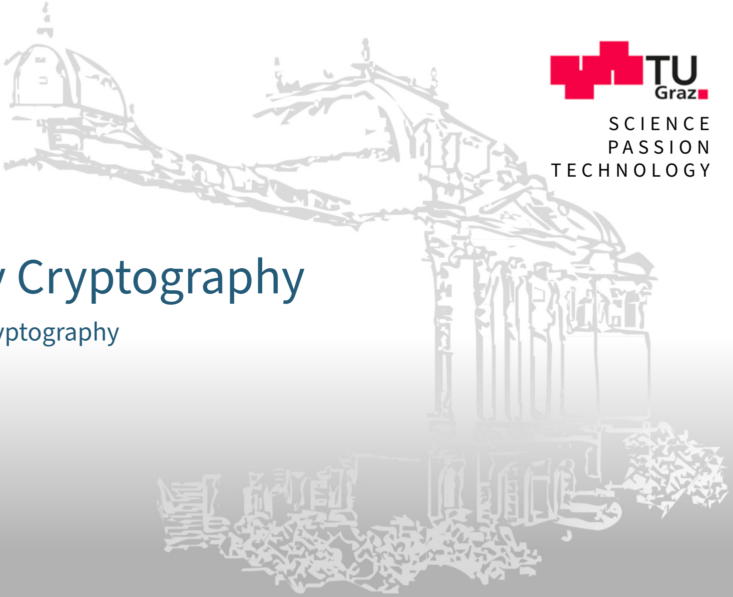
SCIENCE
PASSION
TECHNOLOGY



Modern Public Key Cryptography

Provable Security in Symmetric Cryptography

Daniel Kales

May 25, 2022

Outline

Provable Security in Symmetric Cryptography

Provable Security

- For all previous topics
  - Assumptions of hard problems
  - Reduction of security of scheme to hardness of problem
- Is there something similar in the symmetric world?
  - hard problems $\leftrightarrow$ symmetric primitives
  - cryptographic schemes $\leftrightarrow$ modes of operation
  - reduce the security of higher-level construction to security of primitive!

Provable Security

- For all previous topics
  - Assumptions of hard problems
  - Reduction of security of scheme to hardness of problem
- Is there something similar in the symmetric world?
  - hard problems $\leftrightarrow$ symmetric primitives
  - cryptographic schemes $\leftrightarrow$ modes of operation
  - reduce the security of higher-level construction to security of primitive!

Ideal Primitives

Idea: replace concrete building blocks with ideal variants

- Ideal block cipher (Ideal cipher model)
- Ideal hash function (RO model)

Prove protocol secure if primitive is replaced by ideal variant.

Algorithmic Description of an Ideal Cipher

Ideal Cipher

Internally keep a table T, storing key-plaintext-ciphertext tuples.
On Query (K, P):

1. Check if $(K, P, *)$ already in table T. If yes, return corresponding C.
2. If not, generate a random C.
3. Check if $(K, *, C)$ already in T. If yes, go to (2).
4. Add (K, P, C) to T and return C.

Decryption similar.

Problems with the Ideal Cipher model

Some theoretical problems with RO and IC models

- existence of (contrived) constructions that are secure in the IC model, but are insecure when instantiated with any cipher

Concrete problems:

- Ideal ciphers: select one of $2^n!$ possible permutations at random
- Real ciphers: key selects one of 2^k possible permutations

Often more reasonable to assume other properties of building blocks.

Pseudorandom Generator

Pseudorandom Generator (PRG)

A function $G : \{0,1\}^n \mapsto \{0,1\}^m$, with $m \gg n$ is a pseudorandom generator if:

- G is efficient
- for all PPT adversaries $\mathcal{A}$:

$$\left|\Pr[x \xleftarrow{R} \{0,1\}^n, \mathcal{A}(G(x)) = 1] - \Pr[r \xleftarrow{R} \{0,1\}^m, \mathcal{A}(r) = 1]\right| \leq \epsilon(n).$$

Last bullet: Indistinguishability from truly random output. (Equivalent definition: No PPT algorithm can predict the next output bit with non-negligible advantage given previous output.)

