Outline

- Definitions (review)
- Iterated constructions (review)
- Hash functions based on block ciphers
- Hash functions based on modular arithmetic
- Conclusions

Hash functions (1)

- are secure; they can be reduced to 2 classes based on linear transformations of variables. The properties of these 12 schemes with respect to weaknesses of the underlying block cipher are studied. The same approach can be extended to study keyed hash functions (MACs) based on block ciphers and hash functions based on modular arithmetic. Finally a new attack is presented on a scheme suggested by R. Merkle. This slide is now shown at the VI Spanish meeting on Information Security and Cryptology in a presentation on the state of hash functions.

Hash functions (2)

- cryptographic hash function
  - MAC
  - MDC
    - OWHF
    - CRHF

This talk: only MDCs (Manipulation Detection Codes), which are often called ‘hash functions’
Formal definitions: (2nd) preimage resistance

Notation: $\Sigma = \{0, 1\}$, $l(n) > n$

A one-way hash function $H$ is a function with domain $D = \Sigma^{l(n)}$ and range $R = \Sigma^n$ that satisfies the following conditions:

- preimage resistance: let $x$ be selected uniformly in $D$ and let $M$ be an adversary that on input $h(x)$ uses time $\leq t$ and outputs $M(h(x)) \in D$. For each adversary $M$,
  $$\Pr_{x \in D} \{h(M(h(x))) = h(x)\} < \epsilon.$$  
  Here the probability is also taken over the random choices of $M$.

- 2nd preimage resistance: let $x$ be selected uniformly in $\Sigma^{l(n)}$ and let $M'$ be an adversary that on input $x$ uses time $\leq t$ and outputs $x' \in D$ with $x' \neq x$. For each adversary $M'$,
  $$\Pr_{x \in D} \{M'(x) = h(x)\} < \epsilon.$$  
  Here the probability is taken over the random choices of $M'$.

Further generalization: Rogaway-Shrimpton, FSE 04

Consider a family of hash functions.
For (2nd) preimage resistance, one can choose the challenge $(x)$ and/or the key that selects the function $\sim 3$ flavours:

- random challenge, random key (Pre and Sec)
- random key, fixed challenge (ePre and eSec – everywhere)
- fixed key, random challenge (aPre and aSec – always)

Complex relationship (see figure on next slide).

Formal definitions: collision resistance

A collision-resistant hash function $H$ is a function family with domain $D = \Sigma^{l(n)}$ and range $R = \Sigma^n$ that satisfies the following conditions:

- (the functions $h_S$ are preimage resistant and second preimage resistant)

- collision resistance: let $F$ be a collision string finder that on input $S \in \Sigma^s$ uses time $\leq t$ and outputs either "?" or a pair $x, x' \in \Sigma^{l(n)}$ with $x' \neq x$ such that $h_S(x') = h_S(x)$. For each $F$,
  $$\Pr_S \{F(H) \neq "?"\} < \epsilon.$$  
  Here the probability is also taken over the random choices of $F$.

Relation between definitions: Rogaway-Shrimpton

![Figure 1: Summary of the relationships among seven notions of hash-function security. Solid arrows represent conventional implications, dotted arrows represent provisional implications (their strength depends on the relative size of the domain and range), and the lack of an arrow represents a separation.](image-url)
Construction (1): iterated hash function

\[
\begin{align*}
H_0 &= IV \parallel x_1 \\
H_1 &= f(H_0 \parallel 0 \parallel x_2) \\
H_2 &= f(H_1 \parallel 1 \parallel x_3) \\
H_3 &= g(H_2)
\end{align*}
\]

- \( f \) compression function/compress
- \( g \) output transformation

unambiguous padding of input to multiple of block length
divide input into blocks \( x_1, x_2, \ldots, x_t \)

Construction (2): relation between security \( f-h \)

[Damgårđ-Merkle 89]
Let \( f \) be a collision resistant function mapping \( l \) to \( n \) bits (with \( l > n \)).

- If the padding contains the length of the input string, and if \( f \) is preimage resistant, the iterated hash function \( h \) based on \( f \) will be a CRHF.
- If an unambiguous padding rule is used, the following construction will yield a CRHF (\( l-n > 1 \)):
  \[
  H_1 = f(H_0 \parallel 0 \parallel x_1) \quad \text{and} \quad H_i = f(H_{i-1} \parallel 1 \parallel x_i) \quad i = 2, 3, \ldots t.
  \]

Construction (3): relation between security \( f-h \)

[Lai-Massey 92]
Assume that the padding contains the length of the input string, and that the message \( X \) (without padding) contains at least two blocks. Then finding a second preimage for \( h \) with a fixed \( IV \) requires \( 2^n \) operations iff finding a second preimage for \( f \) with arbitrarily chosen \( H_{i-1} \) requires \( 2^n \) operations.

