Radboud University

Keying Merkle-Damgård at the Suffix

Bart Mennink

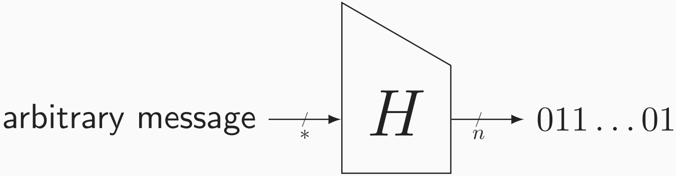
Radboud University

FSE 2025

March 19, 2025

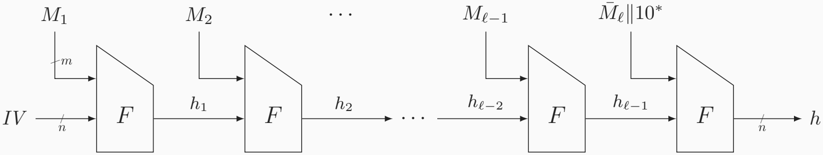
Introduction

Cryptographic Hashing



- Function H from $\{0,1\}^*$ to $\{0,1\}^n$
  - Variable-length input
  - Classically fixed length output (but could be variable as well)

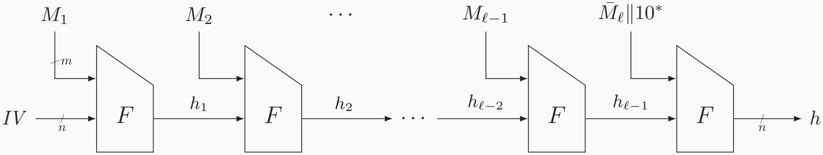
Merkle-Damgård [Dam89, Mer89]



Merkle-Damgård with Strengthening

- Uses compression function F from $n + m$ to n bits
- State initialized using IV
- Message M injectively padded and cut into m-bit blocks
- Consecutive evaluation of compression function F

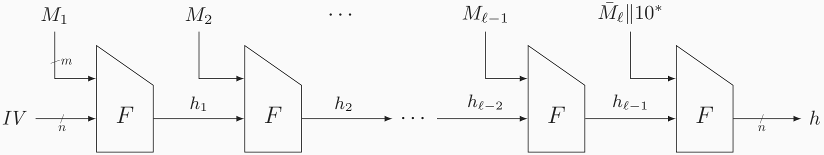
Merkle-Damgård [Dam89, Mer89]



Merkle-Damgård with Strengthening

- Uses compression function F from $n + m$ to n bits
- State initialized using IV
- Message M injectively padded and cut into m-bit blocks
- Consecutive evaluation of compression function F
- Used, among others, in SHA-1/2 [Nat15]

Merkle-Damgård [Dam89, Mer89]

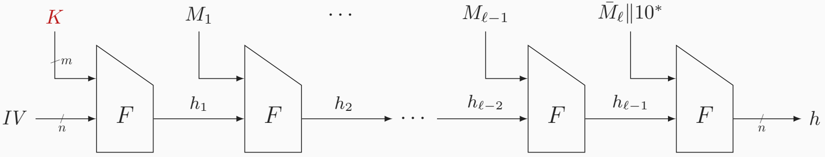


Merkle-Damgård with Strengthening

- Uses compression function F from $n + m$ to n bits
- State initialized using IV
- Message M injectively padded and cut into m-bit blocks
- Consecutive evaluation of compression function F
- Used, among others, in SHA-1/2 [Nat15]

What if we want to do message authentication?

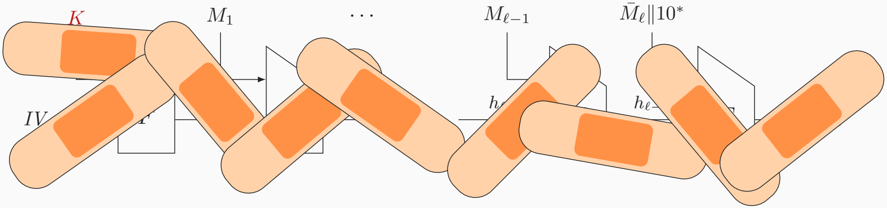
Keying Merkle-Damgård – Length Extension Attack



Keying at the Prefix

- Vulnerable to the length extension attack [Tsu92, KR95]
  - Query tag $h \leftarrow H(K \| M)$
  - Compute $h' \leftarrow F(h, X \| 10^*)$ as forgery for $M \| 10^* \| X$

