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

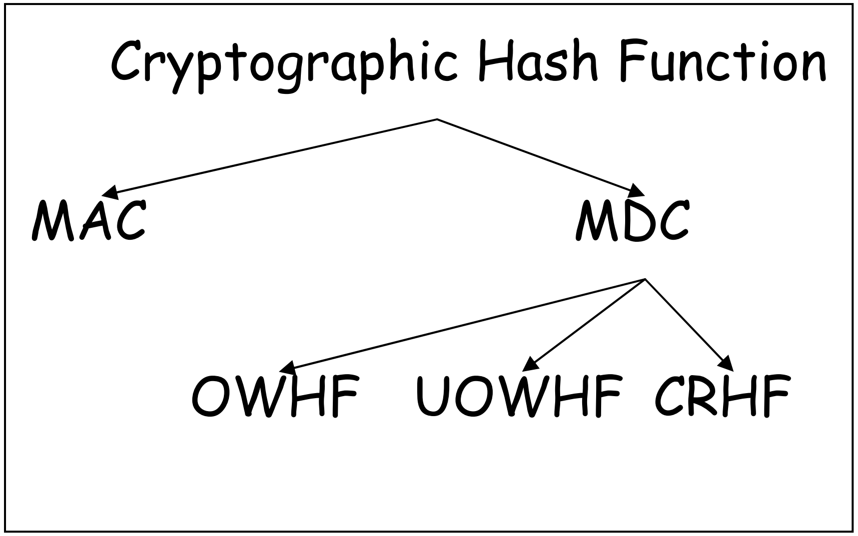
- Hash Functions
  - widely used in many information systems.
  - their security got attention after Aug. 2004.
- Security?
  - There are cryptographic definitions and evaluations.
    - Collision resistant, 2nd pre-image resistant …
  - not easy to apply them for designing real secure systems
    - "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 $h : \{0,1\}^* \to \{0,1\}^n$
- 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

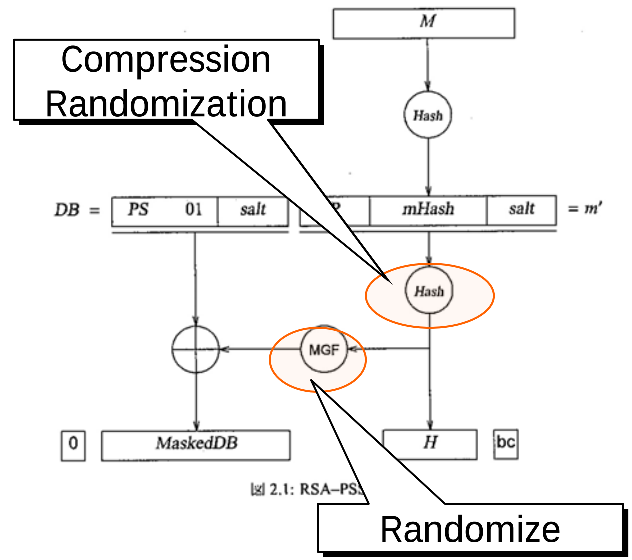
- Hash functions are widely used in real systems
  - For securing information systems
  - Cryptographic algorithm
  - Cryptographic protocols
- Many of them are built into hardware/ software products
  - Good news
    - System designer can easily construct secure system.
  - Bad news
    - In some cases, 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	▪Defined according to system requirements ▪Long to short	▪Not defined ▪Some application need this
Cryptographic Hash	▪No criteria	▪Rigorous security definitions ▪No provable secure construction

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 Secure E-mail
Medium-Term	1 month - 5 years	2^{80}	PKI
Short-Term	Under 1 month	2^{64}	Secure Communication 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 $x, y \in D (x \neq y)$ s.t. $h(x) = h(y)$	Choose $x \in D$ Given $h_K \in H$ Find $y \in D (x \neq y)$ s.t. $h_K(x) = h_K(x')$
Compression function	Dedicated functions Block cipher based Arithmetic	Strongly universal functions
Construction methods	Markle-Damgaard Tree	XOR linear XOR tree 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 (Time-stamping by hash) Integrity check (Software download)	Certification (Time-stamping by signature, Code-singing) Secure E-mail (S/MIME, PGP)
Medium-term	N/A	Certification (PKIX)
Short-term	Secure Communication (IPSEC, SSL/TLS, SSH) Authentication (IEEE 802.1X-EAP, Kerberos, APOP, DKIM) Others (Packet Sampling/filtering)	N/A

4 types of hash functions

Future Standard for Hash function should consider…

	CRHF	UOWHF
Long-term	Type 2 Certification (Time-stamping by hash) Integrity check (Software download)	Type 1 Certification (Time-stamping by signature, Code-singing) Secure E-mail (S/MIME, PGP)
Medium-term	N/A	Certification (PKIX) Type 3
Short-term	Type 4 Secure Communication (IPSEC, SSL/TLS, SSH) Authentication (IEEE 802.1X-EAP, Kerberos, APOP, DKIM) Others (Packet Sampling/filtering)	N/A

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