og1N "51736000959480087314595638140051513827162226171393634016000" #define KAZ_KA_SP_LOg1N 196 // in binary (25 bytes) #define KAZ_KA_SP_Og2N "155208002878440261943786914420154541481486678514180902048000" #define KAZ_KA_SP_LOg2N 197 // in binary (25 bytes) extern void init_random(); extern void clear_random();
5.	28 Aug 2025	Z. Mahad	KAZ KA/3. Source Code/ References_Implementation/KAZ Security Level 256/ <ul style="list-style-type: none"> kaz_api.h 	#define KAZ_KA_SP_g1 "65563" #define KAZ_KA_SP_g2 "65617" #define KAZ_KA_SP_Og1N "99154693887499828557116081873795155652147461554242228686027806044656980768000" #define KAZ_KA_SP_LOg1N 256 // in binary (33 bytes) #define KAZ_KA_SP_Og2N "297464081662499485671348245621385466956442384662726686058083418133970942304000"

				<pre>#define KAZ_KA_SP_LOG2N 258 // in binary (33 bytes) extern void init_random(); extern void clear_random();</pre>
6.	28 Aug 2025	Z. Mahad	<ol style="list-style-type: none"> 1. KAZ KA/3. Source Code/ References_Implementation/KAZ Security Level 128/ 2. KAZ KA/3. Source Code/ References_Implementation/KAZ Security Level 192/ 3. KAZ KA/3. Source Code/ References_Implementation/KAZ Security Level 256/ <ul style="list-style-type: none"> • kaz_api.c 	<pre>// Declare global random state gmp_randstate_t state; // Initialize once void init_random() { unsigned long seed = 123456789UL; // fixed seed for repeatability gmp_randinit_default(state); gmp_randseed_ui(state, seed); } // Cleanup once at the end void clear_random() { gmp_randclear(state); } void KAZ_KEM_RANDOM(mpz_t lb, mpz_t ub, mpz_t out) { mpz_t range, rand_in_range; mpz_inits(range, rand_in_range, NULL); // range = ub - lb + 1 mpz_sub(range, ub, lb); mpz_add_ui(range, range, 1); // rand_in_range ∈ [0, range-1] mpz_urandomm(rand_in_range, state, range); // out = lb + rand_in_range mpz_add(out, lb, rand_in_range);</pre>

				<pre> mpz_clears(range, rand_in_range, NULL); } </pre>
7.	28 Aug 2025	Z. Mahad	KAZ KEM/3. Source Code/ References_Implementation/KAZ Security Level 128/ <ul style="list-style-type: none"> kaz_api.h 	<pre> #define KAZ_KEM_SP_g1 "65563" #define KAZ_KEM_SP_g2 "65617" #define KAZ_KEM_SP_Og1N "277591565561588714832683121976613472000" #define KAZ_KEM_SP_Log1N 128 // in binary (17 bytes) #define KAZ_KEM_SP_Og2N "832774696684766144498049365929840416000" #define KAZ_KEM_SP_Log2N 130 // in binary (17 bytes) extern void init_random(); extern void clear_random(); </pre>
8.	28 Aug 2025	Z. Mahad	KAZ KEM/3. Source Code/ References_Implementation/KAZ Security Level 192/ <ul style="list-style-type: none"> kaz_api.h 	<pre> #define KAZ_KEM_SP_g1 "65563" #define KAZ_KEM_SP_g2 "65617" #define KAZ_KEM_SP_Og1N "51736000959480087314595638140051513827162226171393634016000" #define KAZ_KEM_SP_Log1N 196 // in binary (25 bytes) #define KAZ_KEM_SP_Og2N "155208002878440261943786914420154541481486678514180902048000" </pre>

				<pre>#define KAZ_KEM_SP_Log2N 197 // in binary (25 bytes) extern void init_random(); extern void clear_random();</pre>
9.	28 Aug 2025	Z. Mahad	<p>KAZ KEM/3. Source Code/References_Implementation/KAZ Security Level 256/</p> <ul style="list-style-type: none"> kaz_api.h 	<pre>#define KAZ_KEM_SP_g1 "65563" #define KAZ_KEM_SP_g2 "65617" #define KAZ_KEM_SP_Og1N "99154693887499828557116081873795155652147461554242228686027806044656980768000" #define KAZ_KEM_SP_Log1N 256 // in binary (33 bytes) #define KAZ_KEM_SP_Og2N "297464081662499485671348245621385466956442384662726686058083418133970942304000" #define KAZ_KEM_SP_Log2N 257 // in binary (33 bytes) extern void init_random(); extern void clear_random();</pre>
10.	28 Aug 2025	Z. Mahad	<ol style="list-style-type: none"> KAZ KEM/3. Source Code/References_Implementation/KAZ Security Level 128/ KAZ KEM/3. Source Code/References_Implementation/KAZ Security Level 192/ KAZ KEM/3. Source Code/References_Implementation/KAZ Security Level 256/ 	<pre>// Declare global random state gmp_randstate_t state; // Initialize once void init_random() { unsigned long seed = 123456789UL; // fixed seed for repeatability gmp_randinit_default(state); gmp_randseed_ui(state, seed); }</pre>

			<ul style="list-style-type: none"> • kaz_api.c <pre> // Cleanup once at the end void clear_random() { gmp_randclear(state); } void KAZ_KEM_RANDOM(mpz_t lb, mpz_t ub, mpz_t out) { mpz_t range, rand_in_range; mpz_inits(range, rand_in_range, NULL); // range = ub - lb + 1 mpz_sub(range, ub, lb); mpz_add_ui(range, range, 1); // rand_in_range ∈ [0, range-1] mpz_urandomm(rand_in_range, state, range); // out = lb + rand_in_range mpz_add(out, lb, rand_in_range); mpz_clears(range, rand_in_range, NULL); } </pre>
--	--	--	--