Pseudorandom Function

Pseudorandom Function (PRF)

A function $F : \{0,1\}^k \times \{0,1\}^n \mapsto \{0,1\}^m$ is a pseudorandom function if:

- for any $K \in \{0,1\}^k$, F is efficient
- for all PPT adversaries $\mathcal{A}$:

$$\left|\Pr[K \overset{R}{\leftarrow} \{0,1\}^k, \mathcal{A}^{F_K}(1^n) = 1] - \Pr[f_n \overset{R}{\leftarrow} \mathcal{F}_{n,m}, \mathcal{A}^{f_n}(1^n) = 1]\right| \leq \epsilon(k).$$

Last bullet: Indistinguishability from truly random function (drawn from set $\mathcal{F}_{n,m}$ of all possible functions with input domain $\{0,1\}^n$ and output domain $\{0,1\}^m$)

Building PRFs from PRGs

Suppose we have a secure PRG $G : \{0,1\}^n \mapsto \{0,1\}^{2n}$. How can we construct a PRF from this?

1-bit input PRF F from PRG G

Define $F : \{0,1\}^k \times \{0,1\} \mapsto \{0,1\}^n$ as

$$F(k,x) = \begin{cases} G(k)[0]||G(k)[1]||\ldots||G(k)[n-1], & \text{if } x = 0, \\ G(k)[n]||G(k)[n+1]||\ldots||G(k)[2n-1], & \text{if } x = 1; \end{cases}$$

i.e., call G on k, and return either the first or the second half of the output based on the input bit x.

How to extend to bigger inputs? (Hint: Tree)

Building PRFs from PRGs

Suppose we have a secure PRG $G : \{0,1\}^n \mapsto \{0,1\}^{2n}$. How can we construct a PRF from this?

1-bit input PRF F from PRG G

Define $F : \{0,1\}^k \times \{0,1\} \mapsto \{0,1\}^n$ as

$$F(k,x) = \begin{cases} G(k)[0] || G(k)[1] || \ldots || G(k)[n-1], & \text{if } x = 0, \\ G(k)[n] || G(k)[n+1] || \ldots || G(k)[2n-1], & \text{if } x = 1; \end{cases}$$

i.e., call G on k, and return either the first or the second half of the output based on the input bit x.

How to extend to bigger inputs? (Hint: Tree)

Building PRFs from PRGs

Suppose we have a secure PRG $G : \{0,1\}^n \mapsto \{0,1\}^{2n}$. How can we construct a PRF from this?

1-bit input PRF F from PRG G

Define $F : \{0,1\}^k \times \{0,1\} \mapsto \{0,1\}^n$ as

$$F(k,x) = \begin{cases} G(k)[0]||G(k)[1]||\ldots||G(k)[n-1], & \text{if } x = 0, \\ G(k)[n]||G(k)[n+1]||\ldots||G(k)[2n-1], & \text{if } x = 1; \end{cases}$$

i.e., call G on k, and return either the first or the second half of the output based on the input bit x.

How to extend to bigger inputs? (Hint: Tree)

Pseudorandom Permutations

Pseudorandom Permutation (PRP)

A function $F : \{0,1\}^k \times \{0,1\}^n \mapsto \{0,1\}^n$ is a pseudorandom permutation if:

- for any $K \in \{0,1\}^k$, F is a bijection
- for any $K \in \{0,1\}^k$, F, and F^{-1} are efficient
- for all PPT adversaries $\mathcal{A}$:

$$\left| \Pr[K \xleftarrow{R} \{0,1\}^k, \mathcal{A}^{F_K}(1^n) = 1] - \Pr[f_n \xleftarrow{R} \mathcal{P}_n, \mathcal{A}^{f_n}(1^n) = 1] \right| \leq \epsilon(k).$$

Last bullet: Indistinguishability from truly random permutation (drawn from set $\mathcal{P}_n$ of all possible permutations with domain $\{0,1\}^n$)

The PRF/PRP Switching Lemma

Blockciphers are assumed to behave like pseudorandom permutations. However, in some proofs, it is easier to analyze the security by assuming the use of a PRF instead.

The PRF/PRP Switching Lemma [1]

Let $n > 1$ be an integer and q the number of queries an adversary $\mathcal{A}$ makes to oracles. Then

$$\left|\Pr[\pi \xleftarrow{R} \mathcal{P}_n, \mathcal{A}^{\pi}(1^n) = 1] - \Pr[\rho \xleftarrow{R} \mathcal{F}_{n,n}, \mathcal{A}^{\rho}(1^n) = 1]\right| \leq \frac{q(q-1)}{2^{n+1}} \,.$$

This switch introduces a loss that is quadratic in the number of oracle queries. Very simple game-based proof!

