5.2 Key Exchange

V této kapitole si postupně popíšeme fungování algoritmu pro výměnu klíčů, jehož vývojový diagram je zobrazen na straně ??. Algoritmus je založen na elpitockých křivkách a na Weierstrassově metodě výpočtu eliptických křivek, která je popsána v kapitole (===) a je navržen pro práci v dekadické soustavě. Návrh počítá se vstupy, které jsou generovány pseudonáhodně. Vzhledem k tomu to faktu je v algoritmu implantována několika úrovňová kontrola, která vstupní hodnoty podrobuje několika testům, například test prvočíselnosti, které zajistí, že vstupy jsou použitelné. Výstupem výpočtu je grupa bodů, ze které je získán první bod grupy a řád grupy, který představuje počet dvojic bodů. Tyto hodnoty představují základ pro výpočet veřejného klíče na základě klíče tajného klíče, který je v dalším procesu taktéž generován pseudonáhodně. Pro dosažení naprosté jistoty, že získáme použitelnou grupu bodů, je algoritmus dále vybaven rekurzivním testováním, kdy se ze znalosti prvního bodu a řádu grupy zpětně vypočítá celá grupa pomocí for cyklu. Výstupem tohoto testu je důkaz, že vstupní a výstupní hodnoty jsou použitelné pro další fáze výměny klíčů. Slabinou této fáze algoritmu je proměnlivá doba získání správných hodnot, která je ovlivněna dobou za jak dlouho je algoritmus schopen najít takový vstup, který je použitelný. Rozmezí může být až 2 sekundy.

V další fázi výměny klíčů jsou ověřené vstupní hodnoty odeslány ve formě zašifrovaného payloadu uzlu B. Je výhodnější posílat ověřené vstupní hodnoty, z důvodu menšího počtu potřebných dat k odeslání. Tím se šetří energie nutná pro odeslání dat na fyzické vrstvě. Uzel B ze znalosti těchto hodnot sám vypočítá první bod a řád grupy. Tentokrát již není kontrola výpočtu zapotřebí. Po provedení této fáze, jsou oba komunikační uzly schopné přistoupit na výměnu klíče.

Matematické funkce

V průběhu výpočtu je často zapotřebí použití matematických funkcí jako je mocnění a modulární aritmetika. Tyto funkce jsou obsaženy v jazyce C. Ačkoliv tyto funkce kompilátor bez námitek zkompiluje, vlastnosti těchto funkcí nejsou vhodné pro použití v našem algoritmu a jejich použití by mohlo vést k nesprávným výsledkům. Pro zajištění spolehlivosti byly pro nadefinovány následující fuunkce

Funkce pow, která má za úkol mocnění čísel, pracuje s datovým typem double. V našem algoritmu pracujeme pouze s celými čísly. Pokud použijeme na vstupu do funkce pow celá čísla, může to mít v některých situacích nežádoucí výsledek, na příklad druhá mocnina čísla 5 je 24 namísto správného výsledku 25. Pro předejití takovým to problémů byla vytvořena aplikace pro mocnění, která pracuje s celými čísly a je schopna mocnit i záporná čísla. Aplikace je vytvořena jako samostatný modul, který je v procesu výměny klíčů hlavním programem dle potřeby volán.

