Expedient Non-Malleability Notions for Hash Functions

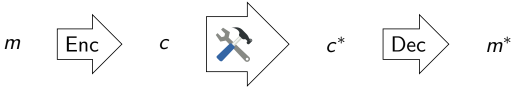
CT-RSA 2011

Paul Baecher, Marc Fischlin, Dominique Schröder

Darmstadt University of Technology, supported by DFG Emmy Noether Program

Introduction: Non-Malleability

- Introduced formally by [DDN00, DDN91]
- in a nutshell, encryption case:



$$(m, m^*) \overset{?}{\in} R$$

- commitments, encryption, zero-knowledge, ...
- what about hash functions?
  - fundamental difference – no private randomness

Non-Malleable Hash Functions

- Given a hash value, output another value such that related preimages exist
- i.e. given H and $H(m)$, output $H(m^*)$ s.t. $(m, m^*) \in R$

Example application: naive authentication

$$(H(\mathsf{secret}||\mathsf{nonce}), \mathsf{nonce}) \quad \rightsquigarrow \quad (H(\mathsf{secret}||\mathsf{nonce}^*), \mathsf{nonce}^*)$$

- First formal foundation in [BCFW09], ASIACRYPT 2009
  Foundations of non-malleable hash and one-way functions

The Simulation Approach

- Simulation-based non-malleability of hash functions [BCFW09]

For every adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$ such that the success probabilities of the following experiments are equal

Adversary's exp.	Simulator's exp.
$x \leftarrow \mathcal{X}$	$x \leftarrow \mathcal{X}$
$y \leftarrow H(x)$	
$y^* \leftarrow \mathcal{A}(y)$	
$x^* \leftarrow \mathcal{A}(x)$	$x^* \leftarrow \mathcal{S}()$
return $R(x, x^*)$	return $R(x, x^*)$

The Simulation Approach

- Simulation-based non-malleability of hash functions [BCFW09]

For every adversary $\mathcal{A}$ there exists a simulator $\mathcal{S}$ such that the success probabilities of the following experiments are equal

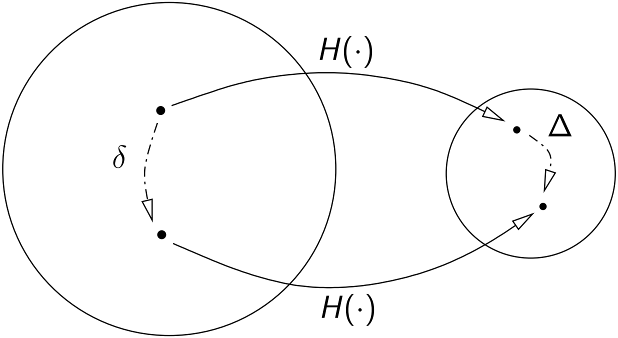
Adversary's exp.	Simulator's exp.
$x \leftarrow \mathcal{X}$	$x \leftarrow \mathcal{X}$
$y \leftarrow H(x)$	
$y^* \leftarrow \mathcal{A}(y)$	
$x^* \leftarrow \mathcal{A}(x)$	$x^* \leftarrow \mathcal{S}()$
return $R(x, x^*)$	return $R(x, x^*)$

- in other words: learning the image y does not help to produce the related value at all
- note: simplified for exposition

The Simulation Approach – Details

- Quite cumbersome for non-theorists
- very strong notion, function must not leak any information
  - otherwise not simulatable
- proving malleability: need to show $\exists \mathcal{A} \forall \mathcal{S} \ldots$
  - for <u>all</u> simulators
- the case of $H(x) = c$
  - non-malleable under this definition!

Our Notion – Approach



Our Notion – Details

H non-malleable iff for all adversaries $\mathcal{A}$ the win probability in the following game is negligible

NM-Game
$x \leftarrow \mathcal{X}$
$y \leftarrow H(x)$
$(y^*, \phi) \leftarrow \mathcal{A}(y)$
Return 1 iff
$\quad \mathrm{H}(\phi(x)) = y^*$

- Transformation function ϕ

On Transformation Functions

Adversary specifies function

- arbitrary functions do not work (consider constant)
- need to restrict this function to some class

On Transformation Functions

Adversary specifies function

- arbitrary functions do not work (consider constant)
- need to restrict this function to some class

Useful classes

- group-induced transformations
- for some group $(G, \odot)$ define $\Phi^{\odot} = \{\phi_\delta : \delta \in G\}$ where $\phi_\delta(x) = x \odot \delta$
- e.g. induces "bit-flips" for $(\{0,1\}^\ell, \oplus)$
- originates from related-key attacks on PRFs, [Luc04, BC10]

Comparing Both Notions

We have

- simulation-based non-malleability (SNM)
- game-based non-malleability (GNM)

our notion is strictly weaker:

(1) SNM $\Rightarrow$ GNM

(2) GNM $\not\Rightarrow$ SNM

intuitions

(1) GNM-adversary can be transformed easily into SNM-adversary, but simulator cannot succeed without contradicting min-entropy

