On the Security of Hash Functions Employing Blockcipher Post-processing

Donghoon Chang[1], Mridul Nandi[2], Moti Yung[3]

[1] National Institute of Standards and Technology (NIST), USA

[2] C R Rao AIMSCS, Hyderabad, India

[3] Google Inc. and Department of Computer Science, Columbia University, New York, USA

Outline

- Indifferentiability
- Preimage awareness
- Limitation and motivation
- New notion: Computable Message Awareness or CMA
- Applications: Davis-Meyer, PGV, DBL
- Future works and Conclusion

PRO or Indifferentiability

Motivation of Indifferentiability

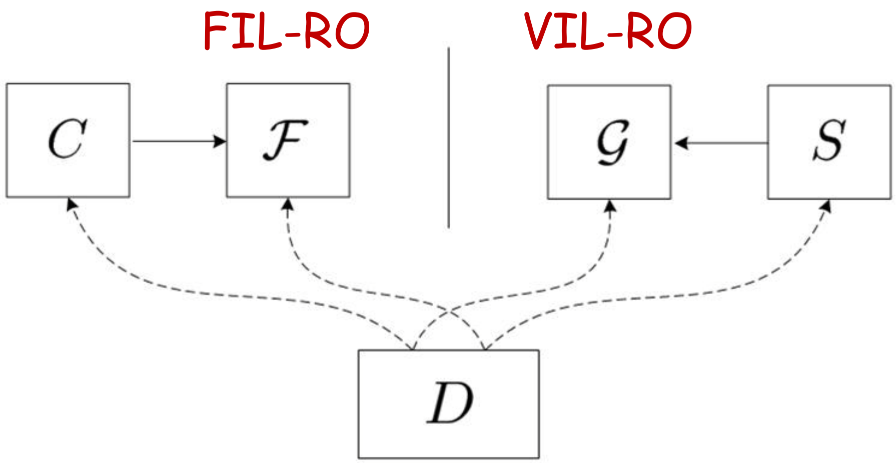
- Introduced by Maurer, Renner, and Holenstein [TCC-04]

Let F be a FIL-RO and G be a VIL-RO.

If C^F (e.g. hash design) is PRO then,

any secure scheme using G is also secure when G is replaced by C^F

Indifferentiability or PRO



- Two points to remember:

1. The simulator S **simulates the underlying primitive F** of C^F such that C behaves like G
2. S can ***access G as an oracle*** but has **NO *information about G-queries of D***

Indifferentiable Security Notion

- Applied to Practical Hash Designs (Coron, Dodis, Malinaud, and Puniya in CRYPTO-05).
  - MD is not PRO, however
  - Prefix-free-MD, chop-MD, NMAC, HMAC are PRO

- It guarantees that the hash domain extensions have no structural flaw.

- NIST recommended random oracle property for SHA-3.

Indifferentiable Security Notion

- **Modular Approach**
  - Split the domain into two or more components
  - Prove the required security properties of each component separately
  - Good for understanding and proving security analysis
  - May end up with better modes

- Dodis, Ristenpart and Shrimpton [DRS Eurocrypt-09] introduced the concept of **Preimage Awareness** and showed that this new (weaker) property can be used for modular approach of proof for PRO.

Preimage Awareness (PrA)

Preimage Awareness (PrA)

- Security Notion for Hash Function
- Motivated by Security Notion of Plaintext-awareness for public-key encryption
- Weaker than a Random Oracle assumption

Preimage Awareness (Informal)