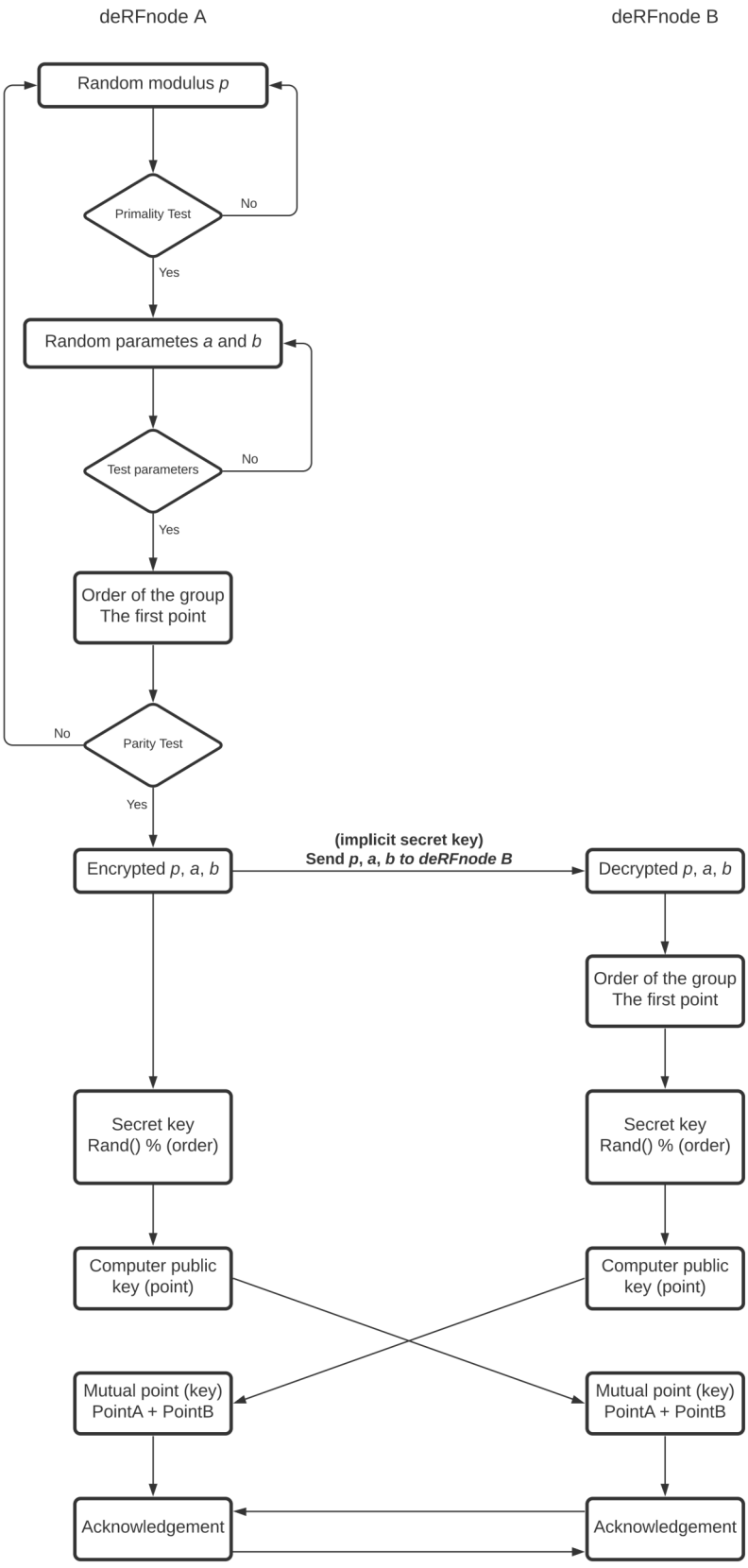

Fig. 5.3: Key exchange algorithm

Další matematickou aplikací je algoritmus pro výpočet zbytku pomocí modulární aritmetiky. Tato funkce je obdobně jako mocnění součástí jazyka C. Ovšem tato funkce nepracuje se zápornými čísly, které se v průběhu výpočtů v našem hlavním algoritmu vyskytují. Proto byla vytvořena aplikace, která po zavolání a obdržení vstupů provede výpočet a vrátí zpět výslednou hodnotu. Vedle potřeby klasického výpočtu zbytku po dělení je tu i požadavek na inverzní variantu. V aplikaci je pomocí cyklu while prováděn výpočet tak dlouho, než se zbytek rovná 1. Počet opakování je zároveň výsledkem. Tato aplikace je opět navržena tak, že bez problému funguje i se zápornými čísly. Poslední z matematických aplikací je test prvočíselnosti. Při inicializaci hlavního algoritmu dochází k volbě hodnoty modulo pomocí pseudonáhodné funkce. Pseudonáhodný výběr generuje jakékoliv reálné číslo. Ovšem pro náš výpočet je nutné, aby hodnota modulo byla prvočíslem.