Keying Merkle-Damgård – Length Extension Attack



Keying at the Prefix

- Vulnerable to the length extension attack [Tsu92, KR95]
  - Query tag $h \leftarrow H(K \| M)$
  - Compute $h' \leftarrow F(h, X \| 10^*)$ as forgery for $M \| 10^* \| X$
- We need a band-aid

Keying Merkle-Damgård – Alternative Options

Enveloped Merkle-Damgård: $H(K\|M\|K)$ **[Tsu92]**

Suffix Keyed Merkle-Damgård: $H(M\|K)$ **[Tsu92]**

Keying Merkle-Damgård – Alternative Options

Enveloped Merkle-Damgård: $H(K\|M\|K)$ [Tsu92]

- Evolved into $\mathrm{HMAC}(K, M) = H\big(K_{out}\|H(K_{in}\|M)\big)$ [BCK96]
- HMAC: secure if F is a pseudorandom function (PRF) [BCK96, Bel06]

Suffix Keyed Merkle-Damgård: $H(M\|K)$ [Tsu92]

Keying Merkle-Damgård – Alternative Options

Enveloped Merkle-Damgård: $H(K\|M\|K)$ [Tsu92]

- Evolved into HMAC$(K,M) = H\big(K_{out}\|H(K_{in}\|M)\big)$ [BCK96]
- HMAC: secure if F is a pseudorandom function (PRF) [BCK96, Bel06]
- Similar result applies to enveloped Merkle-Damgård [Yas07]

Suffix Keyed Merkle-Damgård: $H(M\|K)$ [Tsu92]

Keying Merkle-Damgård – Alternative Options

Enveloped Merkle-Damgård: $H(K\|M\|K)$ [Tsu92]

- Evolved into HMAC$(K, M) = H\big(K_{out}\|H(K_{in}\|M)\big)$ [BCK96]
- HMAC: secure if F is a pseudorandom function (PRF) [BCK96, Bel06]
- Similar result applies to enveloped Merkle-Damgård [Yas07]

Suffix Keyed Merkle-Damgård: $H(M\|K)$ [Tsu92]

- Vulnerable to offline collision attack in $2^{n/2}$ evaluations of F [PvO95]

Keying Merkle-Damgård – Alternative Options

Enveloped Merkle-Damgård: $H(K\|M\|K)$ [Tsu92]

- Evolved into HMAC$(K, M) = H\big(K_{out}\|H(K_{in}\|M)\big)$ [BCK96]
- HMAC: secure if F is a pseudorandom function (PRF) [BCK96, Bel06]
- Similar result applies to enveloped Merkle-Damgård [Yas07]

Suffix Keyed Merkle-Damgård: $H(M\|K)$ [Tsu92]

- Vulnerable to offline collision attack in $2^{n/2}$ evaluations of F [PvO95]
- Not much analysis since

Intermezzo: Modern Approach

HMAC: Bad Solution to a Bad Problem

- Novel approach:
  - Take H that is indifferentiable from random oracle [MRH04]
  - Sponge [BDPV07], Merkle-Damgård with permutation [HPY07], . . .

Intermezzo: Modern Approach

HMAC: Bad Solution to a Bad Problem

- Novel approach:
  - Take H that is indifferentiable from random oracle [MRH04]
  - Sponge [BDPV07], Merkle-Damgård with permutation [HPY07], ...
- Generic security of keyed constructions follows by composition

Intermezzo: Modern Approach

HMAC: Bad Solution to a Bad Problem

- Novel approach:
  - Take H that is indifferentiable from random oracle [MRH04]
  - Sponge [BDPV07], Merkle-Damgård with permutation [HPY07], ...
- Generic security of keyed constructions follows by composition

Example: Sponge [BDPV07]

- Sponge$(K\|M)$ works fine (see also KMAC [Joh16])
- Sponge$(K\|M\|K)$ works fine
- Sponge$(M\|K)$ works fine

Intermezzo: Modern Approach

HMAC: Bad Solution to a Bad Problem

- Novel approach:
  - Take H that is indifferentiable from random oracle [MRH04]
  - Sponge [BDPV07], Merkle-Damgård with permutation [HPY07], . . .
- Generic security of keyed constructions follows by composition

Example: Sponge [BDPV07]

- Sponge$(K\|M)$ works fine (see also KMAC [Joh16])
- Sponge$(K\|M\|K)$ works fine
- Sponge$(M\|K)$ works fine $\longleftarrow$ even achieves leakage resilience [DM19]

Completing the Picture for Merkle-Damgård

Prefix Keyed Merkle-Damgård

- Vulnerable to the length extension attack

Enveloped Merkle-Damgård and HMAC

- Both got various proofs [BCK96, Bel06, Yas07]
- All rely on PRF security of F

Completing the Picture for Merkle-Damgård

Prefix Keyed Merkle-Damgård

- Vulnerable to the length extension attack

Enveloped Merkle-Damgård and HMAC

- Both got various proofs [BCK96, Bel06, Yas07]
- All rely on PRF security of F

Suffix Keyed Merkle-Damgård?

