# On Randomizing Hash Functions to Strengthen the Security of Digital Signatures.

**Gauravaram, Praveen**

*Link to article, DOI:*
10.1007/978-3-642-01001-9_5

*Publication date:*
2009

*Document Version*
Publisher's PDF, also known as Version of record



*Citation (APA):*
Gauravaram, P. (2009). On Randomizing Hash Functions to Strengthen the Security of Digital Signatures. [Sound/Visual production (digital)]. Advances in Cryptology - EUROCRYPT 2009 : 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, 01/01/2009DOI: 10.1007/978-3-642-01001-9_5

# On Randomizing Hash Functions to Strengthen the Security of Digital Signatures

Praveen Gauravaram and Lars R. Knudsen

Department of Mathematics
Technical University of Denmark

EUROCRYPT 2009
27 April 2009

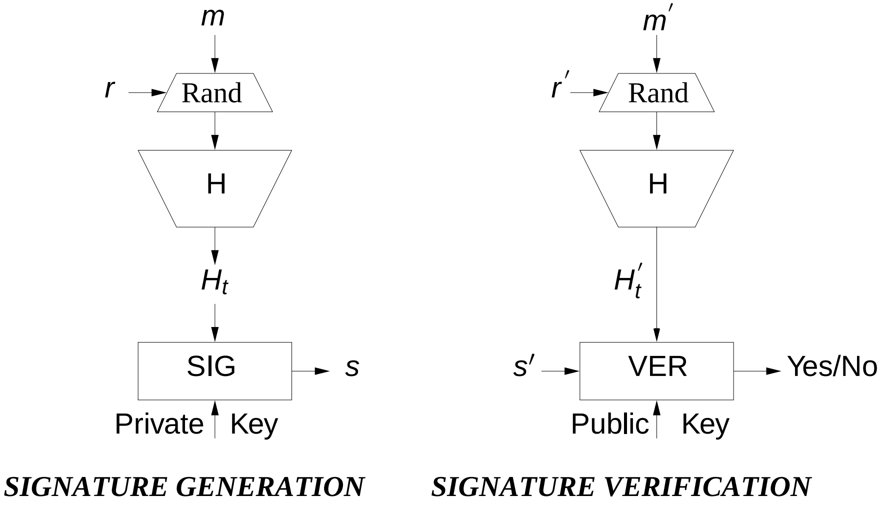
# Outline

1. Strengthening digital signatures via randomized hashing
2. Generic forgery attack on RMX-hash-then-sign schemes
3. Forging some randomize-hash-then-sign schemes

# Outline

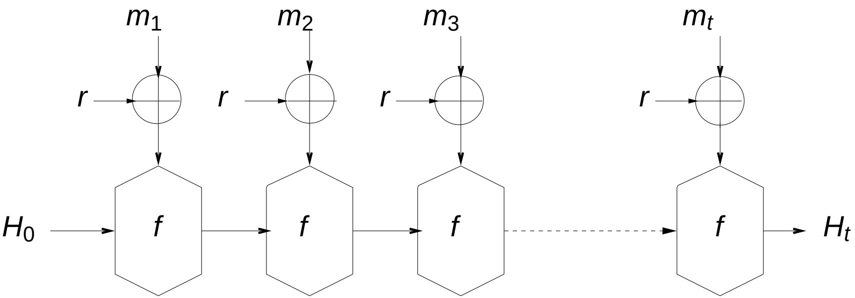
1. Strengthening digital signatures via randomized hashing
2. Generic forgery attack on RMX-hash-then-sign schemes
3. Forging some randomize-hash-then-sign schemes

# Randomize-hash-then-sign signatures



# Randomized hashing (Halevi and Krawczyk-Crypto'06)

$H_r(m) = H((m_1 \oplus r)\|(m_2 \oplus r)\| \ldots \|(m_t \oplus r))$ where $H$ is an $n$-bit hash function.

