V klidu a bezpečí (4)

Tématem tohoto dílu bude výklad rodiny Hammingových kódů. Zaměříme se přitom zejména na jejich binární podobu, u které si podrobně vyložíme způsoby detekce a opravy chyb. Ukážeme si též ilustrativní příklad hardwarové realizace popsaného kódu.

Většina publikací zabývajících se problematikou implementace bezpečnostních kódů vychází ve svém výkladu většinou ze zajeté šablony, kdy je nejdříve věnován krátký prostor užitým algebraickým strukturám a poté jsou vysvětlovány jednotlivé rodiny ECC, a to už bez větší návaznosti na obecné principy, vyložené v úvodní pasáži. Je pravda, že takovýto postup má výhodu v tom, že čtenáře vede nejkratší cestou přímo k návrhu příslušného kodéru a dekodéru a nezabývá se přitom "zbytečnými" detaily. Jak jsem už říkal úvodem, má tento seriál sloužit hlavně jako implementační příručka. To znamená, že ani zde se nebudeme zabývat přílišnými detaily. Na druhou stranu by bylo ale jistě škoda vynechat takové drobnosti, které sice samy o sobě nejsou nutné pro praktickou realizaci konkrétního typu ECC, avšak jejichž pochopení umožní výrazně kvalitnější objasnění všech souvislostí v celém systému, a to za cenu jen nepatrně vyšších nároků na čtenářovu pozornost.

Hammingovy kódy, kterým se budeme za okamžik věnovat, jsou ideální k tomu, abychom si na nich názorně demonstrovali přechod od obecné teorie lineárních kódů, kterým byl věnován minulý díl tohoto seriálu, k návrhu konkrétní rodiny kódu s požadovanými vlastnostmi. Zapomeňme proto nyní na okamžik na to, že Hammingovy kódy už existují. Představme si, že neznáme nic jiného než obecnou teorii ECC a že pomocí těchto znalostí chceme navrhnout lineární kód, který bude schopen opravit jednonásobné chyby (bude typu SEC). Takto vybaveni se nyní vydejme na stejnou cestu, na jakou se roku 1950 vydal Dr. Hamming, a stejně jako on tehdy očekávejme s napětím, k jakým výsledkům nakonec dospějeme.

O minimální váze a matici H

Představme si, že máme obecný lineární kód φ typu (n,k) s kontrolní maticí H typu [n-k,n]. Podle tvrzení T3.4 víme, že výpočet minimální kódové vzdálenosti $d_{min}(\varphi)$ můžeme převést na hledání minima váhy přes množinu všech kódových slov. Pokusme se nyní toto užitečné tvrzení ještě rozšířit s cílem najít a dokázat nějakou souvislost mezi minimální kódovou vzdáleností a vlastnostmi matice H.

Pro tento účel se ještě jednou podívejme, jak vypadá proces dekódování přijatého slova x. Minule jsme si řekli, že přijatá slova klasifikujeme jako kódová či nekódová na základě jejich syndromu, který počítáme jako $s = Hx^T$. Tvrzení T3.5 nám říká, že slovo x je kódové právě tehdy, když je jeho syndrom s nulový. Popišme si nyní, co vlastně znamená požadavek na nulový syndrom z pohledu matice H, a zabývejme se přitom pouze rozborem použité operace násobení vektoru maticí. Snadno nahlédneme, že sloupcový vektor s vlastně představuje lineární kombinaci vektorů sloupců matice H. Příslušné koeficienty této kombinace jsou přitom představovány přímo jednotlivými souřadnicemi vektoru x. Zapíšeme-li toto pozorování formálně, potom platí, že $s^T = (s_1, s_2, ..., s_{n-k})$, kde $s_i = \sum_{j=1}^{n} h_{ij} * x_j$.

To, co jsme právě získali, je vztah mezi váhou kódového slova a lineární závislostí vektorů tvořících sloupce matice H. Tento vztah nám říká, že pokud existuje nenulové kódové slovo x o váze w(x), potom v příslušné matici H existuje w(x) lineárně závislých sloupců – *tvrzení T4.1.* Uvedená závislost je pochopitelně netriviální.