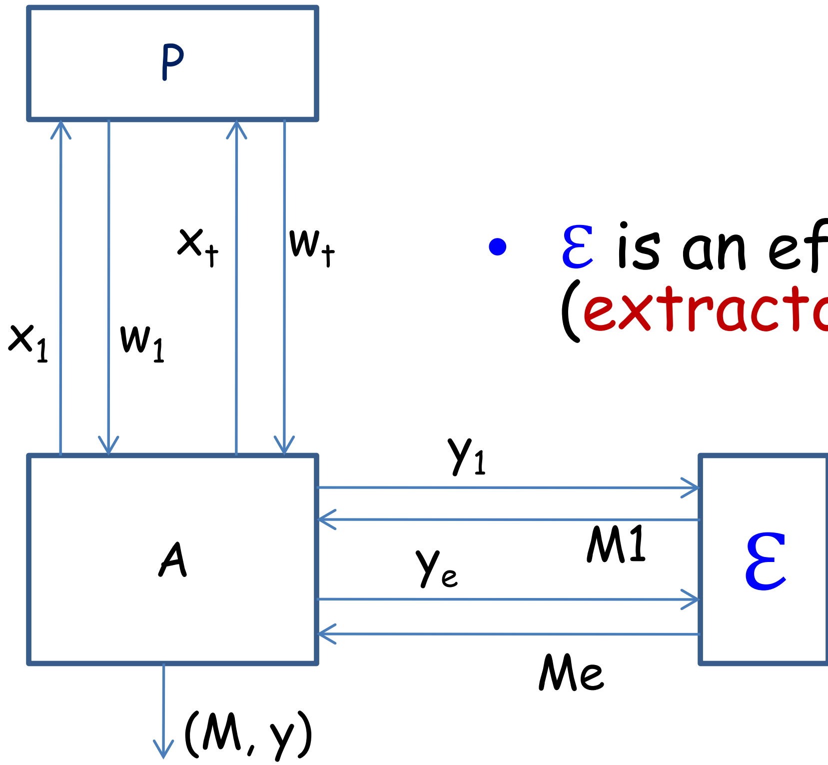
- Security Notion for Hash Function
- Motivated by Security Notion of

A hash function is preimage-aware if it is difficult for any efficient algorithm to come up with a hash output without being aware of the corresponding input message.

Definition of PrA (Formal)

- H^P is a hash function based on an ideal primitive **P**
  - e.g. MD^f with compression function f
- A PrA-adversary A makes
  - P queries and
  - commits (potential H^P outputs) $y_1, \ldots, y_e$ adaptively in an interleaved manner
- $\alpha_i = ((x_1,w_1), \ldots, (x_i,w_i))$
  - the first i **query-response pairs** of P (called an **advice string**)

Definition of PrA (Formal)



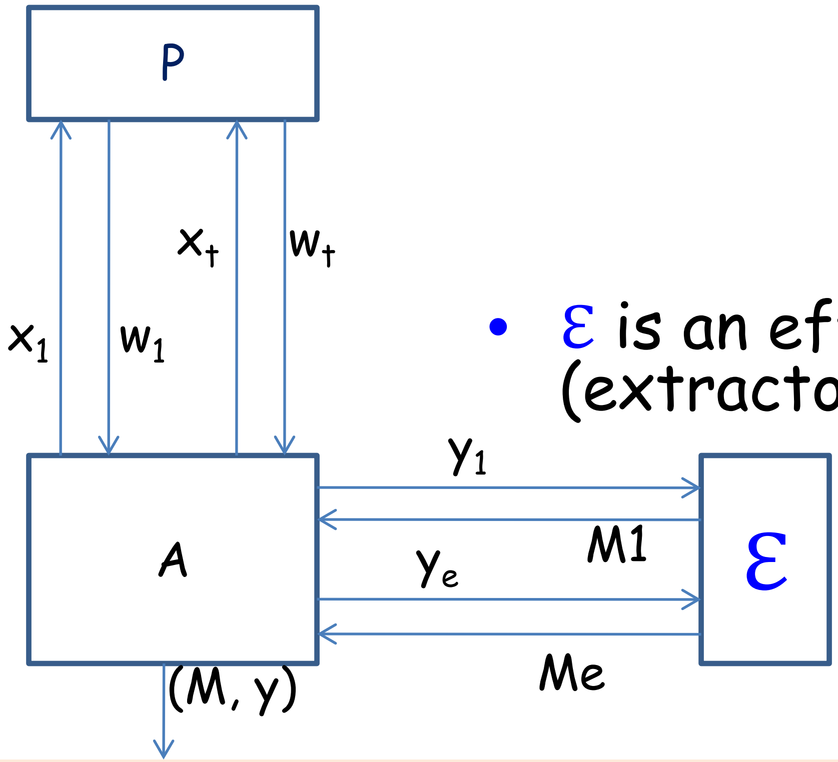
- ℰ is an efficient algorithm (extractor) : ℰ(y, α)=M

- A wins if A later finds M with access to P such that

$$H^P(M) = y_s \text{ and } M \neq M_s.$$

i.e. either A finds collision or preimage on a committed value for which no efficient algorithm can't find preimage.