Hashing

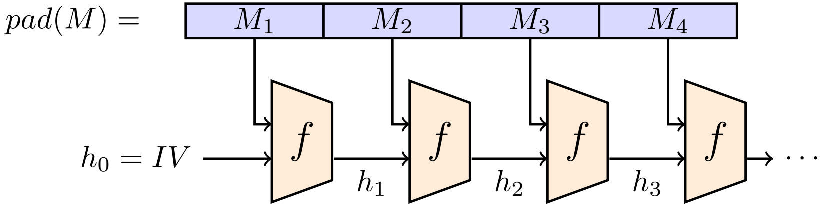
#

History and Present

Merkle-Damgård (MD) Construction

Popular construction to build hash functions with unlimited input domain

- used to build MD-5, SHA-1, SHA-2, ...
- Requires a compression function *f*



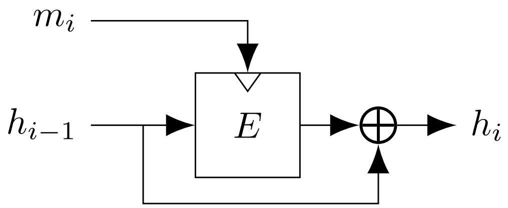
How to build a Compression Function

- Building a compression function $y = f(x_1, x_2)$ from scratch is non-trivial
- Required properties:
  - Easy to compute y given x_1, x_2
  - (Second) pre-image resistance: given y (and x_1, x_2, so that $y = f(x_1, x_2)$), it should be hard to find x_1^*, x_2^* so that $y = f(x_1^*, x_2^*)$ (and $(x_1, x_2) \neq (x_1^*, x_2^*)$)
  - Collision resistance: it should be hard to find $(x_1, x_2), (x_1^*, x_2^*)$ so that $f(x_1, x_2) = f(x_1^*, x_2^*)$
- Common strategy: Build from existing primitives

The Davies-Meyer Construction

Popular construction to build compression function from a block cipher

- Requires a block cipher E_k
- Secure in the ideal cipher model!



Security of Davies-Meyer in the IC model

Lemma (Security of Davies-Meyer [4])

In the IC model, given an m-bit value H, finding any pair (X, K), so that $E_K(X) \oplus X = H$ takes expected time 2^{m-1}.

Proof. To obtain a valid answer, the adversary must find a triple (X, Y, K) such that $Y = E_K(X)$ and $X \oplus Y = H$. Therefore, he must query either $E_K(X) = Y$ or $D_K(Y) = X$. Suppose one has called the box unsuccessfully j times with $E_{K_1}(X_1) = Y_1, \dots, E_{K_j}(X_j) = Y_j$. Due to the nature of the ideal cipher, any value not queried is uniformly distributed over all other possible Y values. For any K, X, only one of these responses can yield in success, so the probability of success at step j is bounded by $1/(2^m - j)$. The probability of failure after j steps is given by $(1 - 1/(2^m - 1)) \cdot (1 - 1/(2^m - 2)) \cdots (1 - 1/(2^m - j - 1)) \geq 1 - j/2^m)$.Thus, the expected running time until success is at least 2^{m-1} steps.

Security of Davies-Meyer in the IC model

Lemma (Security of Davies-Meyer [4])

In the IC model, given an m-bit value H, finding any pair (X, K), so that $E_K(X) \oplus X = H$ takes expected time 2^{m-1}.

Proof. To obtain a valid answer, the adversary must find a triple (X, Y, K) such that $Y = E_K(X)$ and $X \oplus Y = H$. Therefore, he must query either $E_K(X) = Y$ or $D_K(Y) = X$. Suppose one has called the box unsuccessfully j times with $E_{K_1}(X_1) = Y_1, \dots, E_{K_j}(X_j) = Y_j$. Due to the nature of the ideal cipher, any value not queried is uniformly distributed over all other possible Y values. For any K, X, only one of these responses can yield in success, so the probability of success at step j is bounded by $1/(2^m - j)$. The probability of failure after j steps is given by $(1 - 1/(2^m - 1)) \cdot (1 - 1/(2^m - 2)) \cdots (1 - 1/(2^m - j - 1)) \geq 1 - j/2^m$).Thus, the expected running time until success is at least 2^{m-1} steps.