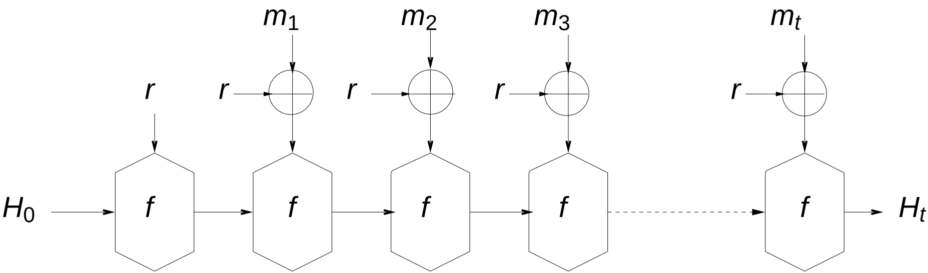


# Properties

- Target collision resistance (TCR): Difficulty in finding $m^* \neq m$ after committing to $m$ and receiving $r$, such that $H_r(m) = H_r(m^*)$.
- If $f$ is c-SPR or e-SPR then $H_r$ is TCR.
- Signing using $H_r$:
  - Use $\mathrm{SIG}$ algorithm to sign the pair $(r, H_r(m))$.
  - Certain signature schemes either do not accommodate signing of both $r$ and $H_r(m)$ (DSA) or requires implementation changes (RSA).

# RMX Hash function mode

$$\tilde{H}_r(m) = H(r \| (m_1 \oplus r) \| (m_2 \oplus r) \| \ldots \| (m_t \oplus r))$$



# Properties

- Enhanced target collision resistance (eTCR): Difficulty in finding $(r^*, m^*) \neq (r, m)$ after committing to $m$ and receiving $r$, such that $\tilde{H}_r(m) = \tilde{H}_{r^*}(m^*)$.
- If $f$ is c-SPR or e-SPR then $\tilde{H}_r$ is eTCR
- Signing using $\tilde{H}_r$:
  - Use $\mathrm{SIG}$ algorithm to sign just $\tilde{H}_r(m)$

# Known results on randomize-hash-then-sign schemes

Forging signatures based on $H_r$ and $\tilde{H}_r$ (via off-line) requires:

- Solving a cryptanalytical problem which is related to finding second preimages in $H$.
  - Kelsey-Schneier second preimage attack on $H$ for a message of $2^k$ blocks in $2^{n-k}$ work.
- Breaking c-SPR or e-SPR property of $f$.

To forge randomize-hash-then-sign signatures:

- How many queries to the signer are required?
- What properties of $f$ or $H$ can we exploit?
- For how many times we need to play the game of TCR/eTCR?

# Known results on randomize-hash-then-sign schemes

Forging signatures based on $H_r$ and $\tilde{H}_r$ (via off-line) requires:

- Solving a cryptanalytical problem which is related to finding second preimages in $H$.
  - Kelsey-Schneier second preimage attack on $H$ for a message of $2^k$ blocks in $2^{n-k}$ work.
- Breaking c-SPR or e-SPR property of $f$.

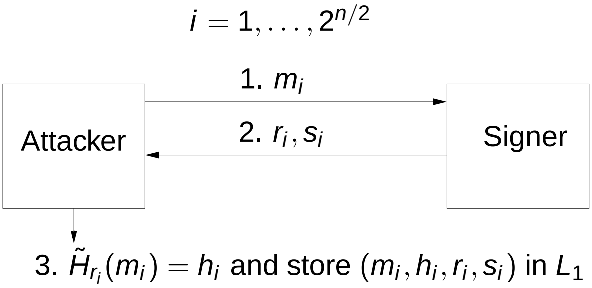
To forge randomize-hash-then-sign signatures:

- How many queries to the signer are required?
- What properties of $f$ or $H$ can we exploit?
- For how many times we need to play the game of TCR/eTCR?

