Agenzia per la
Cybersicurezza Nazionale

LINEE GUIDA FUNZIONI CRITTOGRAFICHE

Codici di Autenticazione di Messaggi (MAC)

MAGGIO 2026

Versione	Data di pubblicazione	Note
1.0	**07/12/2023**	**Prima pubblicazione**
1.1	**12/05/2026**	**Aggiunta sezione “Scopo del documento”, ulteriori modifiche minori**

Sommario

pag.

Indice delle figure

pag.

Indice delle tabelle

Scopo del documento

Questo documento costituisce parte della serie "Linee Guida Funzioni Crittografiche" e fornisce le raccomandazioni di ACN in merito ai **codici di autenticazione di messaggi**, da adottare in tutti i contesti in cui si necessita di garantire integrità e autenticazione di dati inviati o salvati. Per ulteriori informazioni sulle Linee Guida e sulle utilità delle funzioni crittografiche in base ai contesti specifici, si faccia riferimento al documento introduttivo della serie [1].

Ogni documento tiene in considerazione le minacce presenti al giorno della sua pubblicazione. Data la diversa natura dei sistemi informativi di destinazione, non è possibile garantire che queste raccomandazioni possano essere utilizzate senza adattamenti specifici. In qualsiasi caso, la pertinenza dell'attuazione delle soluzioni proposte deve essere sottoposta, preventivamente, a valutazione e validazione da parte dei responsabili della sicurezza dei sistemi informativi di destinazione.

I contenuti delle Linee Guida sono indirizzati a sviluppatori, produttori di dispositivi e fornitori di servizi digitali, al fine di promuovere l'utilizzo di primitive crittografiche e relativi parametri di configurazione sicuri fin dalla fase di progettazione di prodotti, reti, applicazioni e servizi. Questi documenti sono altresì rivolti a security manager, referenti e responsabili della cybersicurezza di soggetti pubblici e privati affinché verifichino che i sistemi utilizzati dalla propria organizzazione siano conformi alle raccomandazioni fornite.

La legge 28 giugno 2024, n. 90 relativa a "Disposizioni in materia di rafforzamento della cybersicurezza nazionale e di reati informatici", all'articolo 9, stabilisce a tal fine che *«le strutture di cui all'articolo 8 della presente legge nonché quelle che svolgono analoghe funzioni per i soggetti di cui all'articolo 1, comma 2-bis, del decreto-legge 21 settembre 2019, n. 105, convertito, con modificazioni, dalla legge 18 novembre 2019, n. 133, e al decreto legislativo 18 maggio 2018, n. 65, verificano che i programmi e le applicazioni informatiche e di comunicazione elettronica in uso, che utilizzano soluzioni crittografiche, rispettino le linee guida sulla crittografia nonché quelle sulla conservazione delle password adottate dall'Agenzia per la cybersicurezza nazionale e dal Garante per la protezione dei dati personali e non comportino vulnerabilità note, atte a rendere disponibili e intellegibili a terzi i dati cifrati»*.

Il documento è stato curato dal Centro Nazionale di Crittografia istituito presso ACN.

Lista dei simboli matematici utilizzati

$\{0,1\}$	Campo binario dei valori assumibili da un singolo bit
$\{0,1\}^n$	Spazio vettoriale delle stringhe binarie di lunghezza n
$\{0,1\}^*$	Insieme di stringhe binarie di lunghezza arbitraria
$\|\|$	Concatenazione di stringhe
$\oplus$	Operazione XOR, cioè somma bit a bit tra stringhe binarie