Důkaz tohoto tvrzení je snadný a vychází z toho, že je-li slovo x kódové, potom má nulový syndrom, což můžeme zapsat ve tvaru $\sum_{j=1}^{n} h_{ij} * x_j = (0,0,...,0)$. Z této lineární kombinace sloupců matice H můžeme dále vynechat všechny vektory, které mají příslušné koeficienty x_j nulové; tím nám zbude rovnice, která říká, že nějakých w(x) sloupců z matice H je lineárně závislých. Vzhledem k tomu dále, že uvažujeme pouze nenulová slova x, víme, že tato závislost je netriviální – *důkaz P4.1.*

Právě uvedené tvrzení, které nás upozorňuje na jistou souvislost mezi váhou kódového slova a lineární závislostí sloupců v matici H, nás spolu s T3.4 přímo vybízí k formulaci následujícího stěžejního tvrzení: Lineární kód φ s kontrolní maticí H má minimální kódovou vzdálenost $d_{min}(\varphi) = d$ právě tehdy, když d představuje nejmenší celé číslo, pro které v matici H existuje d lineárně závislých sloupců – *tvrzení T4.2.*

Důkaz tohoto tvrzení je poměrně snadný, avšak poněkud delší, takže si jej zde předvádět nebudeme. Kdo chce, může si jej zkusit jako drobné cvičení. Pro nás je teď důležité, že jeho pomocí můžeme v našem případě, kdy hledáme kód opravující jednonásobné chyby ($d_{min}(\varphi) = 3$),

formulovat ihned následující pomocné tvrzení: lineární kód j s kontrolní maticí H má minimální kódovou vzdálenost $d_{min}(\varphi) = 3$ právě tehdy, když libovolná dvojice sloupců z matice H je lineárně nezávislá a současně v H existuje nějaká trojice lineárně závislých sloupců – *tvrzení T4.3.*

Konstrukce matice H

Z předchozího výkladu víme, že pro náš kód hledáme takovou kontrolní matici H typu [n-k,n], jejíž parametry odpovídají tvrzení T4.3. Víme dále, že sloupce této matice jsou tvořeny n-k rozměrnými aritmetickými vektory, kterých je celkem n. Odtud již dostáváme přímo návod na sestrojení matice H výběrem vhodných n-k rozměrných vektorů.

Vektory pro sloupce matice H budeme vybírat z vektorového prostoru V(r,q). Pro přehlednější zápis jsme si zavedli proměnnou r = (n-k), kterou si označíme jako řád hledaného kódu – *definice D4.1*. Podotýkám, že zavedení vektorového prostoru V(r,q) pro sloupce H není samo-účelnou snahou o zesložitění celého výkladu. Vzhledem k tomu, že se tu bavíme o jejich lineární závislosti a nezávislosti, nám už bohužel nestačí chápat je jako pouhé q-nární posloupnosti délky r.

Zaměřme se nyní na postup, jakým z V(r,q) vybereme potřebných n po dvou nezávislých vektorů. První, co víme, je, že v žádném kroku nesmíme vybrat nulový vektor – ten je totiž s libovolným jiným vektorem lineárně závislý. Kromě této podmínky máme při výběru prvního sloupce již naprosto "volné" ruce. Vybereme tudíž libovolný nenulový vektor $v_1 \in V(r,q)$. V dalším kroku jsme už omezeni – můžeme vybrat pouze takový vektor $v_2 \in V(r,q)$, který není násobkem v_1 (jinak by byly v_1 a v_2 lineárně závislé, což nechceme). Při výběru v_3 si pak musíme stejným způsobem dávat pozor na to, aby nebyl násobkem ani jednoho z vektorů v_1 a v_2. Budeme-li tímto způsobem postupovat až do konce, potom máme jistotu, že libovolná dvojice z námi vybraných n vektorů je lineárně nezávislá, takže je můžeme použít jako sloupce matice H. Poznamenejme, že se dá snadno dokázat, že na takto vybrané množině vektorů existuje taková trojice, která lineárně závislá je – tím jsme splnili i druhou část podmínky v T4.3.