# Outline

1. Strengthening digital signatures via randomized hashing
2. Generic forgery attack on RMX-hash-then-sign schemes
3. Forging some randomize-hash-then-sign schemes

# Generic forgery of RMX-hash-then-sign schemes (Dang-Perlner)




4. Choose some $r^*$, do $\tilde{H}_{r^*}(m_i^*) = h_i^*$ and store $(h_i^*, m_i^*)$ in $L_2$.
5. Find $(r, m) \in L_1$ and $m^* \in L_2$ such that $\tilde{H}_r(m) = \tilde{H}_{r^*}(m^*)$.
6. Signature on $m$ is also valid on $m^*$.

# Limitations

Attack does not succeed when:

- Same salt $r$ is used for both hashing and signing (DSA, ECDSA, RSA-PSS).
- Signatures are based on TCR hashing $H_r$.

We can overcome these limitations when $f$ has fixed points. Davies-Meyer compression function used in many popular hashes such as MD5 and SHA family has this property.

# Limitations

Attack does not succeed when:

- Same salt $r$ is used for both hashing and signing (DSA, ECDSA, RSA-PSS).
- Signatures are based on TCR hashing $H_r$.

We can overcome these limitations when $f$ has fixed points. Davies-Meyer compression function used in many popular hashes such as MD5 and SHA family has this property.

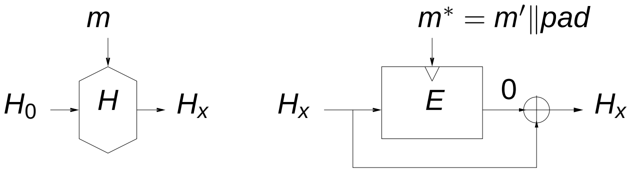
# Outline

1. Strengthening digital signatures via randomized hashing
2. Generic forgery attack on RMX-hash-then-sign schemes
3. Forging some randomize-hash-then-sign schemes

# Forging $H$-SIG scheme using Dean's method

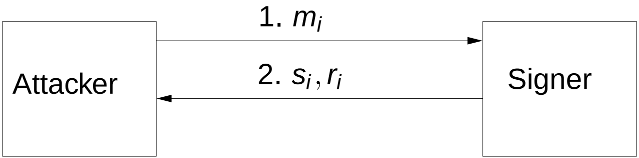
Exploit the fixed point property of Davies-Meyer.

1. Find $2^{n/2}$ hash values of $H$ for equal length messages $m_i$. Store $(H_i, m_i)$ in $L_1$.
2. Find $2^{n/2}$ fixed points $(H_i, m_i^*)$ for $f$. Store them in $L_2$.
3. Find $(H_x, m) \in L_1$ and $(H_x, m^*) \in L_2$ such that $H(m) = H_x = f(H_x, m^*)$.
4. $H(m) = H(m \| m^*) \Rightarrow SIG(H(m)) = SIG(H(m \| m^*))$.



# Forging RMX-hash-then-sign schemes

- On-line phase:
  Ask the signer for the signatures $s_i$ on $2^{n/2}$ equal length messages $m_i$. Store $(r_i, m_i, s_i)$ in $L_1$.

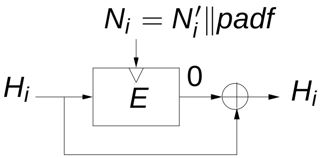



Note that $s_i = \mathrm{SIG}(\tilde{H}_{r_i}(m_i))$ and $i = 1, \ldots, 2^{n/2}$.

- Off-line phase: Compute $\tilde{H}_{r_i}(m_i)$ and add to $L_1$.

# Forging RMX-hash-then-sign schemes

- Off-line phase:
  Compute $2^{n/2}$ fixed points $(H_i, N_i)$ for $f$. Store them in $L_2$.