1 Introduzione

Nell'ambito delle comunicazioni elettroniche si può richiedere l'autenticazione di un messaggio, anche senza che questo venga cifrato, quando il destinatario della comunicazione vuole poter verificare che il messaggio sia stato inviato da un mittente specifico, con cui condivide la conoscenza di un dato segreto.
In crittografia, lo strumento necessario a garantire l'autenticazione di un messaggio viene chiamato codice di autenticazione del messaggio o MAC (Message Authentication Code). Tale codice consiste in una stringa da associare al messaggio stesso che può essere generata in modo corretto solo da chi possiede la chiave segreta. In particolare, i MAC sono in grado di garantire, oltre all'autenticazione del mittente, anche l'integrità del messaggio. Gli algoritmi per la generazione di MAC devono soddisfare alcune proprietà di sicurezza, al fine di garantire che nessuno sia in grado di creare un MAC valido senza essere in possesso della chiave.
Questo documento fornisce indicazioni sui migliori algoritmi per la generazione di MAC in termini di sicurezza, dei quali si raccomanda l'utilizzo. Si farà ampio riferimento a funzioni di hash, cifrari a blocchi e firme digitali, per le cui raccomandazioni e nozioni tecniche si rimanda ai documenti dedicati [2, 3, 4].
Il documento presenta la seguente struttura: nel capitolo 2 si presentano gli schemi di autenticazione del messaggio, evidenziando le differenze con altre primitive crittografiche e alcuni possibili attacchi a cui devono essere resistenti; nel capitolo 3 vengono introdotte le specifiche degli algoritmi per generare MAC raccomandati; infine, nel capitolo 4 si richiamano brevemente le raccomandazioni sugli algoritmi e i parametri da utilizzare.

2 Schemi di autenticazione del messaggio

Un **MAC**, anche detto **tag di autenticazione**, viene generato tramite uno schema crittografico simmetrico. Questo algoritmo prevede che, data una chiave segreta K di lunghezza k, la stringa in input M venga affiancata da un valore $\mathrm{MAC}_K(M)$ lungo n bit, chiamato MAC di M. In formule, fissata la chiave K,

$$\mathrm{MAC}_K : \{0,1\}^* \longrightarrow \{0,1\}^n.$$

I MAC sono stati sviluppati per rispondere alla possibile esigenza di dover **autenticare** un messaggio, ossia di voler garantire che, una volta ricevuto il messaggio, il destinatario possa essere sicuro dell'identità del mittente. Questa proprietà viene assicurata dall'utilizzo della chiave segreta condivisa: il mittente, usando la chiave, genera il tag del messaggio e lo invia congiuntamente al messaggio; il destinatario, partendo dal messaggio e dalla chiave comune, genera un altro tag e lo confronta con quello ricevuto, accettando il messaggio solamente se i due tag coincidono. Inoltre, il processo di creazione del tag assicura l'**integrità** del messaggio, permettendo al ricevente di verificare se il dato trasmesso è stato modificato.
L'utilizzo dei MAC è necessario in diversi contesti applicativi. Un esempio pratico può essere la memorizzazione delle password, le quali vengono salvate negli archivi solo dopo l'applicazione ripetuta di MAC. Un altro caso molto frequente è lo scambio di messaggi autenticati: il messaggio cifrato si invia unitamente al tag dello stesso, permettendo al ricevente di verificare la sua integrità e l'identità del mittente, mentre la confidenzialità del dato scambiato viene assicurata dalla cifratura.

2.1 Differenze con altre primitive crittografiche

Come osservato precedentemente, i MAC garantiscono integrità e autenticazione di un dato input. Se per diversi aspetti appaiono simili alle funzioni di hash e alle firme digitali, essi assolvono a una funzione nettamente diversa da queste altre primitive crittografiche.
Infatti, le funzioni di hash assicurano l'integrità ma non l'autenticazione dei dati. Questo perché tutti coloro che sono in possesso del messaggio possono calcolarne il digest, mentre per ottenere un MAC è presupposta la conoscenza di una chiave simmetrica.
Rispetto alle firme, invece, manca la proprietà di non-ripudio. Infatti, chiunque sia in grado di verificare un MAC, può anche generarne uno, essendo uno schema simmetrico che prevede il possesso della stessa chiave, sia da parte del mittente che del destinatario.