BUT:

- very few hash functions have a strong compression function
- very few hash functions are designed based on a strong compression function in the sense that they treat \( x_i \) and \( H_{i-1} \) in the same way.

Preservation of properties by iteration: active research area

Block Ciphers

\[ E : \{0, 1\}^n \times \{0, 1\}^k \rightarrow \{0, 1\}^n \]
\[ E_K : \{0, 1\}^n \rightarrow \{0, 1\}^n \]

- Block cipher: family of permutations in the block space
- Every key selects one permutation
- Block length \( n \Rightarrow 2^n! \approx 2^{(n-1)2^n} \) permutations
- Key length \( k \Rightarrow 2^k \) selectable permutations

<table>
<thead>
<tr>
<th></th>
<th>year</th>
<th>( n )</th>
<th>( k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>1977</td>
<td>64</td>
<td>56</td>
</tr>
<tr>
<td>3-DES</td>
<td>1978</td>
<td>64</td>
<td>112, 168</td>
</tr>
<tr>
<td>IDEA</td>
<td>1991</td>
<td>64</td>
<td>128</td>
</tr>
<tr>
<td>AES</td>
<td>1997</td>
<td>128</td>
<td>128, 192, 256</td>
</tr>
</tbody>
</table>
Hash Functions Based on Block Ciphers (1)

Why:
- trust
- reduce design, evaluation, and implementation effort
- compact implementation
- a nice research problem :-)

Why not:
- slow (key schedule)
- export restrictions
- weaknesses which are not relevant to encryption

rate = # blocks hashed per encryption

Hash Functions Based on Block Ciphers (2)

[Rabin 78]

Merkle’s meet-in-the-middle attack: (2nd) preimage takes time $2^{n/2}$
- select $2^{n/2}$ values for $(x_1, x_2)$ and compute forward $H_2^f$
- select $2^{n/2}$ values for $(x_3, x_4)$ and compute backward $H_2^b$
- by the birthday paradox you expect to find a match and hence a second preimage

extensions:
- [Quisquater 89] low memory version
- [Lai-Massey 92] if it takes $2^n$ steps to invert the compression function, finding a second preimage requires $2^{1+\frac{n+1}{2}}$ steps

Hash Functions Based on Block Ciphers (3)

assume $k = n$
single block length hash functions:
- 12 ‘secure’ schemes of rate 1; 1 in ISO/IEC 10118-1
- collision $2^n/2$, (2nd) preimage $2^n$
- not sufficient for DES, AES

extensions:
- [Quisquater 89] low memory version
- [Lai-Massey 92] if it takes $2^n$ steps to invert the compression function, finding a second preimage requires $2^{1+\frac{n+1}{2}}$ steps

Hash Functions Based on Block Ciphers (4)

12 secure constructions (2 up to affine equivalence):
- Matyas-Meyer-Oseas: $H_i = E_{H_{i-1}}(x_i) \oplus x_i$
- Miyaguchi-Preneel: $H_i = E_{H_{i-1}}(x_i) \oplus x_i \oplus H_{i-1}$
- Davies-Meyer: $H_i = E_{x_i}(H_{i-1}) \oplus H_{i-1}$

security proof in ideal cipher model
[Winternitz 82] [Black-Rogaway-Shrimpton 02]

$B_{k,n}$: set of all block ciphers with $k$-bit keys and $n$-bit blocks
the cardinality of this set is $|B_{k,n}| = \binom{2^n}{2k}$
an ideal (block) cipher is a block cipher selected according to the uniform distribution from the set $B_{k,n}$
Hash Functions Based on Block Ciphers (5)

Proofs in ideal cipher model offer protection against generic attacks. Small deviation from being ideal can result in devastating attacks on hash functions based on block ciphers.