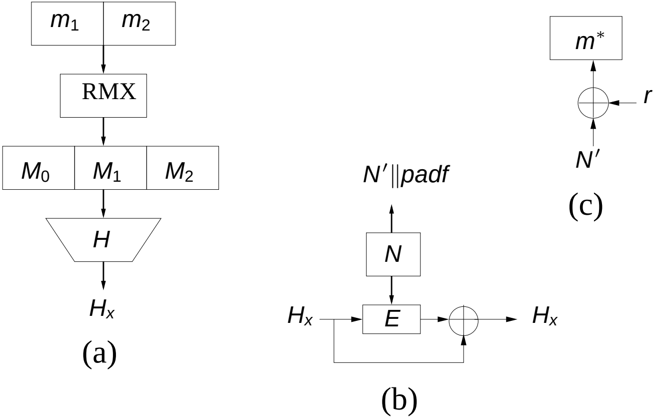

$$i = 1, \ldots, 2^{n/2}$$

# Forging RMX-hash-then-sign schemes

- Off-line phase:
  1. Find $(m, r, \tilde{H}_r(m)) \in L_1$ and $(N, \tilde{H}_r(N)) \in L_2$ such that $\tilde{H}_r(m) = H_x = f(H_x, N)$.
  2. This implies $\tilde{H}_r(m) = H_x = \tilde{H}_r(m \| m^*)$ where $m^* = N \oplus r$.
  3. $\text{SIG}(\tilde{H}_r(m)) = \text{SIG}(\tilde{H}_r(m \| m^*))$.
  4. Output $m \| m^*$ as the forgery of $m$.

Complexity: $2^{n/2}$ chosen messages, $2^{n/2+1}$ evaluations of $f$ and $2^{n/2}$ memory

# Forging RMX-hash-then-sign schemes



# Applications of the approach

- Independent of the size of $r$.
- RMX-hash-then-sign scheme in NIST's SP 800-106.
- Applies to signatures based on $H_r(m)$, $H(r\|H_r(m))$ and $\mathrm{RO}(H_r(m))$.
- Many others (please see article).

$n$-bit hash with at least $2n$-bit intermediate state thwarts the attack as in many SHA-3 candidates.

# Applications of the approach

- Independent of the size of $r$.
- RMX-hash-then-sign scheme in NIST's SP 800-106.
- Applies to signatures based on $H_r(m)$, $H(r \| H_r(m))$ and $\mathrm{RO}(H_r(m))$.
- Many others (please see article).

*n*-bit hash with at least 2*n*-bit intermediate state thwarts the attack as in many SHA-3 candidates.

# Summary of our analysis

1. Complements previous analysis:
   - Off-line birthday attacks do not help.
   - Attacks must be online but not much over birthday complexity.
   - Worth investigating SPR properties of compression functions in the SHA-3 competition.
2. Security of RMX-hash-then-sign schemes is similar to that of HMAC.

Nitpick in Conclusion: “Our research shows that randomized hashing is not easy to implement safely.”

One may have to be careful while choosing the same salt for both hashing and signing. Further research is required.

# Summary of our analysis

1. Complements previous analysis:
   - Off-line birthday attacks do not help.
   - Attacks must be online but not much over birthday complexity.
   - Worth investigating SPR properties of compression functions in the SHA-3 competition.
2. Security of RMX-hash-then-sign schemes is similar to that of HMAC.

Nitpick in Conclusion: "Our research shows that randomized hashing is not easy to implement safely."

One may have to be careful while choosing the same salt for both hashing and signing. Further research is required.

# Summary of our analysis

1. Complements previous analysis:
   - Off-line birthday attacks do not help.
   - Attacks must be online but not much over birthday complexity.
   - Worth investigating SPR properties of compression functions in the SHA-3 competition.
2. Security of RMX-hash-then-sign schemes is similar to that of HMAC.

Nitpick in Conclusion: “Our research shows that randomized hashing is not easy to implement safely.”

One may have to be careful while choosing the same salt for both hashing and signing. Further research is required.

# Acknowledgments


- Danish Research Council for Technology and Production Sciences grant number 274-08-0052.
- European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II.
- Shai Halevi and Hugo Krawczyk
- Peter Bellen, Gregor Leander, Krystian Matusiewicz and Erik Zenner (MAT, DTU).
  Guo Jian (NTU, Singapore) and Choudary Gorantla (QUT, Australia).

**Thank You**