Security of Davies-Meyer in the IC model

Lemma (Security of Davies-Meyer [4])

In the IC model, given an m-bit value H, finding any pair (X, K), so that $E_K(X) \oplus X = H$ takes expected time 2^{m-1}.

Proof. To obtain a valid answer, the adversary must find a triple (X, Y, K) such that $Y = E_K(X)$ and $X \oplus Y = H$. Therefore, he must query either $E_K(X) = Y$ or $D_K(Y) = X$. Suppose one has called the box unsuccessfully j times with $E_{K_1}(X_1) = Y_1, \dots, E_{K_j}(X_j) = Y_j$. Due to the nature of the ideal cipher, any value not queried is uniformly distributed over all other possible Y values. For any K, X, only one of these responses can yield in success, so the probability of success at step j is bounded by $1/(2^m - j)$. The probability of failure after j steps is given by $(1 - 1/(2^m - 1)) \cdot (1 - 1/(2^m - 2)) \cdots (1 - 1/(2^m - j - 1)) \geq 1 - j/2^m)$.Thus, the expected running time until success is at least 2^{m-1} steps.

Security of Davies-Meyer in the IC model

Lemma (Security of Davies-Meyer [4])

In the IC model, given an m-bit value H, finding any pair (X, K), so that $E_K(X) \oplus X = H$ takes expected time 2^{m-1}.

Proof. To obtain a valid answer, the adversary must find a triple (X, Y, K) such that $Y = E_K(X)$ and $X \oplus Y = H$. Therefore, he must query either $E_K(X) = Y$ or $D_K(Y) = X$. Suppose one has called the box unsuccessfully j times with $E_{K_1}(X_1) = Y_1, \ldots, E_{K_j}(X_j) = Y_j$. Due to the nature of the ideal cipher, any value not queried is uniformly distributed over all other possible Y values. For any K, X, only one of these responses can yield in success, so the probability of success at step j is bounded by $1/(2^m - j)$. The probability of failure after j steps is given by $(1 - 1/(2^m - 1)) \cdot (1 - 1/(2^m - 2)) \cdots (1 - 1/(2^m - j - 1)) \geq 1 - j/2^m)$.Thus, the expected running time until success is at least 2^{m-1} steps.

Security of Davies-Meyer in the IC model

Lemma (Security of Davies-Meyer [4])

In the IC model, given an m-bit value H, finding any pair (X, K), so that $E_K(X) \oplus X = H$ takes expected time 2^{m-1}.

Proof. To obtain a valid answer, the adversary must find a triple (X, Y, K) such that $Y = E_K(X)$ and $X \oplus Y = H$. Therefore, he must query either $E_K(X) = Y$ or $D_K(Y) = X$. Suppose one has called the box unsuccessfully j times with $E_{K_1}(X_1) = Y_1, \dots, E_{K_j}(X_j) = Y_j$. Due to the nature of the ideal cipher, any value not queried is uniformly distributed over all other possible Y values. For any K, X, only one of these responses can yield in success, so the probability of success at step j is bounded by $1/(2^m - j)$. The probability of failure after j steps is given by $(1 - 1/(2^m - 1)) \cdot (1 - 1/(2^m - 2)) \cdots (1 - 1/(2^m - j - 1)) \geq 1 - j/2^m$). Thus, the expected running time until success is at least 2^{m-1} steps.

Security of Davies-Meyer in the IC model

Lemma (Security of Davies-Meyer [4])

In the IC model, given an m-bit value H, finding any pair (X, K), so that $E_K(X) \oplus X = H$ takes expected time 2^{m-1}.