Podle tvrzení T4.3 jsme právě obdrželi kontrolní matici H kódu, který můžeme použít na opravu jednonásobných chyb. Zatím však nejsme se vším úplně hotovi; ještě nám zbývá určit, jaké má tento kód vlastně parametry. Víme o něm sice, že je typu (n,k), ale konkrétní hodnoty těchto proměnných zatím neznáme. Snadno se můžeme přesvědčit, že parametry námi navrhovaného kódu nemohou být zcela libovolné. Pokud bychom například zvolili příliš velkou hodnotu n, která mj. také určuje počet sloupců matice H, mohlo by se nám snadno stát, že nebudeme z prvků prostoru V(r,q) schopni vybrat potřebný počet po dvou nezávislých vektorů.

Schopnost výběru příslušných vektorů pro sloupce H je v této chvíli prakticky jediným omezením, které musíme respektovat. Pokusme se proto vyjádřit hodnoty (n,k) pomocí řádu r. Z obecné teorie lineárních kódů víme, že počet informačních bitů k můžeme vyjádřit jako k = n-r. Námi zavedený řád r totiž reprezentuje počet kontrolních bitů kódových slov. Zbývá nám ještě určit hodnotu n. Tu vyjádříme jako maximální počet po dvou nezávislých vektorů, které jsme schopni vybrat z prostoru V(r,q), následujícím vzorcem: $n = (q^r-1)/(q-1)$. Odvození uvedeného vzorce je v podstatě "jen" malým procvičením kombinatoriky, takže se jím zde nebudeme hlouběji zabývat.

Hammingův kód

Právě jsme ukázali, jakým způsobem můžeme sestrojit konkrétní lineární kód, který je schopen opravovat jednonásobné chyby a jehož typ (n,k) je závislý na volitelném parametru, který jsme označili jako řád daného kódu. Obdobným způsobem, jaký jsme si tu dnes ukázali, postupoval (možná s malinko odlišným matematickým aparátem) před rovnými padesáti lety i Dr. Hamming. Podle něho se celá rodina lineárních kódů, které jsou konstruovány právě popsaným způsobem, označuje jako takzvané Hammingovy kódy.

Ještě než se pustíme do další části výkladu, uvedeme si několik základních vlastností Hammingových kódů, jejichž návrh jsme si právě popsali. Začneme třeba hned jejich definicí: q-nární Hammingův kód řádu r je lineární kód typu (n,k), kde $n = (q^r-1)/(q-1)$ a k = n-r, s kontrolní maticí H typu [r,n], jejíž sloupce tvoří po dvou nezávislé vektory z prostoru V(r,q). Minimální kódová vzdálenost všech Hammingových kódů je rovna třem – *definice D4.2*.

Pokud bychom někdy zapomněli postup, jakým jsme Hammingovy kódy odvodili, postačí nám pamatovat si jejich definici – z té bychom měli být schopni daný kód celkem snadno vytvořit.

Poznamenejme, že ačkoliv uvedený popis konstrukce a vlastní definice Hammingových kódů počítá s libovolnou abecedou kódových slov, v praxi se nejčastěji setkáme s binárními (q = 2) zástupci těchto kódů. Tím se nám zjednoduší výrazy pro n a k následovně: $n = 2^r-1$, $k = n-r = 2^r-1-r$.

Na obrázku vidíme příklad binárního Hammingova kódu řádu r = 3, který je podle uvedených

vztahů typu (7,4). Generující matice tohoto kódu byla podle tvrzení T3.6 vypočtena z matice H, která byla nejdříve upravena na tvar $H = (-B^T | E_{n-k})$. Zde stojí za zmínku fakt, že po celou dobu našich úvah nad konstrukcí Hammingova kódu jsme ani jednou nepoužili matici G – místo toho jsme se opřeli pouze o matici H. Tento fakt je možné brát jako ukázku toho, že matice G a H poskytují do určité míry nezávislý pohled na definici hledaného kódu, a je jen otázkou konkrétní situace, který pohled se nám hodí víc. Jak uvidíme dále, hraje v případě Hammingových kódů matice H prim.

