

# Classification of Hash Functions Suitable for Real-life Systems

---

Yasumasa Hirai (NTT DATA Corp.)

Takashi Kurokawa (NICT)

Shin'ichiro Matsuo (NTT DATA Corp.)

Hidema Tanaka (NICT)

Akihiro Yamamura (NICT)



# Background

---

- Hash Functions
  - widely used in many information systems.
  - their security got attention after Aug. 2004.
- Security?
  - There are cryptographic definitions and evaluations.
    - Collision resistant, 2nd pre-image resistant ...
  - not easy to apply them for designing real secure systems
    - "Is collision resistance suitable for security of our system? "
  - We need security classification which bridge cryptographic security and system security.



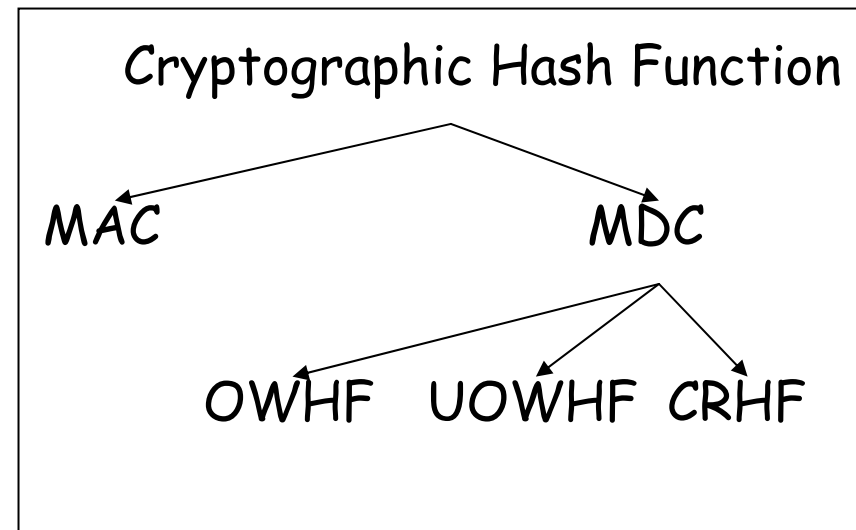
# Outline

---

- (Existing) Cryptographic Classification
- Current usages of hash functions in real-life systems
- Gaps between cryptographic class and current usage
- Proposal of new classification
- Other issues for real-life systems
- Conclusion

# Cryptographic Classification

- Cryptographic hash functions are classified into four categories
  - MAC (omit in this talk)
  - OWHF (omit in this talk)
  - CRHF
  - UOWHF





# Collision Resistant Hash Function

---

- Hash function  $h : \{0, 1\}^* \rightarrow \{0, 1\}^n$
- Computational cost to find  $x$  and  $x'$  s.t.  $h(x) = h(x')$  is not smaller than  $2^{n/2}$  .
- There are efficient realizations:
  - Example: SHA-256/384/512, SHA-1(?)
- hard to prove their security
- widely used and hard to replace



# Universal One-Way Hash Function

---

- Keyed hash function s.t.
  - Adversary choose  $x$
  - For randomly chosen  $h_K$ , it is hard to find  $y (\neq x)$   
s.t.  $h_K(x) = h_K(y)$ .
- can construct provable secure signature scheme with UOWHF
- Few practical realizations.
  - Less efficient than CRHF (Performance, Key size)



# Hash functions in Real-Systems

---

- Hash functions are widely used in real systems
  - For securing information systems
  - Cryptographic algorithm
  - Cryptographic protocols
- Many of them are built into hardware/ software products
  - Good news
    - System designer can easily construct secure system.
  - Bad news
    - In some cases, he choose bad hash function without knowledge
    - In some cases, he does not know status of chosen hash
    - In some cases, he does not know if the system use hash...
- Study about hash usage in real-system is important!