Completing the Picture for Merkle-Damgård

Prefix Keyed Merkle-Damgård

- Vulnerable to the length extension attack

Enveloped Merkle-Damgård and HMAC

- Both got various proofs [BCK96, Bel06, Yas07]
- All rely on PRF security of F

Suffix Keyed Merkle-Damgård?

- What security does it actually achieve (black-box, leakage resilience)?

Completing the Picture for Merkle-Damgård

Prefix Keyed Merkle-Damgård

- Vulnerable to the length extension attack

Enveloped Merkle-Damgård and HMAC

- Both got various proofs [BCK96, Bel06, Yas07]
- All rely on PRF security of F

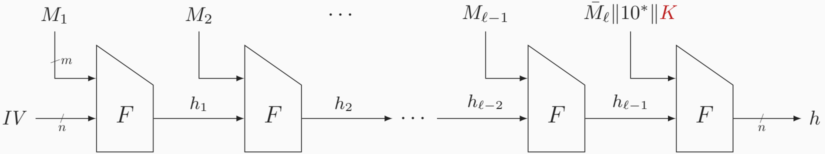
Suffix Keyed Merkle-Damgård?

- What security does it actually achieve (black-box, leakage resilience)?
- Can we prove security without using random oracle model for F?

Suffix Keyed and Suffix Blinded Merkle-Damgård

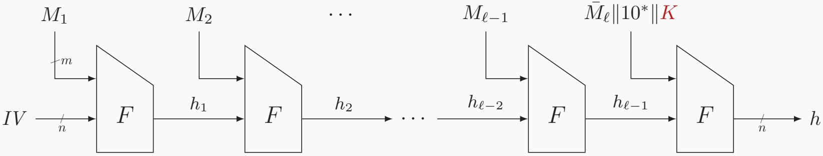
Suffix Keyed and Suffix Blinded Merkle-Damgård

Suffix Keyed Merkle-Damgård (sukMD)

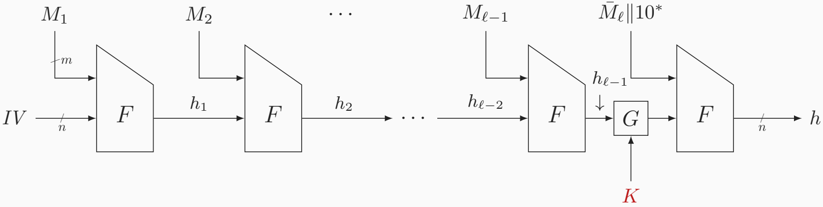


Suffix Keyed and Suffix Blinded Merkle-Damgård

Suffix Keyed Merkle-Damgård (sukMD)

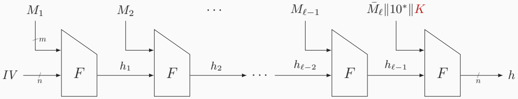


Suffix Blinded Merkle-Damgård (subMD)



Black-Box Security Results (1/2)

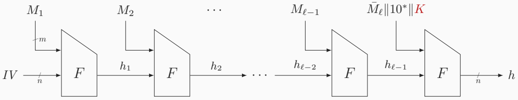
Suffix Keyed Merkle-Damgård (sukMD)



- PRF secure if
  - F is collision resistant
  - F is right-input PRF secure

Black-Box Security Results (1/2)

Suffix Keyed Merkle-Damgård (sukMD)

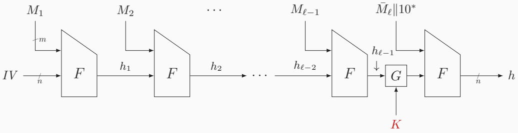


- PRF secure if
  - F is collision resistant
  - F is right-input PRF secure

- PRF attack on sukMD implies either:
  - PRF attack on final F, or
  - a collision in $h_{\ell-1}$ (which can be further reduced to collision in F)

Black-Box Security Results (2/2)