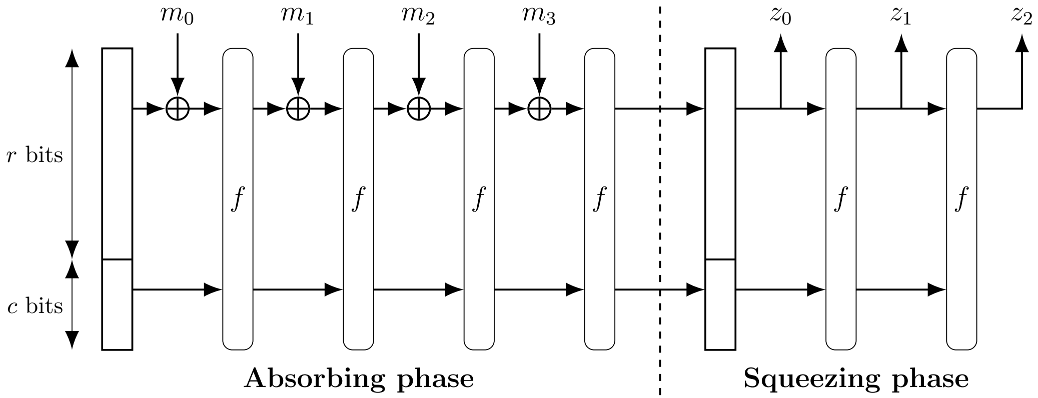
Proof. To obtain a valid answer, the adversary must find a triple (X, Y, K) such that $Y = E_K(X)$ and $X \oplus Y = H$. Therefore, he must query either $E_K(X) = Y$ or $D_K(Y) = X$. Suppose one has called the box unsuccessfully j times with $E_{K_1}(X_1) = Y_1, \dots, E_{K_j}(X_j) = Y_j$. Due to the nature of the ideal cipher, any value not queried is uniformly distributed over all other possible Y values. For any K, X, only one of these responses can yield in success, so the probability of success at step j is bounded by $1/(2^m - j)$. The probability of failure after j steps is given by $(1 - 1/(2^m - 1)) \cdot (1 - 1/(2^m - 2)) \cdots (1 - 1/(2^m - j - 1)) \geq 1 - j/2^m)$.Thus, the expected running time until success is at least 2^{m-1} steps.

The Sponge Construction

Absorb and Squeeze

The Sponge Construction

A modern approach to build hash functions based on an unkeyed permutation.



The Sponge Construction (cont.)

- Very flexible design
- Provable security if f is a random permutation
- Security against collision and pre-image attacks only based on the size of the capacity c and the output

Used (and popularized) by SHA-3 winner Keccak.

- Permutation design with large security margin (only 4 out of 24 rounds attackable)
- Framework extended to many other use cases (e.g., authenticated encryption)

Security of the Sponge

Indistinguishability and Indifferentiability

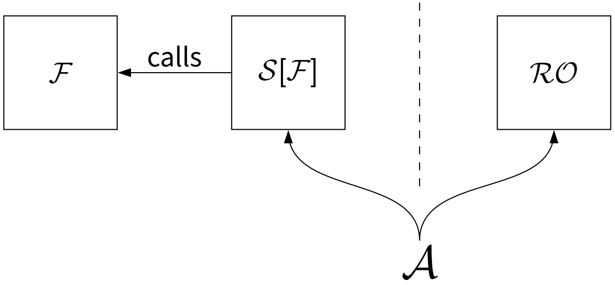
Inner Collisions in Sponges

- State collision: $\text{absorb}(A) = \text{absorb}(B)$, $A \neq B$
- Inner state collision: $\widehat{\text{absorb}}(A) = \widehat{\text{absorb}}(B)$, $A \neq B$
  - Inner State: only concerning the capacity part of the state

Uniform output in the absence of inner collisions [3]

Let f be a random permutation and pad a sponge-compliant padding rule. The bits of the outputs returned by $\text{Sponge}[f, pad, r]$ to a sequence of queries are uniformly and independently distributed if no inner collisions occur during the queries.

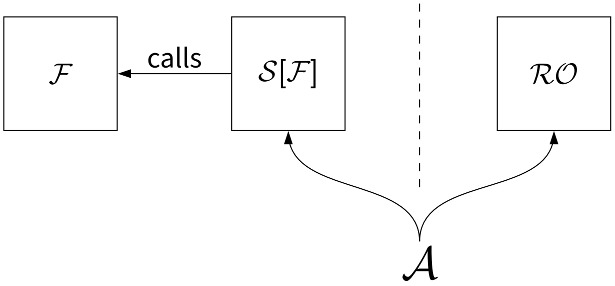
The Indistinguishability Game



An adversary $\mathcal{A}$ is given either access to a random oracle, or an oracle using a sponge construction $\mathcal{S}$, which internally calls the permutation oracle F. His task is to distinguish between the two cases.

