УДК 621.391

АНАЛІЗ МЕТОДІВ ПОБУДОВИ КОДІВ АВТЕНТИФІКАЦІЇ ПОВІДОМЛЕНЬ

О.В. Сєвєрінов[1], О.Ю. Іохов[2], О.С. Жученко[1], В.П. Лисечко[1]
([1]Харківський університет Повітряних Сил ім. І. Кожедуба,
[2]Науковий метрологічний центр (військових еталонів))

У статті проводиться аналіз методів побудови кодів автентифікації повідомлень (MAC кодів) для забезпечення стійкості до нав'язування хибної інформації у системах управління.

автентифікація повідомлень, MAC коди

Постановка проблеми. Досвід локальних конфліктів останніх двох десятиріч та проведений аналіз якісних характеристик сучасних систем управління і зв'язку свідчить, що їх ефективне застосування залежить у першу чергу від безпеки інформації, яка циркулює в них. При чому одною з основних задач безпеки даних є забезпечення їх цілісності та істинності. Це пояснюється тим, що при успішній реалізації погроз, зловмисник може порушити управління за рахунок нав'язування неправдивої інформації, порушуючи її цілісність та істинність. Здатність системи протистояти введенню до неї хибної інформації, несанкціонованому доступу до інформації, що передається або приймається, нав'язуванню хибних режимів роботи називається автентичністю.

У теперішній час відомо багато різних методів та алгоритмів забезпечення автентичності інформації, які відрізняються швидкістю обчислень, рівнем захищеності від загальних атак та довжинами кодів. Таким чином, необхідне проведення аналізу відомих методів побудови кодів автентифікації повідомлень, щоб вибрати найбільш прийнятний для систем управління.

Аналіз літератури. Результати досліджень провідних спеціалістів у галузі захисту інформації, а також аналіз літератури [1, 2] показали, що одним з найбільш перспективних методів автентифікації є введення в інформацію надмірності (імітовставки) на основі використання кодів автентифікації повідомлень (MAC кодів), що дозволяє з заданою імовірністю встановлювати дійсність переданого повідомлення. Можна відокремити три загальні підходи до побудови MAC кодів [2]. MAC коди, побудовані із застосуванням блокових шифрів (CBC-MAC), MAC коди, побудовані на основі безключових хеш-функцій (HMAC, MDX-MAC), MAC коди, побудовані з використанням сімейства універсальних хеш-функцій.

Мета статті. Проведення аналізу методів побудови кодів автенфикації повідомлень для забезпечення стійкості до нав'язування хибної інформації у системах управління.

Аналіз методів побудови автенфикації повідомлень на основі застосування блокових шифрів та безключових хеш-функцій. Проведений аналіз основних методів та алгоритмів формування MAC кодів показав, що існуючий міжнародний стандарт ISO/IEC 9797-1 визначає алгоритми для MAC кодів, що використовують блоковий шифр в режимі зчеплення за шифр текстом (CBC MAC). Вітчизняний стандарт ГОСТ 28147-89 використовує алгоритм CBC MAC у четвертому режимі для генерації 64-бітної імітовставки.

Для отримання імітовставки відкриті дані M розбиваються на блоки M(j) довжиною 64 біти. Перший блок M(1) піддається перетворенню, яке відповідає першим 16 циклам алгоритму шифрування в режимі простої заміни, причому в якості ключа виробки імітовставки використовується ключ шифрування даних. Отримане 64-розрядне число складається за модулем 2 з другим блоком відкритих даних M(2). Результат підсумовування також піддається перетворенню. Останній блок M(n), при необхідності доповнений до 64-розрядного числа нулями, складається за модулем 2 з результатом роботи на (n–1)-му кроці і зашифровується. З отриманого 64-розрядного числа виділяється відрізок I_p довжиною p бітів, який є хеш-функцією відкритих даних (імітовставкою). Значення параметра p визначається виходячи з необхідної імовірності обману.

При розшифруванні аналогічно формується імітовставка I'_p, яка порівнюється з імітовставкою I_p, що міститься у прийнятому повідомленні. У випадку розбіжності отримане повідомлення вважається хибним.

По оцінкам провідних спеціалістів практична реалізація алгоритму ГОСТ 28147-89 є низькошвидкісною (на сучасній ПЕОМ досягає приблизно 100Кбіт/сек.) та має розмір MAC коду у 64 біт, що також вже є недостатнім, для багатьох практичних задач.

Для вивчення основних характеристик MAC кодів і їх порівняльної оцінки в науково-дослідному проекті NESSIE [3] були розглянуті основні практичні алгоритми MAC кодів і відібрані чотири кращі:

– Two-Track-MAC: K.U.Leuven, Бельгія і Debis AG, Німеччина;

– UMAC: розробка корпорації Intel (США), Університету з штату Невада в Рено (США), Науково-дослідної лабораторії IBM (США), Technion, (Ізраїль) і Університете з Каліфорнії в Девисі (США);

– CBC-MAC (ISO/IEC 9797-1);

– HMAC (ISO/IEC 9797-1).