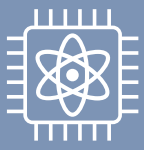
2.2 Attacchi ai MAC

La principale tipologia di attacco di cui i MAC possono essere oggetto è quella della **contraffazione**, cioè la generazione di un tag valido per un messaggio, senza essere in possesso della chiave segreta. Questo genere di attacco può essere effettuato in due diversi contesti: qualora l'attaccante possieda alcuni messaggi casuali con i rispettivi tag, l'attacco viene definito **known-message**. Se, invece,

l'attaccante ha la possibilità di scegliere alcuni messaggi specifici e riesce a recuperare i tag corrispondenti, allora l'attacco viene detto **chosen-message**. Il primo caso è il contesto più comune e quindi si richiede un'alta resistenza in tali circostanze; il secondo scenario permette, generalmente, di avere un attacco più efficace, sebbene raramente si verifichi nella pratica. In ogni caso, per garantire la sicurezza di un MAC, si richiede che lo schema sia resistente in entrambe le situazioni.

2.2.1 Attacco di forza bruta

Il modo più semplice per cercare di contraffare un MAC è quello di utilizzare un attacco di **forza bruta**: l'attaccante semplicemente genera un tag casuale e prova ad autenticare il messaggio con esso. La probabilità di indovinare il tag corretto è $1/2^n$, quindi è sufficiente adeguare la lunghezza per ottenere una probabilità abbastanza bassa e, se necessario, impostare un massimo numero di verifiche effettuabili per una data chiave. Le raccomandazioni sulla dimensione da scegliere per il tag si possono trovare nel capitolo 4.



Minaccia quantistica

Come la maggior parte della crittografia simmetrica, i MAC risultano suscettibili agli attacchi perpetrati da un computer quantistico tramite l'**algoritmo di Grover** [5], che garantisce, tuttavia, solo un aumento quadratico della velocità degli attacchi di forza bruta. Quindi, un raddoppio delle dimensioni della chiave permette di garantire lo stesso livello di sicurezza degli standard attuali, di fatto rendendo vani i miglioramenti quantistici.

2.2.2 Attacco di estensione della lunghezza

L'integrità di un messaggio M può essere assicurata tramite una funzione di hash h. Si potrebbe allora pensare di ottenere l'autenticazione, e quindi un MAC, ponendo il vettore di inizializzazione IV di h (di solito fissato ad un certo valore precisato nelle specifiche dell'algoritmo) uguale alla chiave K e utilizzare come tag di M il digest $h(M)$ così ottenuto.
Tuttavia, se la funzione h sfrutta la costruzione di Merkle-Damgård, questo metodo è suscettibile ad un semplice attacco di contraffazione, chiamato **attacco di estensione della lunghezza** [6]. Essendo h pubblica, l'attaccante conosce anche la funzione di compressione f utilizzata nelle singole iterazioni della funzione di hash, ovvero

$$f : \{0,1\}^{n+k} \longrightarrow \{0,1\}^n,$$

dove n è la lunghezza dello stato e $M = M_0 \,\|\, \ldots \,\|\, M_\ell$ viene diviso in blocchi di lunghezza k, ognuno dei quali viene elaborato secondo la regola

$$h(M_0) = f(K, M_0),$$
$$h(M_i) = f\big(h(M_{i-1}), M_i\big).$$

Con questa costruzione, se un attaccante entra in possesso di un qualsiasi messaggio M e del rispettivo tag $h(M)$, può facilmente ottenere un tag valido per qualsiasi estensione di M con un blocco M' di lunghezza k calcolando

$$h(M \| M') = f(h(M), M').$$

Il membro a sinistra dell'equazione è il MAC del messaggio $M \,\|\, M'$, che l'attaccante può facilmente calcolare perché possiede tutti i dati utilizzati nel membro a destra. Il metodo descritto funziona qualora la funzione di hash non esegua una fase di elaborazione preliminare o una trasformazione dell'output. Ciononostante, l'attacco può essere facilmente adattato per compromettere funzioni di hash che effettuano queste operazioni aggiuntive [6].