Definition of PrA (Formal)



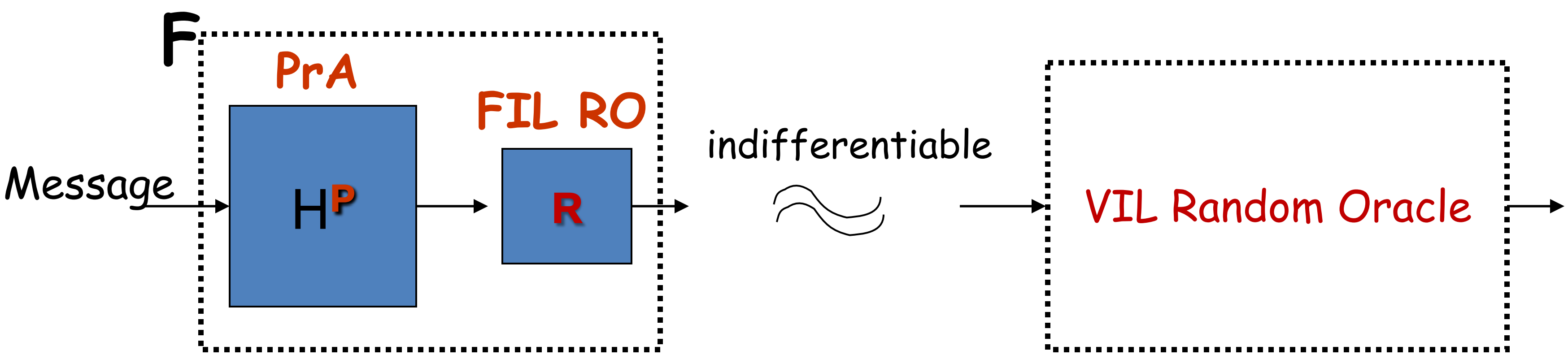
- $\mathcal{E}$ is an efficient algorithm (extractor) : $\mathcal{E}(y, \alpha)=M$

- If no such A exists for an efficient extractor then H^P is called PrA.
- Example: MD^f is PrA if f is so [DRS-09]
- Random oracles are PrA.
- Weaker, easy to verify.

Modular Approach : RO(PrA(·)) = PRO(·)

[Dodis, Ristenpart and Shrimpton Eurocrypt-09]

- When H^P is preimage-aware and R is a FIL random oracle independent from P, then



Corollary: MD with output transformation behaving like a RO independent with a PrA compression function f is PRO.
That is,

$$RO(MD^f(.)) \text{ is PRO}$$

Application

- **Example :** Skein (one of SHA-3 finalists) team proved the indifferentiable security proof of Skein domain extension using this approach.
  - Skein without final output transformation is PrA in the ideal cipher model.
  - Skein's final output transformation is PRO in the ideal cipher model.
  - These two components are believed to behave independently.

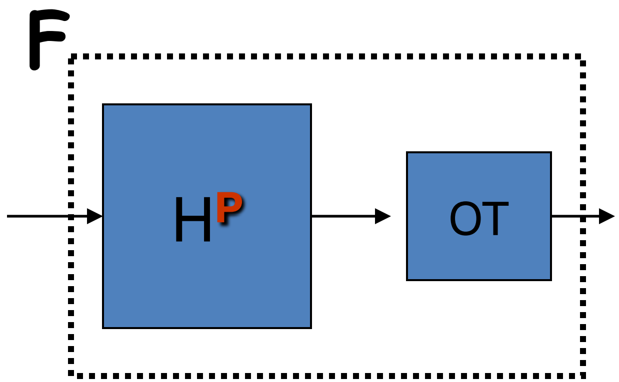
Motivation of Our Results

Limitation of Previous Result

- **Limitation-1:** Many final output transformations of hash functions don't behave as a random oracle
  - Example : Grøstl, Keccak, JH (three of SHA-3 finalists)
- **Limitation-2:** Final output transformations of hash functions may not be independent to the main component
  - Example : Grøstl
- We need more general modular approaches
- We partially resolve the limitation-1

Our Question (an initial step)

- What happens in cases of other output transformations OTs?