Основними характеристиками алгоритмів формування MAC кодів, за якими виконується їх порівняльна оцінка відповідно до рекомендацій

проекту NESSIE, є рівень захищеності MAC кодів від загальних атак, швидкодія алгоритмів формування MAC кодів, статистичні властивості розподілів MAC кодів.

У табл. 1 наведені основні результати щодо параметрів і оцінки швидкодії основних алгоритмів імітозахисту для різних операційних платформ [4 – 6]. Швидкість обчислень визначається кількістю циклів процесора, затрачуваних на один байт повідомлення, що обробляється.

Таблиця 1

Швидкодія алгоритмів формування MAC кодів

Алгоритм	Довжина MAC коду (біт)	Довжина ключа (біт)	Тип ПЕОМ			
			PIII/Linux	Pentium 4	Xeon	AMD
Ttmac	160	160	21	40	37	21
Umac-16	64	128	6,0	6,2	6,1	6,2
Umac-32	64	128	2,9	6,7	6,6	1,9
HMAC-Whirlpool	512	512	72	98	103	100
HMAC-MD4	128	512	4,7	6,4	6,4	4,7
HMAC-MD5	128	512	7,3	9,4	9,4	7,4
HMAC-RIPE-MD	160	512	18	27	26	21
HMAC-SHA-1	160	512	15	25	24	12
HMAC-SHA-2	256 384 512	512	39 84 84	40 124 124	39 132 132	33 72 72
HMAC-Tiger	192	512	21	28	26	20
CBCMAC-Rijndael	128	128	26	26	27	31
CBCMAC-DES	64	56	61	72	69	54
CBCMAC-Shacal	512	160	31	67	74	29

Підсумкові результати проекту NESSIE і аналіз табл. 1 дозволяють зробити ряд висновків.

TTMAC (Two-Track-MAC) має найвищий рівень захисту для MAC примітивів, представлених NESSIE і за підсумковими показниками є переважним. Алгоритм працює на блоках 512 біт, розділених на слова по 32 біти, використовує секретний ключ 160 біти, і MAC код довжиною до 160 біт. Великий розмір внутрішнього стану (320 біт) в Two-Track-MAC дає алгоритму високий рівень захисту від атак, заснованих на внутрішніх колізіях.

Якщо ми позначимо довжину MAC коду m (значення для m підтримується алгоритмом між 32 і 160 бітами), то складність основних атак на цей примітив визначається:

– приблизно 2^{159} обчислень MAC коду і 160/m відомих пар текст – MAC необхідні для вичерпного пошуку ключа;

– вгадування значення MAC коду має імовірність успіху 2^{-m};

– атаки, засновані на внутрішніх колізіях, вимагають приблизно 2^{159} відомих пар текст – MAC код і приблизно 2^{320-m} вибраних текстів.

Разом з тим слід зазначити, що TTMAC має низьку швидкість, що робить проблематичним застосування для додатків, де потрібно хешувати дані великих об'ємів.

Самим високошвидкісним алгоритмом формування MAC кодів є UMAC алгоритм, відомий в модифікаціях UMAC(1999) і UMAC(2000) [6]. UMAC алгоритм представлений проектом NESSIE, як програмно-орієнтований для реалізації на сучасних операційних платформах, і забезпечує надзвичайно високу швидкість обчислень. Розробники UMAC переслідували дві головні цілі: високу швидкість обчислень та доказову секретність.

Рішення цих задач виявилося можливим на основі застосування композиційної схеми з багатократним універсальним хешуванням і криптографічним обчисленням MAC коду. Алгоритм UMAC був розроблений так, щоб забезпечити паралельні обчислення в SIMD архітектурі. SIMD архітектура забезпечується регістрами, які, в деяких інструкціях, можуть поводитися зі словами малого розміру, як з векторами. Одна із найшвидших реалізацій UMAC використовує MMX інструкції Pentium, які поводяться з 64 бітовим регістром, як з чотиривимірним вектором по 16 біт.

Залежно від установки початкових параметрів алгоритму UMAC вивчено п'ять схем: UMAC-STD-30, UMAC-STD-60, UMAC-MMX-15, UMAC-MMX-30 і UMAC-MMX-60.