- DES: weak and semi-weak keys
- SHA-1 (block cipher derived from SHA-1): best known attack $\approx 2^{500}$ but collisions for SHA-1 in $2^{60.x}$
- AES up to 7 rounds exhibits special structure if key is known [Knudsen-Rijmen 07]

Double Block Hash Functions (1)

Double block length hash functions with rate 1:

$$H^1_i = E_{A_i}(B^1_i) \oplus C^1_i$$

$$H^2_i = E_{A_i}(B^2_i) \oplus C^2_i$$

- $A^1_i, B^1_i, C^1_i$ binary linear combinations of $H^1_{i-1}, H^2_{i-1}, x^1_i, x^2_i$ and $x^1_i$.
- $A^2_i, B^2_i, C^2_i$ are binary linear combinations of $H^1_{i-1}, H^2_{i-1}, x^1_i, x^2_i$, and $H^1_i$.

Goal: collision $2^m$, (2nd) preimage $2^{2m}$

But:

- [Hohl+ 94]: compression function has at most security level of single length hash function
- [Knudsen-Preneel-Lai 96]: collisions in time $2^{3m/4}$ or $2^{m/2}$

Double Block Hash Functions (2)

Rate $< 1$:

- [Bracht+ (IBM) 89] MDC-4: rate 1/4

Security for DES:

<table>
<thead>
<tr>
<th></th>
<th>rate</th>
<th>collision</th>
<th>preimage</th>
<th>coll $(f)$</th>
<th>preimage $(f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDC-2</td>
<td>1/2</td>
<td>$2^{55}$</td>
<td>$2^{83}$</td>
<td>$2^{28}$</td>
<td>$2^{54}$</td>
</tr>
<tr>
<td>MDC-4</td>
<td>1/4</td>
<td>$2^{56}$</td>
<td>$2^{109}$</td>
<td>$2^{41}$</td>
<td>$2^{90}$</td>
</tr>
</tbody>
</table>

Problem: proof of security?

[Steinberger 07] (ideal cipher model):

Collisions for MDC-2 at least $2^{75}$ ($> 2^{64}$) for $k = n = 128$ (AES)

Double Block Hash Functions (3): MDC-2

$$E^R_K(X) = E_K(X) \oplus X$$

Exercise: what if swap is dropped? (hint: use multi-collisions)
Double Block Hash Functions (4): MDC-4

\[
\begin{align*}
H_{1i-1} & \quad x_i \quad H_{2i-1} \\
E_1^H & \quad E_2^H \\
E_1 & \quad E_2 \\
H_{1i} & \quad H_{2i}
\end{align*}
\]

Double Block Hash Functions (5)

double block length hash functions:
(with collision resistant compression function)

- [Merkle 89]
  - security proof \(2^{56}\) in ideal cipher model
  - rate between \(1/18\ldots 1/4\), inconvenient block sizes

- [Knudsen-Preneel 96-97]:
  - collision attack requires at least \(2^n\) (plausibility argument)
  - rate \(1/5\ (1/4)\) with 5 (8) parallel encryptions

- [Nandi+ 05]:
  - collision attack requires at least \(2^{2n/3}\)
  - rate \(1/3\) with 3 parallel encryptions

- research topic...

Double Block Hash Functions (6): Merkle

\[
E^H_K(X) = E_K(X) \oplus X
\]

\[
\begin{align*}
0||H_{1i-1} & \quad E_1^H \\
H_{2i-1}||x_i & \quad E_2^H \quad 1||H_{1i-1}
\end{align*}
\]

\[
|H_{1i}| = 55 \\
|H_{2i}| = 57 \\
|x_i| = 7
\]

Double Block Hash Functions (7): Knudsen-Preneel

**Theorem.** If \(\exists\) quaternary code \([n',k',d]\) with \(2k' > n'\), then \(\exists\) parallel hash function s.t. for the compression function, finding a collision takes \(2^{(d-1)n/2}\) operations and finding a (2nd) preimage takes \(2^{(d-1)n}\) operations.

extension: codes over \(GF(2^4)\)

<table>
<thead>
<tr>
<th>(GF(2^2))</th>
<th>(GF(2^4))</th>
<th>Collision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Rate</td>
<td>Code</td>
</tr>
<tr>
<td>[5, 3, 3]</td>
<td>1/5</td>
<td>[6, 4, 3]</td>
</tr>
<tr>
<td>[8, 5, 3]</td>
<td>1/4</td>
<td>[8, 6, 3]</td>
</tr>
<tr>
<td>[12, 9, 3]</td>
<td>1/2</td>
<td>[12, 10, 3]</td>
</tr>
<tr>
<td>[9, 5, 4]</td>
<td>1/9</td>
<td>[9, 6, 4]</td>
</tr>
<tr>
<td>[16, 12, 4]</td>
<td>1/2</td>
<td>[16, 13, 4]</td>
</tr>
</tbody>
</table>
Double Block Hash Functions (8): Knudsen-Preneel