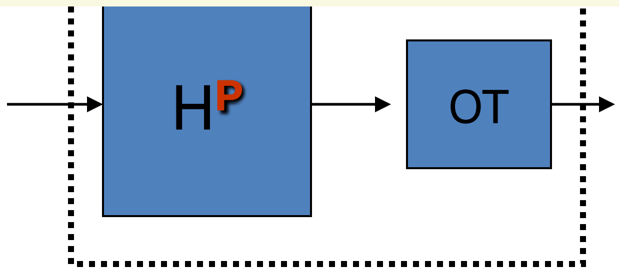
  - E(x)⊕x
  - PGV models
  - Some Double Block Length Constructions
    ex) MDC-2, MDC-4, Tandem DM,….

Our Question (an initial step)

Note that these OT's are **not PRO**. So we can't use previous (RO(PrA()) = PRO) result

Moreover, PrA is **not sufficient**

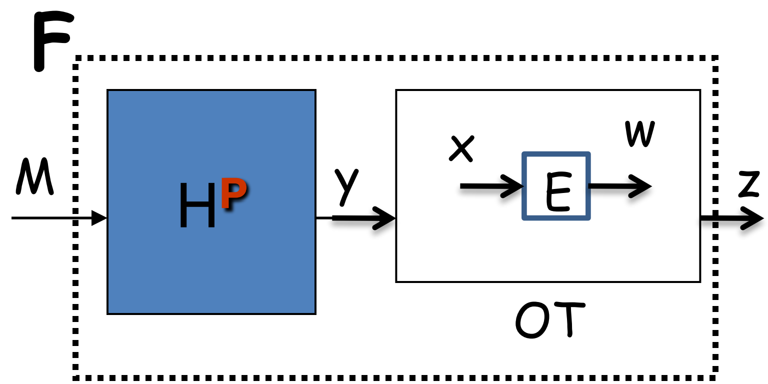
- identity function is PrA but not PRO when output transformation is Davis-Meyer



- E(x)⊕x
- PGV models
- Some Double Block Length Constructions

  ex) MDC-2, MDC-4, Tandem DM,….

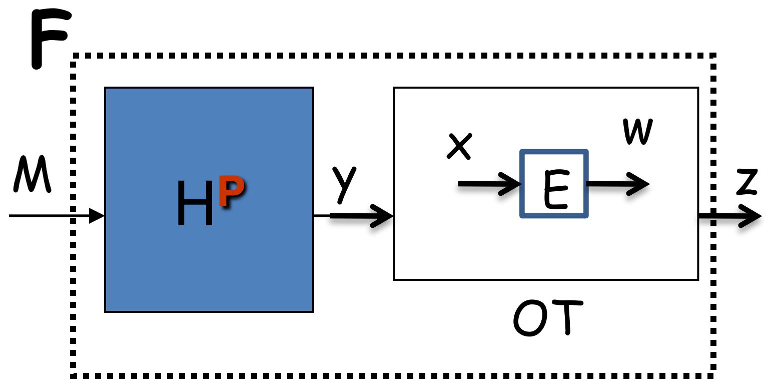
Our Question (an initial step)



- If x and w is uniquely determined from M, $y = H^P(M)$, $z = F(M)$ then, the relation on E (i.e. **E(x) = w**) is obtained by making a F-query and necessary P-queries.

- Since simulator does not know F-query, it has to guess all M (called computable messages) whose outputs are determined by only P-queries.

Our Question (an initial step)



• If x and w is uniquely determined from M, y= $H^P(M)$,

This leads us to introduce new but similar notion called

Computable Message Awareness or CMA

outputs are determined by only P-queries.

Computable Message Awareness or CMA

CMA – Our Formal Definition

- H^{P} is a hash function based on an ideal primitive **P**.
- $a_i = ((x_1,w_1), \ldots, (x_i,w_i))$ is the list of first i query-response pairs of **P**. (called an advice string)