# Requirements in Real-Systems with hash

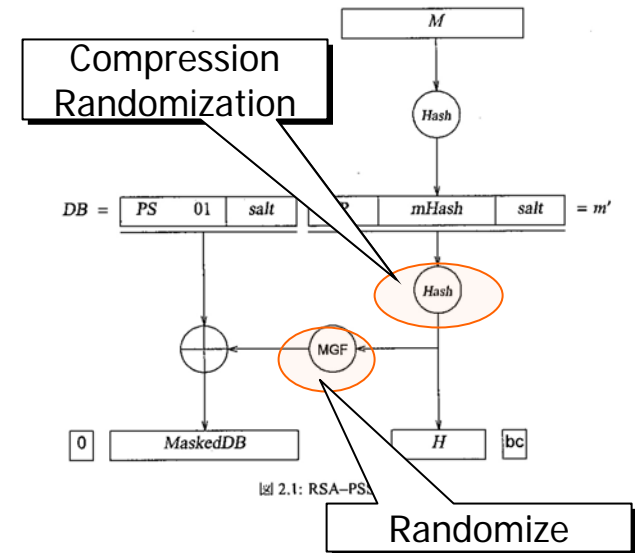
---

- Security requirements
  - Confidentiality
  - Authentication, Certification
  - Integrity
- Requirements in system development aspects
  - Choosing algorithm
  - Development cost, period and system life-cycle
- Requirements from services
  - Compliance
  - Enforcement of products



# Usages of Hash Functions - Certification

- Digital Signature
  - Usage of Hash:
    - Compression
    - Randomization (ex. PSS padding)
  - Required security:
    - second pre-image resistance
    - must be valid 5 years for SOX act, 7 years for HIPPA
- Other examples
  - PKI
  - Time-stamping





# Usages of Hash Functions

## - Authentication

---

- Kerberos
  - Usage of Hash
    - calculate secret key of the entered client
    - Integrity of protocol message
  - Required security
    - secrecy of the client password
    - must be valid for a session
- Other examples
  - IEEE 802.1X-EAP
  - APOP



# Usages of Hash Functions

## - Secure Communication

---

- IPsec
  - Usage of Hash
    - Authentication in key exchange part (IKE)
    - Integrity check (protocol messages)
  - Required security
    - Second pre-image resistance
    - Must be valid in one session
- Other examples
  - SSL/TLS
  - SSH



# Usages of Hash Functions

## - Secure E-mail

---

- S/MIME
  - Usage of Hash
    - Digital signature
  - Required security
    - Second pre-image resistance
    - Must be valid for long period if used for evidence
- Other example
  - PGP



# Usages of Hash Functions

## - Others

---

- Packet Sampling/ filtering (PSAMP: IETF)
  - Usage of Hash
    - Compression for efficient filtering of packets
  - Requirements
    - Collision resistant
    - Output length of hash function can be short
- Other usages
  - Database matching
  - Software Download
  - IDS
  - DKIM



# Security of Hash in Real System

---

- Security requirements of real-system is decided by
  - Risk analysis method (ISMS, ISO15408)
  - Law, industrial standard.
  - Example:
    - Public key certificate must be valid from one to five years
    - Hash value in Cookie must be valid only in one session
    - Digital Signature must be valid for seven years (HIPPA)
- Requirements is represented as valid period.
- Standards for government use requests provable security for signature and encryption.

# Real system vs. Cryptographic Hash

|                    | Valid Period  | Rigorous Security  |
|--------------------|---|--|
| Real System        | <ul style="list-style-type: none"> <li>Defined according to system requirements</li> <li>Long to short</li> </ul> | <ul style="list-style-type: none"> <li>Not defined</li> <li>Some application need this</li> </ul>                        |
| Cryptographic Hash | <ul style="list-style-type: none"> <li>No criteria</li> </ul>   | <ul style="list-style-type: none"> <li>Rigorous security definitions</li> <li>No provable secure construction</li> </ul> |

Gaps



Quantitative security



Add provable security



# Quantitative Security

---

- Classifies security parameter from valid period
- Stronger class helps constructing systems assure security for long years.
  - Time stamping, notary, contract...
- Weaker class is for security for short period
  - sufficient for light weight use (Authentication protocols, key exchange...)
  - short hash is needed (Packet Sampling/ filtering, low-power devices, ...)
- Proposed classes:
  - Long Term Security
  - Medium Term Security
  - Short Term Security