3 Algoritmi raccomandati

Gli schemi crittografici che generano MAC sono basati principalmente su funzioni di hash o su cifrari a blocchi. Di seguito, verranno descritti i tre schemi raccomandati per generare MAC.

3.1 HMAC

Un **HMAC** (Hash-based Message Autentication Code) [7] è un tipo specifico di MAC generato tramite uno schema che utilizza una funzione di hash crittografica e di una chiave crittografica segreta.
Questo schema è stato ideato nel 1996 da Mihir Bellare, Ran Canetti e Hugo Krawczyk [8] ed è stato reso uno standard per l'autenticazione dei messaggi dal NIST nel 2008 [9].

Lo schema di generazione di un HMAC dipende dalla funzione di hash h che si sceglie di adoperare. Data una chiave crittografica K, che deve essere nota solo al mittente e al destinatario, lo schema opera sul messaggio M da autenticare ottenendo il tag come

$$\mathrm{HMAC}_K(M) = h\Big(\big(K \oplus opad\big) \parallel h\big((K \oplus ipad) \parallel M\big)\Big),$$

dove $opad$ e $ipad$ sono stringhe binarie costanti della dimensione di un blocco della funzione di hash, chiamate rispettivamente padding esterno e padding interno. Si evidenzia che, nel caso in cui la dimensione della chiave K sia maggiore di quella di un blocco, prima di calcolare il MAC del messaggio, si deve ridurre la lunghezza di K tramite l'applicazione della funzione di hash h.

Per quanto riguarda la scelta della funzione di hash da utilizzare, si rimanda al documento dedicato [2].
Al momento non esistono attacchi a questo schema che siano migliori del classico attacco di forza bruta, sebbene la scelta della funzione di hash influisca notevolmente sulla sicurezza contro le contraffazioni. Va segnalato, ad esempio, che esistono dei cosiddetti "distinguisher" in grado di identificare gli HMAC generati con la funzione di hash MD5, la quale può essere sfruttata per attaccare la generazione dei tag [10, 11].

3.2 CMAC

L'algoritmo per generare un **CMAC** (Cipher-based Message Authentication Code) [12] utilizza un cifrario a blocchi nella modalità Cipher Block Chaining (CBC) [13]. L'origine di questo schema risale al precedente CBC-MAC [14], che è stato dimostrato essere suscettibile ad attacchi di estensione della lunghezza, motivo per cui Iwata e Kurosawa hanno proposto una variazione chiamata One-Key CBC-MAC o OMAC [15, 16], che ha assunto successivamente l'attuale denominazione di CMAC. Lo schema è stato poi standardizzato dal NIST nel 2005 [17].
In questo algoritmo viene utilizzato un cifrario a blocchi E con una chiave segreta K per calcolare il tag T di un messaggio M.
Come descritto in Figura 1, l'input M viene suddiviso in blocchi, ottenendo $M = M_0 \,\|\, \ldots \,\|\, M_\ell$, i quali vengono cifrati con E in modalità CBC usando sempre la chiave K. Tale modalità prevede che, indicando con C_i il cifrato dell'i-esimo blocco, si abbia

$$C_0 = E_K(M_0), \quad C_i = E_K(C_{i-1} \oplus M_i), \quad 0 < i < \ell.$$

L'ultimo blocco M_ℓ viene prima modificato utilizzando una di due chiavi parziali K_1 o K_2. Queste sono il risultato di una funzione di derivazione (composta da trasformazioni elementari) applicata alla cifratura di un blocco nullo con la chiave K. Il nuovo blocco M'_ℓ viene infine cifrato sempre con la chiave K sfruttando la modalità CBC, ottenendo così il tag di M come

$$T = E_K(C_{\ell-1} \oplus M'_\ell).$$

Eventualmente, il tag T viene adattato alla dimensione richiesta scartando i bit meno significativi.