CMA – Our Formal Definition

- A message M is called computable from α if there exists y such that

$$\Pr[H^P(M)=y|\alpha]=1$$

- There is an efficient algorithm (called a computable message extractor)

$$\mathcal{E}_{comp}$$

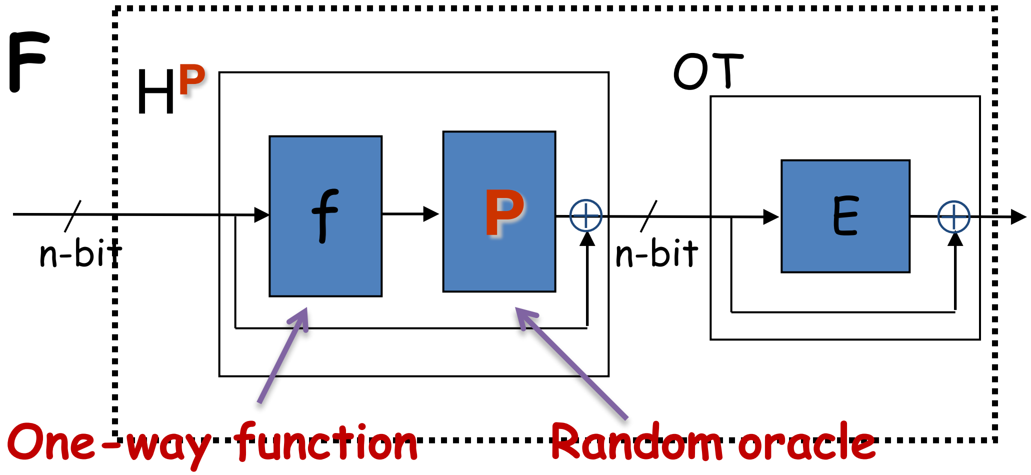
which lists **ALL** computable messages given the advise string α.

- Moreover, for any non-computable messages M,

$$\Pr[\ H^P(M) = y \mid \alpha\ ] \leq \epsilon, \text{ for all } y.$$

Relationship between PrA and CMA

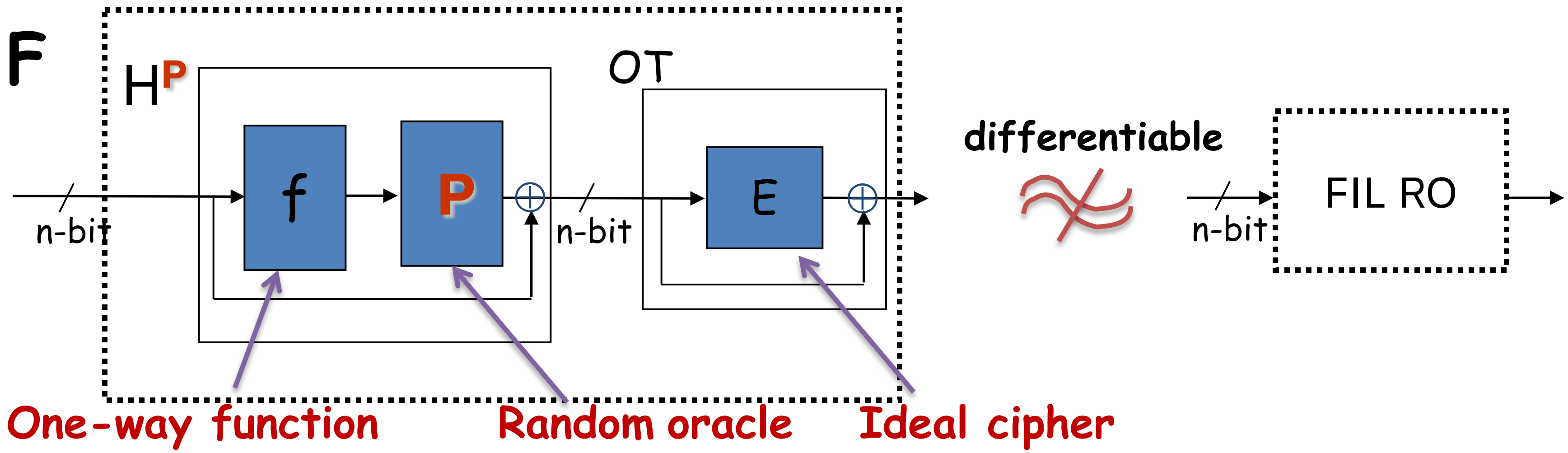
- CMA is defined via presence of efficient extractor only. No commitment and adversary are required.
- CMA is not weaker or stronger notion than PrA.
  - Identity function is not CMA but PrA.
  - $H^P = P^{-1}$ where adversary has only access of P is not PrA but it is CMA.



It is easy to prove that H^P is preimage-resistant and preimage aware but not CMA.