Další základní, avšak zajímavou vlastností Hammingových kódů je, že jsou perfektní ve smyslu tvrzení T2.4 – *tvrzení T4.4.* Důkaz této vlastnosti je možné provést jednoduchým dosazením do uvedené nerovnice a ověřením, že pro dvojice čísel (n,k) odvozených od libovolného řádu r podle D4.2 přechází tato nerovnice v rovnici, což znamená, že každý Hammingův kód je perfektní neboli má nejmenší možnou nadbytečnost.

Detekce a oprava chyb

Vzhledem k tomu, že každý Hammingův kód je především kódem lineárním, platí pro detekci a opravu všechna obecná pravidla, která jsme uvedli v minulém díle. Vzhledem k jistým specifickým vlastnostem Hammingových kódů můžeme tato obecná pravidla navíc upravit do takové podoby, ve které jsou v praxi snáze realizovatelná.

Pokud jde o způsob detekce chyb, zde se nic nezměnilo – nejjednodušší a nejosvědčenější metodou zůstává i nadále indikace chyby na základě nenulového syndromu přijatého slova. Hlavní cíl nasazení Hammingových kódů však bude patrně v aplikacích, které budou provádět nejen detekci chyb, ale které budou tyto chyby rovnou i opravovat. Proto nás budou zajímat hlavně postupy pro opravu chyb.

Poněkud těžkopádný způsob opravy chyb, který jsme si uvedli minule, můžeme v případě Hammingových kódů modifikovat do přijatelnější podoby, a to díky tomu, že se zde zajímáme pouze o opravu jednonásobných chyb.

Úvaha, kterou použijeme pro modifikaci obecné metody, vychází opět ze studia chování operace pro výpočet syndromu, kterou jsme použili už během samotného návrhu Hammingových kódů. Předpokládejme, že jsme vyslali kódové slovo c a místo něho jsme přijali nějaké slovo x = c+e, které je zatíženo chybovým vektorem e. Víme, že hodnota syndromu potom odpovídá přímo chybovému vektoru, neboli $s = He^T$. Jak jsme si dnes ukázali, operace typu $H\alpha^T$ vytvářejí lineární kombinace sloupců matice H. Těchto kombinací se přitom "aktivně účastní" tolik sloupců, jaká je váha vektoru α. Dále víme, že jsme schopni opravovat pouze jednonásobné chyby, což znamená, že $w(e) \leq 1$. To znamená, že do zmíněné lineární kombinace nevstupuje buď žádný sloupec matice H (v takovém případě jsme přijali kódové slovo), nebo pouze jeden tento sloupec. To znamená, že pokud jsme přijali slovo zatížené chybou, potom bude jeho syndrom přímo odpovídat sloupci, jehož umístění v matici H udává přímo pozici chybného znaku v přijatém slově.

Právě popsané pozorování je možné využívat mnoha různými způsoby. Buďto se spokojíme s už beztak příjemným faktem, že syndromy přímo odpovídají sloupcům matice H na příslušných pozicích, anebo se budeme snažit z tohoto faktu získat maximum. Jako příklad si můžeme uvést třeba binární Hammingův kód s kontrolní maticí uvedenou na obrázku HW implementace tohoto kódu. Zde jsme provedli takovou permutaci sloupců, že každý sloupec zároveň představuje binární zápis své vlastní pozice v matici H (vektor (0,0,1) je na první pozici, (1,0,1) na páté atd.). Díky této úpravě nyní nenulový syndrom přijatého slova přímo určuje binární zápis pozice, na které k chybě došlo. Poznamenejme, že tato permutace byla provedena na úkor toho, že matice H již není ve tvaru $(-B^T | E_{n-k})$, což nám ale v tomto případě nevadí.