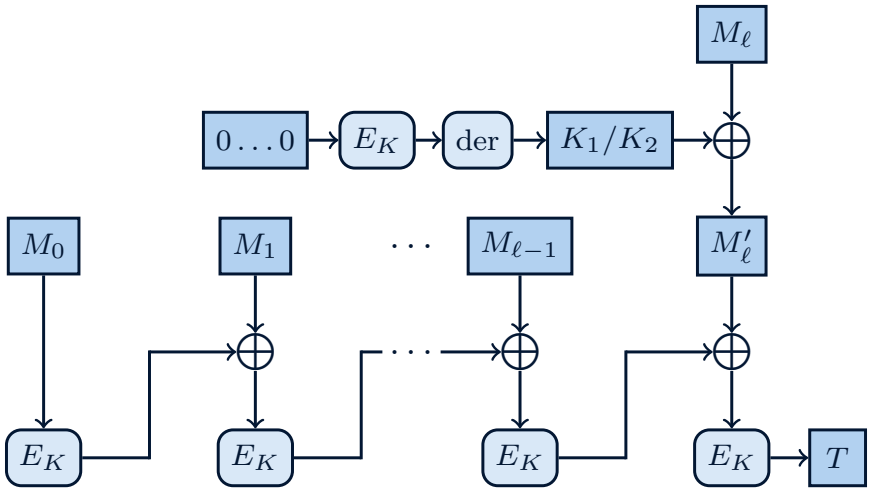


Figura 1 - Algoritmo per generare un tag con CMAC

Per quanto riguarda la scelta del cifrario a blocchi da utilizzare nella generazione di un CMAC, si rimanda al documento dedicato [3]. Al momento, per ognuno dei cifrari consigliati, la sicurezza dell'algoritmo non risulta essere intaccata da attacchi migliori del semplice attacco di forza bruta.

3.3 GMAC

Un codice di autenticazione **GMAC** (Galois Message Authentication Code) [18] viene generato utilizzando, come nel caso precedente, un cifrario a blocchi, ma la modalità adottata è la Galois/Counter Mode (GCM) [19], progettata nel 2004 da David McGrew e John Viega [20]. Il NIST ha standardizzato questa modalità e l'algoritmo di generazione di GMAC nel 2007 [21].

Per la costruzione di un GMAC, è sufficiente ricordare che, oltre al testo in chiaro P, un cifrario GCM prende in input dei dati di autenticazione addizionali (AAD) e un vettore di inizializzazione (IV), che deve essere cambiato ad ogni utilizzo. L'output consiste nel testo cifrato C di P, che ne assicura la confidenzialità, e un tag T relativo sia a P sia agli AAD, che ne garantisce l'autenticazione. Lo schema di generazione di un GMAC dipende dal cifrario E che si sceglie di adoperare e l'idea alla base è quella di sfruttare la GCM con il messaggio M come AAD e un testo in chiaro P vuoto. Il funzionamento dettagliato è rappresentato in Figura 2:

- l'inizializzazione prevede la cifratura tramite E di una stringa di b zeri con la chiave K, dove b è la lunghezza dei blocchi processati da E. Il risultato H è il parametro utilizzato dalla funzione di round f_H, la quale moltiplica l'input per H e riduce il risultato modulo 2^b;
- l'input viene suddiviso in blocchi da b bit ottenendo $M = M_0 \,\|\, \ldots \,\|\, M_\ell$ a seguito di eventuale padding;
- il primo blocco viene processato calcolandone f_H;
- ad ogni round successivo, un blocco di M viene processato calcolandone lo XOR con l'output della precedente applicazione di f_H;
- una volta processati tutti i blocchi, viene calcolato lo XOR tra il risultato dell'ultima applicazione di f_H e un blocco di b bit costituito dalla lunghezza di M seguita da zeri;
- infine, avviene uno XOR finale con la cifratura di un blocco di b bit contenente il IV seguito da zeri e un 1, che restituisce il tag del messaggio M.

Eventualmente, il tag T viene adattato alla dimensione richiesta scartando i bit meno significativi.