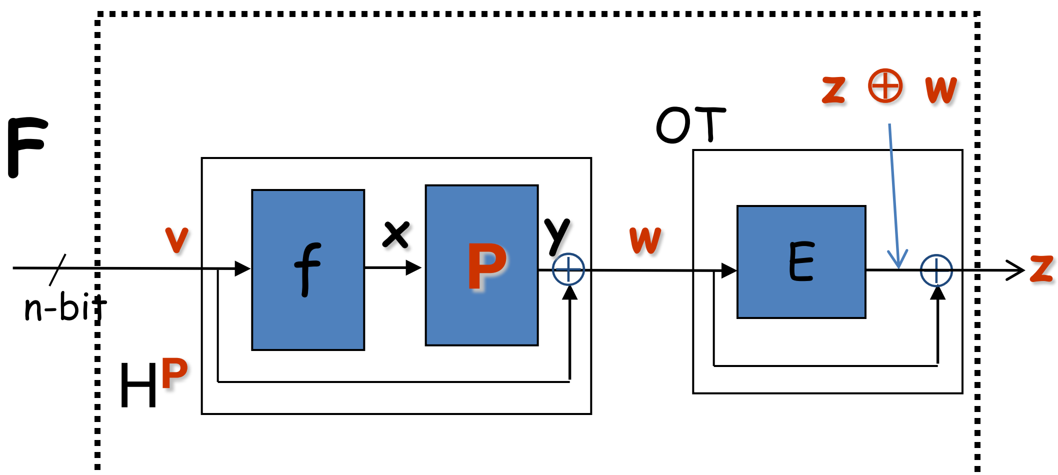
The Case of $OT(x)=E(x)\oplus x$

- F is **differentiable** from a FIL random oracle.



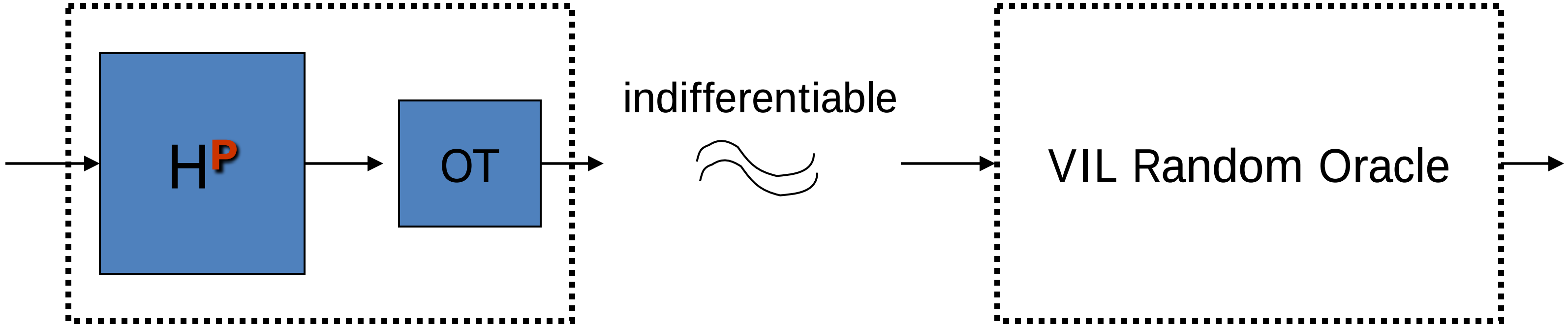
The Case of OT(x)=E(x)⊕x

- An indifferentiable attack on **F**:
  - **Step-1:** Choose v at random compute x = f(v) and make y = P(x) query. v is computable message w.r.t. the advise string
  - **Step-2:** make R(v) query and obtain response z.
  - **Step-3:** Make E^{-1} (z ⊕ w) query and checks the response is w or not.
- NO efficient simulator can compute v (f is one-way) and w (which is v ⊕ y) given (x, y).