Результати випробувань схем UMAC з оцінкою швидкості обчислень представлені в табл. 2 [6, 7]. З аналізу табл. 2 та робіт [6, 7] можна зробити наступні висновки. Максимальний коефіцієнт стиснення досягається на повідомленнях 4 Кбайт. UMAC виконується краще всього на довгих повідомленнях тому, що хеш-функція є тоді більш ефективною через скорочення кількості обчислень, що доводяться на псевдовипадкову функцію PRF. В якості PRF функції застосовується одна з криптографічних хеш-функцій у режимі CBC-MAC або HMAC. Деякий виграш у швидкості з'являється вже при довжинах повідомлення в пару сотень байт.

Таблиця 2

Швидкість обчислення UMAC, яка вимірюється в Гбіт/с (циклу/байт)

Алгоритм	Pentium II	PowerPC	Альфа
UMAC-STD-60	1,49 (1,93)	1,81 (1,58)	1,03 (2,78)
UMAC-STD-30	2,79 (1,03)	2,28 (1,26)	1,79 (1,60)
UMAC-MMX-60	2,94 (0,98)	4,21 (0,66)	0,287 (10,0)
UMAC-MMX-30	5,66 (0,51)	7,20 (0,39)	0,571 (5,02)
UMAC-MMX-15	8,47 (0,33)	10,5 (0,27)	0,981 (2,85)
CBC-MAC-RC6	0,162 (17,7)	0,210 (13,7)	0,068 (42,5)
HMAC-SHA1	0,227 (12,6)	0,228 (12,6)	0,117 (24,5)

Підвищення швидкості обчислень, особливо при повідомленнях малої довжини, було досягнуте в UMAC (2000), де запропоновано дві схеми: UMAC32 (без SIMD паралелізму) і UMAC16 (з SIMD паралелізмом), що дозволило досягти продуктивності в три рази більшої, ніж при перших версіях UMAC-STD і UMAC-MMX (див. табл. 1). Висока швидкість забезпечується за рахунок того, що MAC код обчислюється за схемою: результат хешування $\oplus$ PRF(показник новизни), який посилається одержувачу разом з повідомленням і показником новизни.

UMAC16, аналогічно і UMAC32, використовує три класи хеш-функцій: NH, RPHash і IPHASH.

NH хешує блоки по 2Кбайт, проводячи хеш-код в 32 біти, що відповідає коефіцієнту стиснення 512. Ймовірність колізії не перевищує 2^{-15}. Результат передається до RP рівня хешування, який обчислює вихідний рядок довжиною 128 біт. RP родина використовує три прості поля із 32 бітовим, із 64 бітовим та із 128 бітовим простими модулями, відповідно.

Довжина повідомлення обмежена максимальним значенням 2^{64} біт, і доведено, що цей рівень додає близько 2^{-19} до ймовірності колізії. Якщо імітозахищене повідомлення коротке, RP шар пропускається для оптимізації швидкості обчислення. IP рівень згортає вхідну послідовність з 128 бітами до вихідного із 16 бітами, підтримуючи імовірність колізії майже 2^{-15}. Конструкція з трьома рівнями повторюється неодноразово із незалежними ключами, збільшуючи довжину MAC коду і зменшуючи шанс підробки MAC коду. Задане за умовчанням число – чотири рази, і конкатенація 16 бітних слів обчислює MAC код з 64 бітами імовірністю підробки 2^{-60}. Головна відмінність у UHASH32 полягає в тому, що використовуються слова з 32 бітами і обчислення повторюються за схемою з трьома шарами тільки двічі (значення за умовчанням).

Таблиця 3

Продуктивність алгоритмів UMAC для різних довжин хешуємих даних (цикли/байт)

Алгоритм	43 байт	256 байт	1500 байт	256 Кбайт
UMAC32	16,3	3,8	2,1	1,9
UMAC-STD	52,9	12,3	3,8	1,9
UMAC16	14,0	2,7	1,2	1,0
UMAC-MMX	35,9	4,5	1,7	1,0

Відзначимо ряд недоліків у даному алгоритмі. NH хешування на початковому кроці UMAC алгоритму припускає генерацію 1024 4-х байтових підключів. У принципі це є типовим для багатьох сучасних шифрів і для тривалих сеансів не має істотного впливу на продуктивність алгоритму. Проте, враховуючи короткочасність численних сеансів в сучасних

АСУ військами, це є проблемою, оскільки приводить до зниження продуктивності алгоритму.

Другий рівень RPHASH хешування використовує для обчислень прості поля із 32 бітовим, 64 бітовим та 128 бітовим простими модулями. Це визначається необхідністю переходу до хешування даних все зростаючої довжини. Більшість спецобчислювачів в АСУ військами, які стоять на озброєнні, орієнтовані на 32-х розрядні обчислення. Згідно UMAC алгоритму 32-х розрядні обчислення підтримують хешування 2^{17} бітних даних, що є недостатнім для передачі файлових і мультимедійних додатків в АСУ військами.