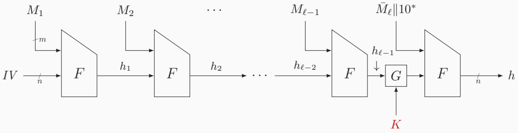
Suffix Blinded Merkle-Damgård (subMD)



- PRF secure if
  - F is collision resistant
  - F is related-key PRF secure (under key relation G)

Black-Box Security Results (2/2)

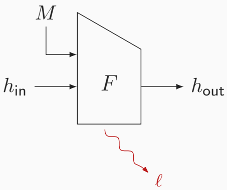
Suffix Blinded Merkle-Damgård (subMD)



- PRF secure if
  - F is collision resistant
  - F is related-key PRF secure (under key relation G)

- Different from previous proof: related-key security of F
- δ-uniform and ε-universal G (e.g., $\oplus$) works

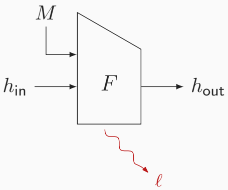
Leakage Resilience



Non-Adaptive Leakage Resilience [DP10]

- Evaluations of F may leak: $L(h_{\text{in}}, M, h_{\text{out}})$

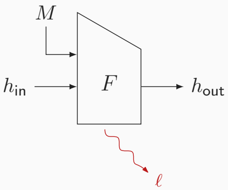
Leakage Resilience



Non-Adaptive Leakage Resilience [DP10]

- Evaluations of F may leak: $L(h_{\text{in}}, M, h_{\text{out}})$
- Adversary may influence the type of function L
  - Strongest possible setting: it may choose L

Leakage Resilience

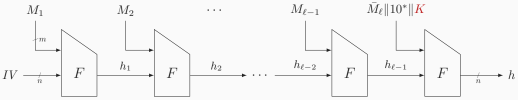


Non-Adaptive Leakage Resilience [DP10]

- Evaluations of F may leak: $L(h_{\text{in}}, M, h_{\text{out}})$
- Adversary may influence the type of function L
    - Strongest possible setting: it may choose L
- We assume that G is strongly protected [DM19]

Leakage Resilience Security Results (1/2)

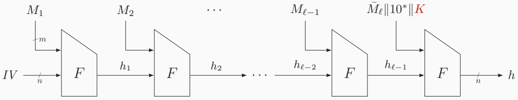
Suffix Keyed Merkle-Damgård (sukMD)



- Insecure under leakage
  - Adversary may vary $h_{\ell-1}$ or $\bar{M}_\ell$ to learn different bits of K
  - Precise attack in paper

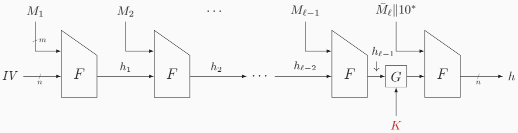
Leakage Resilience Security Results (1/2)

Suffix Keyed Merkle-Damgård (sukMD)



- Insecure under leakage
  - Adversary may vary $h_{\ell-1}$ or $\bar{M}_\ell$ to learn different bits of K
  - Precise attack in paper

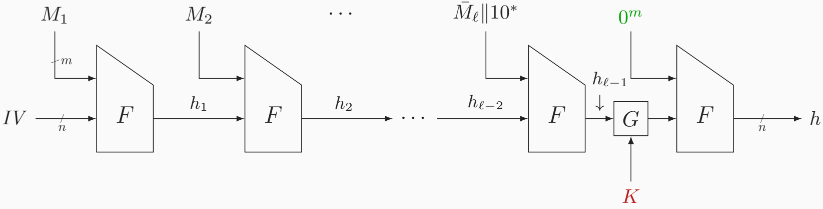
Suffix Blinded Merkle-Damgård (subMD)



- Insecure under leakage
  - Adversary may vary $\bar{M}_\ell$ to learn different bits of $G(K, h_{\ell-1})$
  - Precise attack in paper

Leakage Resilience Security Results (2/2)

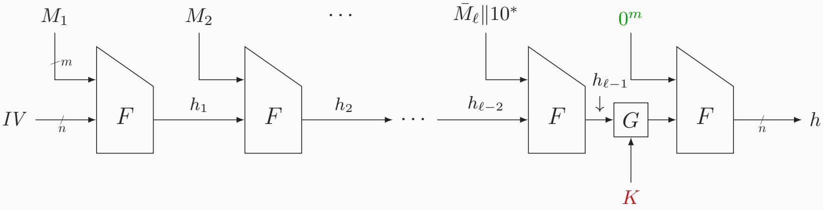
Zero-Padded Suffix Blinded Merkle-Damgård (zsubMD)