Our Main Result

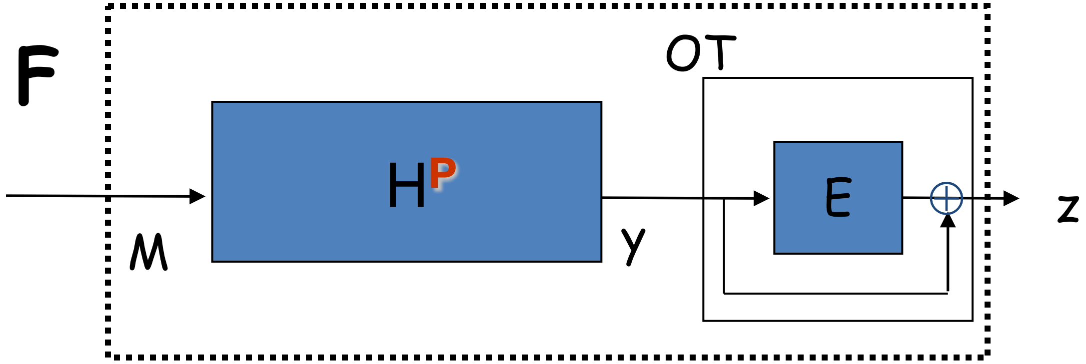
- When H^P is preimage resistant (for a random challenge) preimage-aware, and Computable Message Aware (CMA) (new notion),



where $OT(x)=E(x)\oplus x$ or twelve PGV constructions with an ideal permutation E, and P is independent from E

Our Main Result

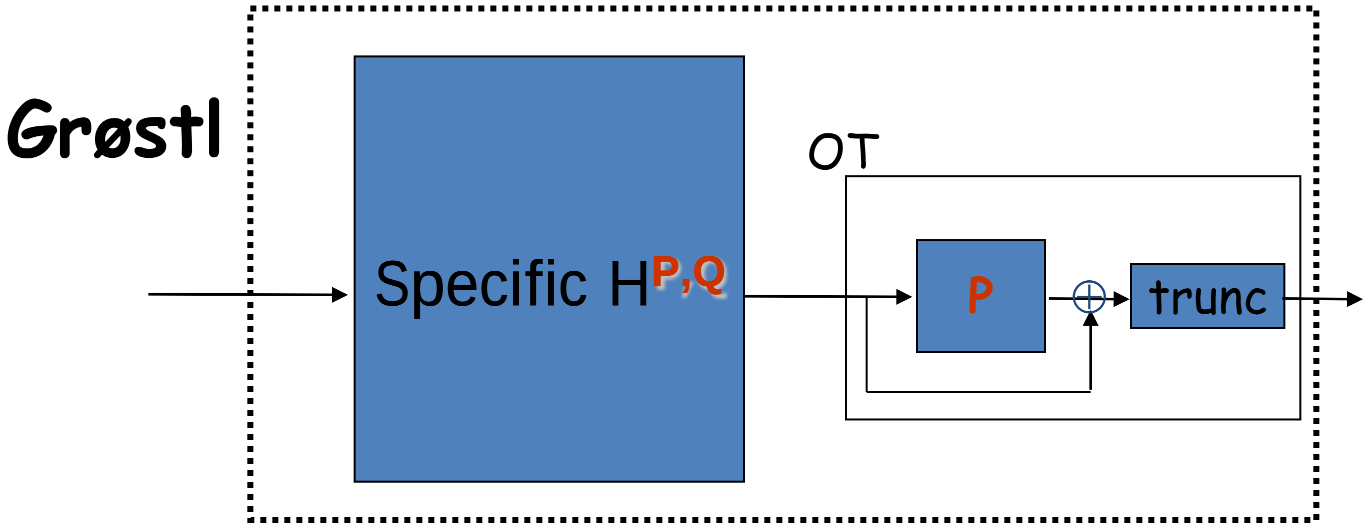
- Case-1: If E query then PrA property takes care since any forward query of OT behaves like a PRO.
- Case-2 (CMA): If E^{-1} query w then simulator first list all computable messages M and checks that w = y ⊕ VIL-RO(M) or not. If yes, then response that y.
- Case-3: If not, then it can response randomly: preimage resistance of H^P for a random challenge.



