Introduction to Cryptology

8.1 - Hash functions: The ROM

Federico Pintore

Mathematical Institute, University of Oxford (UK)

Michaelmas term 2020

The Random Oracle Model

Let Π be a cryptosystem using a hash function H.

The property that H is collision/preimage resistant might be not enough to to be able prove the security of Π.

The Random Oracle Model

Let Π be a cryptosystem using a hash function H.

The property that H is collision/preimage resistant might be not enough to to be able prove the security of Π.

Instead of using cryptosystems that have no proof of security, an alternative approach was proposed.

The Random Oracle Model

In the Random Oracle Model (ROM), each cryptographic hash function H is *idealised*.

- H is considered to be truly random;

The Random Oracle Model

In the Random Oracle Model (ROM), each cryptographic hash function H is *idealised*.

- H is considered to be truly random;
- it is assumed that this random function is public;
- additionally, it can be evaluated only querying it as an oracle (or a black box).

The Random Oracle Model

In the Random Oracle Model (ROM), each cryptographic hash function H is *idealised*.

- H is considered to be truly random;
- it is assumed that this random function is public;
- additionally, it can be evaluated only querying it as an oracle (or a black box).
- In the real world, each ideal hash function is instantiated by an *appropriate* hash function.

The Random Oracle Model

What is the meaning of *appropriate hash function*?

The Random Oracle Model

What is the meaning of *appropriate hash function*?

- No clear definitions!

The Random Oracle Model

What is the meaning of *appropriate hash function*?

- No clear definitions!
- Concrete hash functions are deterministic and fixed, they cannot behave like random functions!

The Random Oracle Model

What is the meaning of *appropriate hash function*?

- No clear definitions!
- Concrete hash functions are deterministic and fixed, they cannot behave like random functions!

What is the relevance of a proof of security in the ROM?

The Random Oracle Model

What is the meaning of *appropriate hash function*?

- No clear definitions!
- Concrete hash functions are deterministic and fixed, they cannot behave like random functions!

What is the relevance of a proof of security in the ROM?

- Perhaps, the scheme does not have “inherent design flaws”;

The Random Oracle Model

What is the meaning of *appropriate hash function*?

- No clear definitions!
- Concrete hash functions are deterministic and fixed, they cannot behave like random functions!

What is the relevance of a proof of security in the ROM?

- Perhaps, the scheme does not have "inherent design flaws";
- i.e, the only possible attacks are those due to weaknesses in the used hash functions.

The Random Oracle Model

Why is the ROM widely used?

The Random Oracle Model

Why is the ROM widely used?

- So far, there have been no successful real-world attacks on real-world schemes that are proven secure in the ROM.

The Random Oracle Model

Why is the ROM widely used?

- So far, there have been no successful real-world attacks on real-world schemes that are proven secure in the ROM.
- Schemes proven secure in the ROM are usually efficient.

The Random Oracle Model

Why is the ROM widely used?

- So far, there have been no successful real-world attacks on real-world schemes that are proven secure in the ROM.
- Schemes proven secure in the ROM are usually efficient.

If hash functions are not idealised in the proof of security for Π, then Π is said to be secure in the standard model.

Definitions in the ROM

In ROM security definitions:

- the probability is taken over the random choice of H;
- whereas, in the real world, H is instantiated by a deterministic function.

Proofs of security in the ROM

In ROM security proofs:

- the adversary $\mathcal{A}$ needs to query an oracle to evaluate H;
- if x has not been queried yet, then the value $H(x)$ is uniform.

Proofs of security in the ROM

In ROM security proofs:

- the adversary $\mathcal{A}$ needs to query an oracle to evaluate H;
- if x has not been queried yet, then the value $H(x)$ is uniform.

In ROM proofs by reduction:

- the oracle is simulated in the reduction.
- Extractability: when $\mathcal{A}$ queries x, the reduction learns x.
- Programmability: the reduction sets the (uniformly distributed) values of $H(x_i)$ to answer $\mathcal{A}$'s queries.

Further Reading I


Mihir Bellare and Phillip Rogaway.
Random oracles are practical: A paradigm for designing efficient protocols.
In Proceedings of the 1st ACM conference on Computer and communications security, pages 62–73. ACM, 1993.


Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche.
Keccak sponge function family main document.
Submission to NIST (Round 2), 3:30, 2009.


Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya.
Merkle-Damgård revisited: How to construct a hash function.
In Advances in Cryptology–CRYPTO 2005, pages 430–448. Springer, 2005.

Further Reading II


Morris J Dworkin.
SHA-3 standard: Permutation-based hash and extendable-output function.
No. Federal Inf. Process. Stds.(NIST FIPS)-202, 2015.


Pierre Karpman, Thomas Peyrin, and Marc Stevens.
Practical free-start collision attacks on 76-step SHA-1.
In Advances in Cryptology–CRYPTO 2015, pages 623–642. Springer, 2015.


Neal Koblitz and Alfred J Menezes.
The random oracle model: a twenty-year retrospective.
Designs, Codes and Cryptography, pages 1–24, 2015.

Further Reading III


Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone.
Handbook of applied cryptography.
CRC press, 1996.


Marc Stevens.
New collision attacks on SHA-1 based on optimal joint local-collision analysis.
In Advances in Cryptology–EUROCRYPT 2013, pages 245–261. Springer, 2013.


Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov.
The first collision for full SHA-1.
In Annual International Cryptology Conference–CRYPTO 2017, pages 570–596. Springer, CHam, 2005.