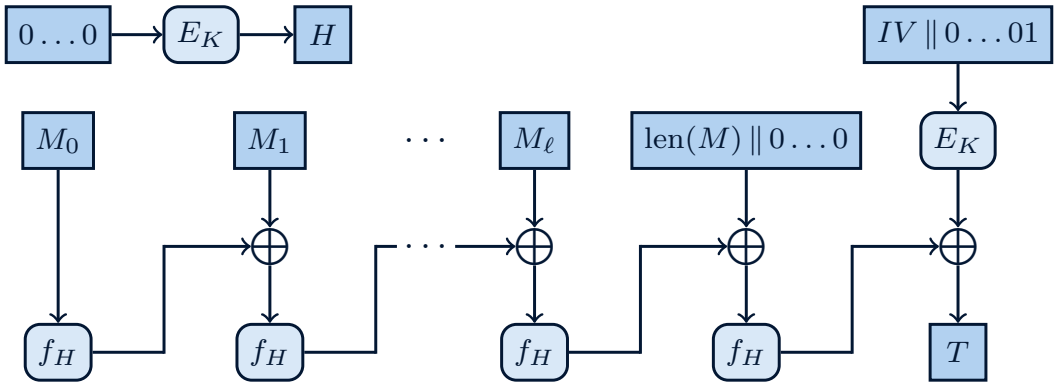


Figura 2 - Algoritmo per generare un tag con GMAC

Per quanto riguarda la scelta del cifrario a blocchi da utilizzare nella generazione di un GMAC, si rimanda al documento dedicato [3]. Al momento, per ognuno dei cifrari consigliati, la sicurezza dell'algoritmo non risulta essere intaccata da attacchi migliori del semplice attacco di forza bruta.

4 Conclusioni

In base a quanto descritto in questo documento, le raccomandazioni sugli algoritmi per generare MAC sono riassunte in Tabella 1.
Come in ogni algoritmo di crittografia simmetrica, la conservazione e la gestione della chiave simmetrica devono necessariamente essere effettuate in maniera adeguata, utilizzando le precauzioni previste nel documento dedicato*.
Al fine di raggiungere un adeguato livello di sicurezza, in ognuno degli schemi presentati si consiglia una chiave segreta di almeno 128 bit e una troncatura dello stato finale, e quindi una dimensione del tag di 96 bit come riassunto nella Tabella 1. In casi particolari ed eccezionali (come la verifica di MAC già generati, i contesti con limitata memoria dei dispositivi, ...), può essere ammessa una dimensione minore di 96 bit, ma mai inferiore a 64 bit.
Per quanto riguarda HMAC, è importante utilizzare una funzione di hash che garantisca un livello di sicurezza adeguato. Per un elenco delle funzioni raccomandate, si rimanda al documento dedicato [2].
Allo stesso modo, riguardo CMAC e GMAC, si raccomanda l'utilizzo di algoritmi di cifratura simmetrica che garantiscano sufficiente sicurezza. Al momento, l'unico cifrario a blocchi completamente supportato dalla comunità scientifica è AES [22], si raccomanda, perciò, l'utilizzo di CMAC e GMAC con solo questo cifrario. Per maggiori informazioni riguardo i cifrari a blocchi, si rimanda al documento dedicato [3].

Algoritmo	Lunghezza della chiave (bit)	Lunghezza del tag (bit)
HMAC	≥128	≥96
CMAC	≥128	≥96
GMAC	≥128	≥96

Tabella 1 - Algoritmi raccomandati per generare MAC

*In fase di pubblicazione.

Bibliografia

[1] ACN. *Introduzione alla Crittografia e alle Linee Guida*. Linee Guida Funzioni Crittografiche, 2026. URL: https://www.acn.gov.it/portale/crittografia.

[2] ACN. *Funzioni di Hash*. Linee Guida Funzioni Crittografiche, 2026. URL: https://www.acn.gov.it/portale/crittografia.