Similarly for other 12 PGV's

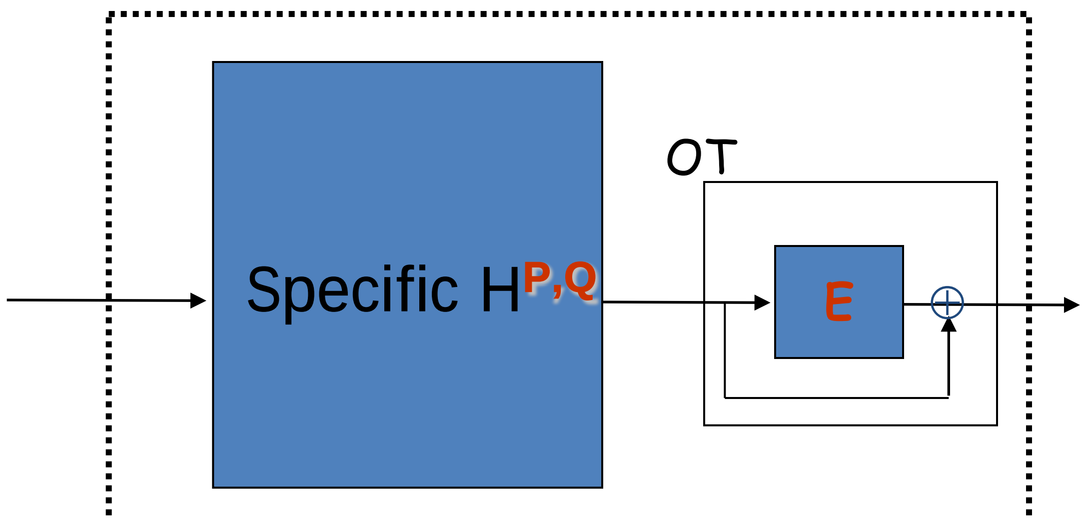
More Results 1/2
(Security Proof of Modified **Grøstl**)

- Two known Results on Grøstl
  - Indifferentiable security proof (by Andreeva et al.)
  - Indiffertiable attack without final truncation (by John Kelsey)



More Results 1/2
(Security Proof of Modified **Grøstl**)

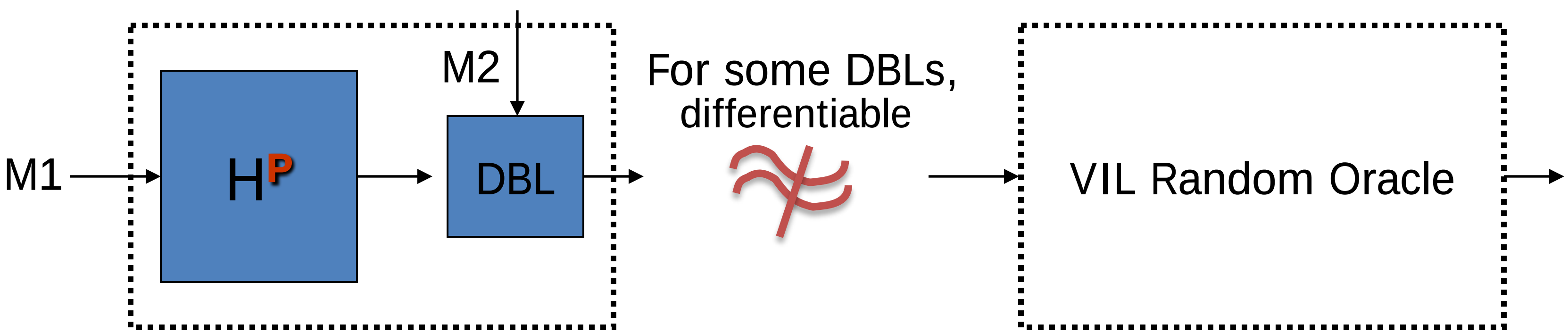
- Our Indifferentiable Security Proof on a modified Grøstl, where P, Q, and E are independent ideal permutations (We DON'T need the final truncation.)



Modified Grøstl

More Results 2/2
(In cases of Some DBLs)

- When H^P is preimage resistant, preimage-aware, and **C**omputable **M**essage **A**ware (CMA),



where DBLs are MDC-2, MDC-4, Tandem DM, etc.

Future Works and a Remark

- We still considered specific output transformations.
- How can we provide a modular approach for more general class of output transformations (OTs)?
  - What security requirements on H^P are needed?
  - What security requirements on OT are needed?
- We have corrected the Proof of

  "$RO(PrA(\cdot)) = PRO(\cdot)$".

Conclusion

- New notion CMA.
- Non-Implication among Preimage, PrA and CMA
- Davis-Meyer, PGV's can be employed as OT
- Some of DBL can not be still employed
- As an application we proved for modified version of Grøstl
- Message from Modular Approach
  - This reduces time to prove and verify the whole security
  - Design efficient H^P with a more load on one-time OT

Questions?