(2) consider a function that leaks one bit, i.e. $H(x) = F(x)||x_1$

Weaker but Useful

GNM is strictly weaker than SNM, but

- can capture a large class of typical attacks
- may be sufficient for proving security of a scheme
- usually easier to handle, easier to verify/refute

Examining Merkle-Damgård

- Recall: $H(m_0||\ldots||m_\ell) = h(\ldots h(h(\mathsf{IV}, m_0), m_1)\ldots, m_\ell)$
- clearly malleable for appending transformations ($\Phi^{||}$), even if h is modeled as a RO
  - also malleable in the simulation sense

Examining Merkle-Damgård

- Recall: $H(m_0||\ldots||m_\ell) = h(\ldots h(h(\mathrm{IV}, m_0), m_1)\ldots, m_\ell)$
- clearly malleable for appending transformations ($\Phi^{||}$), even if h is modeled as a RO
  - also malleable in the simulation sense

- However, for a different (length-preserving) class $\Phi^{\oplus}$:
- h modeled as RO $\Rightarrow$ H is $\Phi^{\oplus}$-non-malleable
  - alleged adversary queries all intermediate values and outputs δ
  - reduction reconstructs original message, contradicts min-entropy

Matyas-Meyer-Oseas-Like Constructions

- Is non-malleability robust?
- consider $h(m) = f(m) \oplus m$ where f is non-malleable
- assuming uniform input distributions, non-malleability of h does not necessarily follow

Matyas-Meyer-Oseas-Like Constructions

- Is non-malleability robust?
- consider $h(m) = f(m) \oplus m$ where f is non-malleable
- assuming uniform input distributions, non-malleability of h does not necessarily follow

$$f(m_0 || m_1) = \quad \mathcal{O}(m_0) \oplus (g(m_0)||g(m_1)) \quad || \quad m_0 \oplus m_1$$

Matyas-Meyer-Oseas-Like Constructions

- Is non-malleability robust?
- consider $h(m) = f(m) \oplus m$ where f is non-malleable
- assuming uniform input distributions, non-malleability of h does not necessarily follow

$$f(m_0||m_1) = \quad \mathcal{O}(m_0) \oplus (g(m_0)||g(m_1)) \quad || \quad m_0 \oplus m_1$$

$$f(m_0||m_1) \oplus m_0||m_1 = \quad m_0 \oplus \mathcal{O}(m_0) \oplus (g(m_0)||g(m_1)) \quad || \quad m_0$$

Matyas-Meyer-Oseas-Like Constructions

- Is non-malleability robust?
- consider $h(m) = f(m) \oplus m$ where f is non-malleable
- assuming uniform input distributions, non-malleability of h does not necessarily follow

$$f(m_0||m_1) = \quad \mathcal{O}(m_0) \oplus (g(m_0)||g(m_1)) \quad || \quad m_0 \oplus m_1$$

$$f(m_0||m_1) \oplus m_0||m_1 = \quad m_0 \oplus \mathcal{O}(m_0) \oplus (g(m_0)||g(m_1)) \quad || \quad m_0$$

- MMO (e.g. Skein) is structurally similar – but f is a cipher

Bellare-Rogaway Encryption Scheme

- IND-CCA encryption scheme from a trapdoor permutation and two random oracles
- instantiating one oracle with $\oplus$-nm hash function retains security
  - improvement over [BCFW09]
- also need preimage hiding property (implied in [BCFW09])

Rehash

- Non-malleability of hash functions is quite new
- simulation-based definition is strong, but comes with deficits
- expedient and useful game-based definition
- relevant applications and constructions

The End

Thank you!

?

References

Mihir Bellare and David Cash.
Pseudorandom functions and permutations provably secure against related-key attacks.
In Tal Rabin, editor, *Advances in Cryptology – CRYPTO 2010*, volume 6223 of *Lecture Notes in Computer Science*, pages 666–684, Santa Barbara, CA, USA, August 15–19, 2010. Springer, Berlin, Germany.

Alexandra Boldyreva, David Cash, Marc Fischlin, and Bogdan Warinschi.
Foundations of non-malleable hash and one-way functions.
In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 524–541, Tokyo, Japan, December 6–10, 2009. Springer, Berlin, Germany.

Danny Dolev, Cynthia Dwork, and Moni Naor.
Non-malleable cryptography.
In *23rd Annual ACM Symposium on Theory of Computing*, pages 542–552, New Orleans, Louisiana, USA, May 6–8, 1991. ACM Press.

Danny Dolev, Cynthia Dwork, and Moni Naor.
Nonmalleable cryptography.
SIAM Journal on Computing, 30(2):391–437, 2000.

Stefan Lucks.
Ciphers secure against related-key attacks.
In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption – FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 359–370, New Delhi, India, February 5–7, 2004. Springer, Berlin, Germany.