HW realizace

Začneme například konstrukcí kodéru neboli obvodovou realizací příslušného zobrazení φ. Zde můžeme buďto vyjít z příslušné generující matice G, anebo stačí využít toho, co víme o matici H. Jak jsme si uvedli minule, reprezentuje matice H koeficienty soustavu homogenních rovnic, jejichž řešení představuje podprostor všech kódových slov. Kódování vysílaných vektorů je proto možné provádět i tak, že každému odeslanému vektoru přiřadíme vždy jedno konkrétní řešení uvedené soustavy rovnic. Hodnost této soustavy je přitom n-k, což přesně odpovídá naší situaci, kdy si k proměnných volíme (ty položíme přímo rovny vstupnímu vektoru) a n-k proměnných potom vypočteme na základě předepsaných rovnic.

Celý postup bude srozumitelnější, jestliže si příslušné rovnice dané maticí H vypíšeme tak, jak je uvedeno na obrázku. Zde vidíme, že nejsnazším postupem pro výpočet příslušného kódového

slova je nejprve podle vstupního slova (z) stanovit výstupní hodnoty na pozicích (3,5,6,7) jako: $x_3 = z_1$, $x_5 = z_2$, $x_6 = z_3$ a $x_7 = z_4$ a podle uvedených rovnic potom dopočítat pozice (1,2,4) jako: $x_1 = x_3 + x_5 + x_7$, $x_2 = x_3 + x_6 + x_7$, $x_4 = x_5 + x_6 + x_7$.

Prakticky je celý postup kódování uveden na obrázku. Poznamenejme, že logické obvody označené znakem $\oplus$ značí logické členy XOR, jejichž použití vychází z algebraických vlastností tělesa Z_2.

Při dekódování přijatého slova je třeba nejprve určit jeho syndrom. Pro tento účel se použije hardwarová realizace rovnic, které vzniknou rozepsáním operace Hx^T. Abychom docílili přehlednosti a jednoduchosti našeho schématu, zahrnuli jsme výpočet syndromu do samostatného bloku označeného jako SYND. Vzhledem k tomu, jak jsme si uspořádali matici H, dostáváme na výstupu obvodu SYND buď nulu (v takovém případě jsme přijali kódové slovo), anebo zde obdržíme přímo číselnou pozici místa, kde došlo k chybě. Vzhledem k tomu, že pracujeme nad Z_2, provedeme opravu této chyby jednoduše opět pomocí “naxorování” jedničkového bitu na příslušnou pozici. Tuto pozici snadno určíme pomocí dekodéru jedna z n, který aplikujeme na výstup obvodu SYND.

Věnujme se nyní ponaučením, které nám měl tento příklad poskytnout. Díky tomu, že jsme se striktně nedrželi podmínky na tvar matice H, mohli jsme provést takovou permutaci jejích sloupců, která nám umožnila poměrně snadnou realizaci obvodu dekodéru. Provedená permutace však měla i své stinné stránky: podle T3.6 jsme nemohli elegantně určit matici G a dále výsledný kód nebyl souvisle systematický. Nakonec se však ukázalo, že ani jedna z těchto věcí nám nevadila, neboť bez použití matice G jsme se obešli zcela, a pokud jde o souvislou systematičnost, právě jsme si ukázali, že pro HW realizaci, kde není problém provádět libovolné permutace přenášených slov, bohatě postačuje podmínka na systematičnost dle D2.1.

Závěr

Dnes jsme se podrobně podívali na nejznámější zástupce lineárních kódů, a to na Hammingovu rodinu ECC. Na příkladu těchto kódů jsme si dále rozšířili naše obecné znalosti lineárních kódů a ukázali jsme si, jak se tyto vědomosti v praxi aplikují při návrhu konkrétních druhů kódování. Dále jsme si zde ukázali hardwarovou realizaci binárního Hammingova kódu, kde jsme upozornili na jeho specifické vlastnosti, které je možné využít pro jeho efektivní implementaci.

V příštím díle nás čeká popis Golayových kódů, což je další nejznámější rodina lineárních kódů.

Tomáš Rosa (tomas.rosa@decros.cz)

Literatura:

[ROMA92] Roman, S.: Coding and Information Theory, Springer-Verlag, 1992.