[3] ACN. *Cifrari a Blocchi e Modalità di Funzionamento*. Linee Guida Funzioni Crittografiche, 2026. URL: https://www.acn.gov.it/portale/crittografia.

[4] ACN. *Firme Digitali*. Linee Guida Funzioni Crittografiche, 2026. URL: https://www.acn.gov.it/portale/crittografia.

[5] L. K. Grover. «A fast quantum mechanical algorithm for database search». In: *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC)*. 1996, pp. 212–219. DOI: 10.1145/237814.237866.

[6] D. R. Stinson e M. Paterson. *Cryptography: Theory and Practice (4th edition)*. Chapman e Hall/CRC, 2017. DOI: 10.1201/9781315282497.

[7] ISO. *Information security – Message authentication codes (MACs) – Part 2: Mechanisms using a dedicated hash-function*. ISO/IEC 9797-2. 2021. URL: https://www.iso.org/standard/75296.html.

[8] M. Bellare, R. Canetti e H. Krawczyk. «Keying hash functions for message authentication». In: *Advances in Cryptology - CRYPTO '96*. Lecture Notes in Computer Science. 1996, pp. 1–15. DOI: 10.1007/3-540-68697-5_1.

[9] NIST. *The Keyed-Hash Message Authentication Code (HMAC)*. FIPS 198-1. U.S. Department of Commerce, 2008. DOI: 10.6028/NIST.FIPS.198-1.

[10] X. Wang et al. «Cryptanalysis on HMAC/NMAC-MD5 and MD5-MAC». In: *Advances in Cryptology - EUROCRYPT 2009*. Lecture Notes in Computer Science. 2009, pp. 121–133. DOI: 10.1007/978-3-642-01001-9_7.

[11] T. Peyrin, Y. Sasaki e L. Wang. «Generic related-key attacks for HMAC». In: *Advances in Cryptology - ASIACRYPT 2012*. Lecture Notes in Computer Science. 2012, pp. 580–597. DOI: 10.1007/978-3-642-34961-4_35.

[12] International Organization for Standardization. *ISO/IEC 9797-1:2011: Information technology - Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher*. 2022.

[13] ISO. *Information technology – Security techniques – Modes of operation for an n-bit block cipher*. ISO/IEC 10116. 2017. URL: https://www.iso.org/standard/64575.html.

[14] A. J. Menezes, P. C. V. Oorschot e S. A. Vanstone. *Handbook of applied cryptography (1st edition)*. CRC Press, 1997. DOI: 10.1201/9780429466335.

[15] T. Iwata e K. Kurosawa. «OMAC: One-key CBC MAC». In: *Fast Software Encryption (FSE)*. Lecture Notes in Computer Science. 2003, pp. 129–153. DOI: 10.1007/978-3-540-39887-5_11.

[16] T. Iwata e K. Kurosawa. *OMAC: One-key CBC MAC - Addendum*. NIST submission. 2003. URL: http://www.nuee.nagoya-u.ac.jp/labs/tiwata/omac/docs/omac-ad.pdf.

[17] M. Dworkin. *Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication*. SP 800-38B. NIST, 2005. DOI: 10.6028/NIST.SP.800-38B.

[18] ISO. *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 3: Mechanisms using a universal hash-function*. ISO/IEC 9797-3. 2011. URL: https://www.iso.org/standard/51619.html.

[19] ISO. *Information security – Authenticated encryption*. ISO/IEC 19772. 2020. URL: https://www.iso.org/standard/81550.html.

[20] D. A. McGrew e J. Viega. *The Galois/Counter Mode of Operation (GCM)*. NIST submission. 2005. URL: https://csrc.nist.rip/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf.

[21] M. Dworkin. *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. SP 800-38D. NIST, 2007. DOI: 10.6028/NIST.SP.800-38D.

[22] NIST. *Advanced Encryption Standard (AES)*. FIPS 197. U.S. Department of Commerce, 2023. DOI: 10.6028/NIST.FIPS.197-upd1.