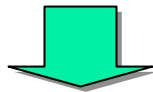


# Quantitative Classification

| Class       | Period            | Security Parameter(example) | Usages                                    |
|-------------|-------------------|-----------------------------|---|
| Long-Term   | Over 5 years      | $2^{128}$                   | Certification<br>Secure E-mail            |
| Medium-Term | 1 month - 5 years | $2^{80}$                    | PKI                                       |
| Short-Term  | Under 1 month     | $2^{64}$                    | Secure<br>Communication<br>Authentication |

# Adding Rigorous Security

- Collision resistant is sufficient for most usages
- Some applications require rigorous security
  - Digital signatures for government PKI, time-stamping etc. must be provable secure scheme.
  - Hash functions for such signature scheme should aware provable security.



Hash standard should add provable secure hash class to conventional collision resistant hash class.

# Qualitative classification

|                      | CRHF  | UOWHF  |
|----------------------|---|--|
| Key                  | No  | Length grows with the message size   |
| Adversary goal       | Find $x, y \in D (x \neq y)$<br>s.t. $h(x) = h(y)$      | Choose $x \in D$<br>Given $h_K \in H$<br>Find $y \in D (x \neq y)$ s.t. $h_K(x) = h_K(x')$ |
| Compression function | Dedicated functions<br>Block cipher based<br>Arithmetic | Strongly universal functions   |
| Construction methods | Markle-Damgaard<br>Tree                                 | XOR linear<br>XOR tree<br>Shoup (extended Markle-Damgaard)                                 |
| Standard             | ISO 10118-3   | No   |



# New classification

---

- To cover from cryptographic strong class to light weight and practically secure class...
  - New classification must contain quantitative index as well as qualitative index. (From short-term to long-term)
  - Qualitative index must cover strong class to light and practical security.
  - New classification will become 2 dimensional matrix.

# Mapping of usages to new classification

## Qualitative security

Quantitative security

|             | CRHF   | UOWHF  |
|-------------|--|--|
| Long-term   | Certification<br>(Time-stamping by hash)<br>Integrity check<br>(Software download)   | Certification<br>(Time-stamping by signature,<br>Code-signing)<br>Secure E-mail<br>(S/MIME, PGP) |
| Medium-term | N/A  | Certification<br>(PKIX)  |
| Short-term  | Secure Communication<br>(IPSEC, SSL/TLS, SSH)<br>Authentication<br>(IEEE 802.1X-EAP, Kerberos, APOP,<br>DKIM)<br>Others<br>(Packet Sampling/filtering) | N/A  |

# 4 types of hash functions

Future Standard for Hash function should consider...

|             | CRHF   | UOWHF  |
|-------------|--|--|
| Long-term   | Certification<br>(Time-stamping by hash)<br>Integrity check<br>(Software download)   | Certification<br>(Time-stamping by signature,<br>Code-signing)<br>Secure E-mail<br>(S/MIME, PGP) |
| Medium-term | N/A  | Certification<br>(PKIX)  |
| Short-term  | Secure Communication<br>(IPSEC, SSL/TLS, SSH)<br>Authentication<br>(IEEE 802.1X-EAP, Kerberos, APOP,<br>DKIM)<br>Others<br>(Packet Sampling/filtering) | N/A  |

*Type 2* (circled in red) is associated with the CRHF Long-term cell.

*Type 1* (circled in red) is associated with the UOWHF Long-term cell.

*Type 4* (circled in red) is associated with the CRHF Short-term cell.

*Type 3* (circled in red) is associated with the UOWHF Medium-term cell.



## Other issues

---

- Interoperability with existing systems
  - Length of hash value
  - Affections are not limited into crypto protocol.
  - data structure of communication, database and so on
- Implementation for embedded hardware
  - Smartcard is key device for secure services.
  - Few smartcard implements SHA-2 family
  - We need secure hash for smartcard



# Conclusions

---

- Survey of
  - Existing cryptographic security
  - Current usage of hash functions
- Pointed out gap between both security
- Proposed new classification for future hash functions