Proof idea: Based on previous theorem, distinguishing $\mathcal{S}[\mathcal{F}]$ and $\mathcal{RO}$ only possible if inner collisions occur. Advantage bounded by inner collision probability.

The Indistinguishability Game

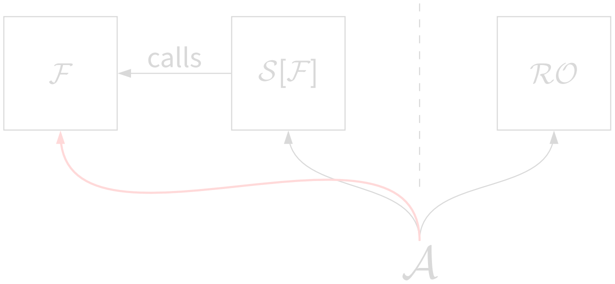


An adversary $\mathcal{A}$ is given either access to a random oracle, or an oracle using a sponge construction $\mathcal{S}$, which internally calls the permutation oracle F. His task is to distinguish between the two cases.
Proof idea: Based on previous theorem, distinguishing $\mathcal{S}[\mathcal{F}]$ and $\mathcal{RO}$ only possible if inner collisions occur. Advantage bounded by inner collision probability.

Problems with Indistinguishability

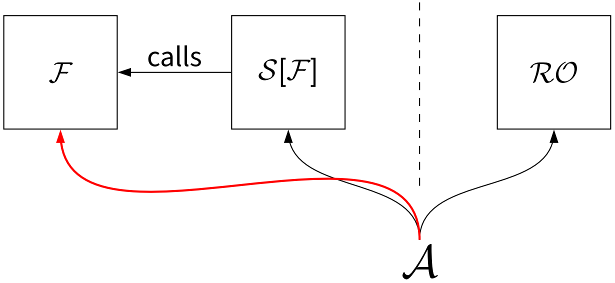
Think about an instantiation of a sponge-based construction (e.g., Keccak).

Knowledge of used f is public! Security assumptions are no longer satisfied, since $\mathcal{A}$ has access to oracle $\mathcal{F}$.

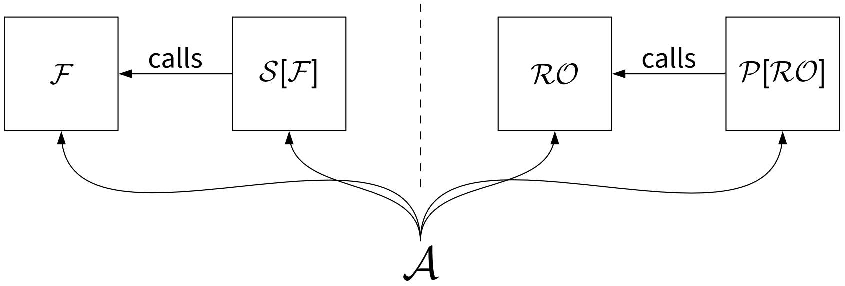


Problems with Indistinguishability

Think about an instantiation of a sponge-based construction (e.g., Keccak). Knowledge of used f is public! Security assumptions are no longer satisfied, since $\mathcal{A}$ has access to oracle $\mathcal{F}$.



Extending to Indifferentiability



Extending the setting to give $\mathcal{A}$ access to both parts of the system. The right side is extended with a simulator P who simulates the permutation and can call the random oracle for consistency.

Indifferentiability of Sponges

In [2], the authors give a concrete simulator $\mathcal{P}$ for the generic padded sponge construction, providing an argument for its security against generic attacks.
The concrete proof and simulator use an argument based on graph representations and result in an advantage of

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{Indiff.}} \leq \frac{(1 - 2^{-r})N^2 + (1 + 2^{-r})N}{2^{c+1}} .$$

This is the reason the size of the capacity part is the dominating factor for the security of sponge constructions.

Further Reading I

[1] Mihir Bellare and Phillip Rogaway.
Code-based game-playing proofs and the security of triple encryption.

[2] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
On the indifferentiability of the sponge construction.
In *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 181–197. Springer, 2008.

[3] Bertoni Guido, Daemen Joan, P Michaël, and VA Gilles.
Cryptographic sponge functions, 2011.

[4] R. S. Winternitz.
A secure one-way hash function built from des.
In *1984 IEEE Symposium on Security and Privacy*, pages 88–88, 1984.