- Difference: padding with m zeros 0^m

Leakage Resilience Security Results (2/2)

Zero-Padded Suffix Blinded Merkle-Damgård (zsubMD)



- Difference: padding with m zeros 0^m
- Leakage resilient PRF secure if
  - F is collision resistant
  - F is related-key leakage resilient PRF secure (under key relation G)

Conclusion

Conclusion

In-Depth Analysis of Keying Merkle-Damgård

	black-box	leakage resilient
Suffix keyed Merkle-Damgård	✓	✗
Suffix blinded Merkle-Damgård	✓	with zero-pad

- Results directly extend to Merkle-Damgård with permutation [HPY07]

Conclusion

In-Depth Analysis of Keying Merkle-Damgård

	black-box	leakage resilient
Suffix keyed Merkle-Damgård	✓	✗
Suffix blinded Merkle-Damgård	✓	with zero-pad

- Results directly extend to Merkle-Damgård with permutation [HPY07]

Conditions

- F must be collision resistant and (somehow) PRF secure
- G must be "good enough" $\longleftarrow$ how to instantiate?
- Key must be of size at most $\min\{m, n\}$, otherwise it overflows

Conclusion

In-Depth Analysis of Keying Merkle-Damgård

	black-box	leakage resilient
Suffix keyed Merkle-Damgård	✓	✗
Suffix blinded Merkle-Damgård	✓	with zero-pad

- Results directly extend to Merkle-Damgård with permutation [HPY07]

Conditions

- F must be collision resistant and (somehow) PRF secure
- G must be "good enough" $\longleftarrow$ how to instantiate?
- Key must be of size at most $\min\{m, n\}$, otherwise it overflows

Thank you for your attention!

References i

Mihir Bellare, Ran Canetti, and Hugo Krawczyk.
Keying Hash Functions for Message Authentication.
In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1996.

Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Sponge functions.
Ecrypt Hash Workshop 2007, May 2007.

References ii

- Mihir Bellare.
  New Proofs for NMAC and HMAC: Security without collision-resistance.
  In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*, volume 4117 of *Lecture Notes in Computer Science*, pages 602–619. Springer, 2006.

- Ivan Damgård.
  A Design Principle for Hash Functions.
  In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 1989.

References iii

- Christoph Dobraunig and Bart Mennink.
  Security of the Suffix Keyed Sponge.
  IACR Trans. Symmetric Cryptol., 2019(4):223–248, 2019.

- Yevgeniy Dodis and Krzysztof Pietrzak.
  Leakage-Resilient Pseudorandom Functions and Side-Channel Attacks on Feistel Networks.
  In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2010.

References iv

Shoichi Hirose, Je Hong Park, and Aaram Yun.
A Simple Variant of the Merkle-Damgård Scheme with a Permutation.
In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 113–129. Springer, 2007.

John Kelsey, Shu-jen Chang, Ray Perlner.
NIST Special Publication 800-185: SHA-3 Derived Functions: cSHAKE, KMAC, TupleHash, and ParallelHash, December 2016.

References v

Burt Kaliski and Matt Robshaw.
Message Authentication with MD5.
CryptoBytes, 1(1):5–8, 1995.

Ralph C. Merkle.
One Way Hash Functions and DES.
In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 428–446. Springer, 1989.

References vi

- Ueli M. Maurer, Renato Renner, and Clemens Holenstein.
  Indifferentiability, Impossibility Results on Reductions, and Applications to the Random Oracle Methodology.
  In Moni Naor, editor, *Theory of Cryptography, First Theory of Cryptography Conference, TCC 2004, Cambridge, MA, USA, February 19-21, 2004, Proceedings*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.

- National Institute of Standards and Technology.
  FIPS 180-4: Secure Hash Standard (SHS), August 2015.

- Bart Preneel and Paul C. van Oorschot.
  MDx-MAC and Building Fast MACs from Hash Functions.
  In Don Coppersmith, editor, *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, volume 963 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 1995.

- Gene Tsudik.
  Message authentication with one-way hash functions.
  Comput. Commun. Rev., 22(5):29–38, 1992.

Kan Yasuda.
Boosting Merkle-Damgård Hashing for Message Authentication.
In Kaoru Kurosawa, editor, *Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, volume 4833 of *Lecture Notes in Computer Science*, pages 216–231. Springer, 2007.