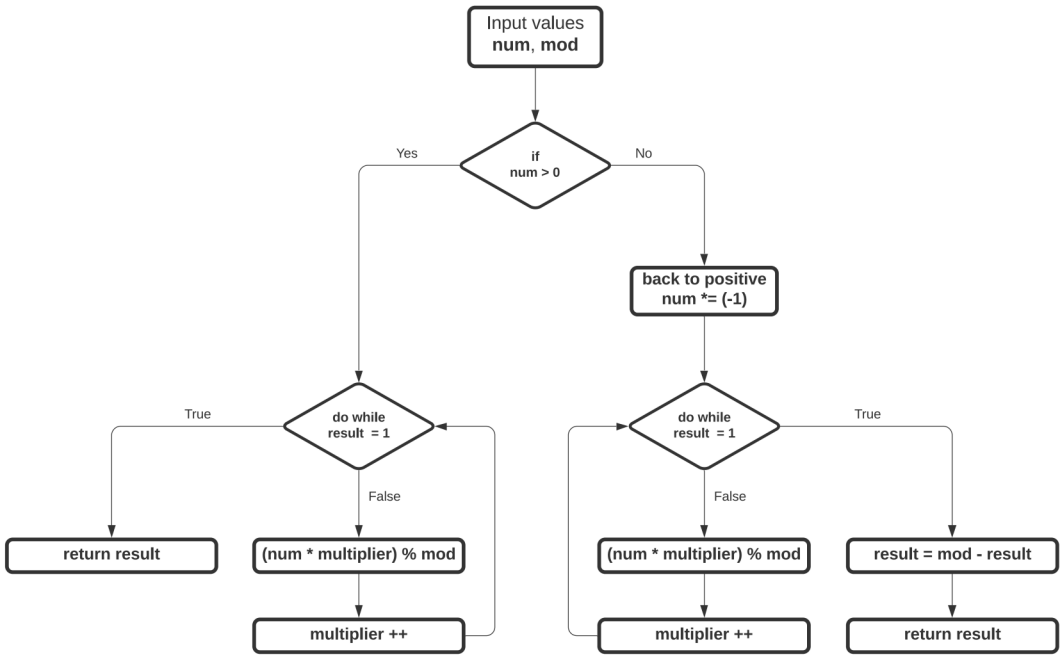

Fig. 5.4: Inverse modulo function [11]

Výpočet prvního bodu a řádu grupy

Použití této části algoritmu je podmíněno úspěšným průběhem předchozí části algoritmu, který vygeneroval zaručeně funkční grupu bodů a je tedy znám první bod grupy a jeho řád. Výpočet bodu, který je součtem dvou bodů probíhá následujícím způsobem. V úvodní části výpočtu algoritmus na základě hodnot rozhodne, zda se jedná o součet dvou stejných bodů nebo dvou rozdílných bodů, popřípadě zda součet dle teorie eliptických křivek jedná o nulový bod. Z metody výpočtu je zřejmé, že

první součet je součet první a prvního bodu. Tedy se jedná o součet stejného bodu. V další iteraci proběhne součet výsledku a počátečním bodem. Díky návrhu není nutné složitě sestavovat posloupnost použití jednotlivých metod, která v některých případech může vést k chybným výsledkům. Vhodná metoda výpočtu je situačně volena samotným programem. V této části aplikace jsou obsaženy části kódu, které představují programovou implementaci vzorců pro výpočet lambda a souřadnic X a Y. V průběhu výpočtu jsou dle potřeby volány matematické funkce, které jsou popsány v podkapitole (==).
Je nutné zmínit, že do aplikace vstupuje i proměnná, která sebou nese velikost grupy. Prakticky se jedná o počet opakování, které by vedly k výpočtu všech bodů grupě. Pokud však číslo bude menší, výsledkem bude jiný bod z grupy. Tímto způsobem na základě náhodně zvoleného čísla, který představuje tajný klíč. Vypočtený bod bude naopak představovat veřejný klíč a odešle se protistraně.

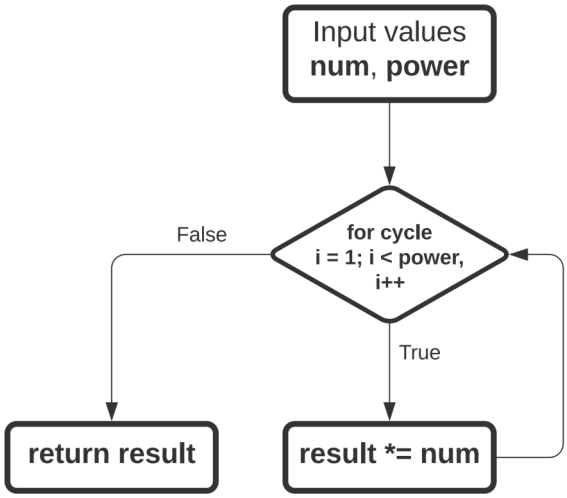

Fig. 5.5: Power function [11]