Аналіз методів побудови автенфикації повідомлень на основі використанням сімейства універсальних хеш-функцій. Як показує проведений аналіз [7], самі високошвидкісні і криптографічно стійкі MAC коди засновані за допомогою універсальних хеш-функцій. Перевагою універсальних хеш-функцій є те, що вони забезпечують доказову імовірність колізії і мають прозорі комбінаторні властивості (дозволяють аналізувати різні результати при оцінці ймовірності колізії). До таких схем відносяться універсальні класи хеш-функцій на основі поліноміального хешування; схеми універсального хешування на основі довгих АГ кодів; композиційні схеми універсального хешування.

Практичні MAC алгоритми є багаторівневими схемами, адаптованими до швидкісного хешування даних різної довжини. Для забезпечення стійкості алгоритмів формування MAC кодів використовують перевірені криптографічні примітиви, крипостійкість яких доведена або вселяє довіру у спеціалістів.

Аналіз практичних алгоритмів формування кодів автентифікації повідомлень показав, що вони можуть включати класи хеш-функцій з великим коефіцієнтом стиснення для даних дуже великого об'єму, але при цьому зберігаються колізійні властивості [4 – 6]. Тому інтерес представляють схеми універсального хешування на основі довгих алгеброгеометричних кодів. Для відомих до теперішнього часу алгеброгеометричних кодів імовірність колізії обмежується в кращому випадку значенням обернено пропорційним квадрату розмірності поля Z_q. Застосування алгеброгеометричних кодів для побудови універсальних класів хеш-функцій забезпечує найкращі співвідношення між розміром хеш-кодів і ймовірністю колізії при обчисленнях в кінцевих полях, а також мінімізує витрати за ключовими даними [8]. Аналіз загальних характеристик алгеброгеометричних кодів показав , що найкращими властивостями з точки зору їх кодової відстані та складності обчислення є розширений код Ріда-Соломона, коди на кривих Ерміта та коди на кривих Сузукі [9].

Однак використання універсального хешування на основі довгих алгеброгеометричних кодів можливе лише при застосуванні композиційних схеми універсального хешування, які ефективно знижує ймовірність колізії. Разом з тим в полі обмеженої розмірності найбільш ефективним у відношенні між складністю обчислення та ймовірністю колізії, на відміну від композиційних схем, є застосування каскадної схеми хешування [8].

Висновки. Проведений аналіз методів побудови кодів автенфикації повідомлень показав, що використання відомих методів не дозволяє забезпечити необхідний рівень стійкості до вводу хибної інформації при малої складності алгоритму і високої швидкості обчислень. Виконання цих умов можливе при використанні у системах управління каскадної схеми хешування на основі довгих алгеброгеометричних кодів.

ЛІТЕРАТУРА

1. *Пятибратов А.П. Вычислительные системы, сети и телекоммуникации: Учеб. для вузов. – М., 2003. – 512 с.*
2. *International Organization for Standardization, ISO/IEC 9797-2: Information Technology – Security Techniques - Message Authentication Codes (MACs) – Part 2: "Mechanisms using a dedicated hash-function", 2002. – P. 152-153*
3. *New European Schemes for Signatures, Integrity and Encryption. – NESSIE Project.: http://cryptonessie.org.*
4. *Performance of Optimized Implementations of the NESSIE Primitives // NESSIE Performance Evaluation. – 2003. – V 2.0, Feb 20. – P. 52-78.*
5. *Black, J., Halevi, S., Krawczyk, H., Krovetz, T. and Rogaway, P. UMAC: Fast and secure message authentication. In Advances in Cryptology // CRYPTO '99. – 1999. – Vol. 1666 of Lecture Notes in Computer Science, Springer-Verlag. – P. 216-233.*
6. *T. Krovetz, J. Black, S. Halevi, A. Hevia, H. Krawczyk, and P. Rogaway. «UMAC». Primitive submitted to NESSIE, Sept. – 2000. – P. 5-157.*
7. *Stinson D. Universal hashing and authentication codes // Design, Codes and Cryptography. – 1994. – Vol. 4. – P. 369-380.*
8. *Халимов Г.З., Иохов А.Ю. Каскадное универсальное хеширование с использованием АГК кодов // Восточно-европейский журнал передовых технологий. –2005. – Вып. 2/2 (14). – С. 111-119.*
9. *Halevi, S. and Krawczyk, H. MMH: Software message authentication in the Gbit/second rates. In Proceedings of the 4th Workshop on Fast Software Encryption // Springer-Verlag. – 1997. – Vol. 1267. – P. 172-189.*

Надійшла 30.03.2006

Рецензент: доктор технічних наук, професор Ю.В. Стасєв,
Харківський університет Повітряних Сил ім. І. Кожедуба.