For n-bit block cipher with n-bit keys, using [8, 5, 3] shortened Hamming Code over GF(2^2)

\[ H_1 = h^1(H_{i-1}, x_1) \]
\[ H_2 = h^2(H_{i-1}, x_{i-1}) \]
\[ H_3 = h^3(H_{i-1}, H_{i-1}) \]
\[ H_4 = h^4(H_{i-1}, H_{i-1}) \]
\[ H_5 = h^5(H_{i-1}, x_{i-1}) \]
\[ H_6 = h^6(H_{i-1}, x_{i-1} \oplus H_{i-1} \oplus H_{i-1} \oplus H_{i-1} \oplus H_{i-1} \oplus H_{i-1} \oplus x_{i-1}^2) \]
\[ H_7 = h^7(H_{i-1}^1 \oplus H_{i-1}^2 \oplus x_{i-1} \oplus x_{i-1}^2) \]
\[ H_8 = h^8(H_{i-1}^1 \oplus H_{i-1}^2 \oplus H_{i-1}^3 \oplus H_{i-1}^4 \oplus H_{i-1}^5 \oplus H_{i-1}^6 \oplus H_{i-1}^7 \oplus H_{i-1}^8) \]

Double Block Hash Functions (7): Nandi

rate 1/3 with 3 parallel encryptions

\[ H_1 = h(x_i \parallel H_{i-1}^1) \oplus h(H_{i-1}^1 \parallel H_{i-1}^2) \]
\[ H_2 = h(H_{i-1}^2 \parallel x_i) \oplus h(H_{i-1}^1 \parallel H_{i-1}^2) \]

ideal cipher model: collision attack requires at least \(2^{2n/3}\)

Single Block Hash Functions with \(k = 2n\)

[Merkle 78] hash rate \((k-n)/n\)
\[ H_i = E_{x_i \parallel H_{i-1}}(c) \]
with \(c\) a constant string

preimage resistance of compression function follows from security of block cipher

Double Block Hash Functions with \(k = 2n\) (1)

[Lai-Massey 92] Abreast Davies-Meyer with rate 1/2

\[ H_1 = E_{H_{i-1}^1 \parallel x_i}(H_{i-1}^1) \oplus H_{i-1}^1 \]
\[ H_2 = E_{x_i \parallel H_{i-1}^1}(H_{i-1}^2 \oplus c) \oplus H_{i-1}^2 \oplus c \]
for a constant \(c\)

claimed security against collision attack: \(2^n\)
(variant: Tandem Davies-Meyer)

[Nandi 05] rate 2/3
but [Knudsen-Muller 05] collisions in time \(2^{2n/3}\); also problems with truncation
Double Block Hash Functions with \( k = 2n \) (2)

[Hirose 06] hash rate: \( (k-n)/2n \)

\[
H^1_i = E_{H^2_{i-1}||x_i}(H^1_{i-1}) \oplus H^1_{i-1} \\
H^2_i = E_{H^2_{i-1}||x_i}(H^1_{i-1} \oplus c) \oplus H^1_{i-1} \oplus c
\]

for a constant \( c \)

ideal cipher model: collision attack requires at least \( 2^n \)

AES-256: collisions \( 2^{128} \), rate 1/2 with 2 parallel encryptions (only 1 key schedule)

Based on Algebraic Structures

Why:
- sometimes one can prove security reductions
- compact implementation
- fast (knapsack-type problems)

Why not:
- mathematical structure can be exploited
- slow (modular exponentiation)
- vulnerable to trapdoors

Permutation based hashing (or fixed key)

[Rogaway-Steinberger 08]:

compression functions based on permutation \( \pi \)
- \( 2n \) to \( n \) bits rate at most 1/3 for collisions in time \( 2^{n/2} \)
- \( 3n \) to \( 2n \) bits rate at most 1/5 for collisions in time \( 2^n \)
- \( mn \) to \( tn \) bits with \( k \) permutation calls (rate \( r = (m-t)/k \)):
  - collisions in \( k \cdot 2^{n(1-(m-t)/k)} \)
  - preimages in \( k \cdot 2^{n(1-(m-t)/k)} \)

hash functions: rate \( r \) \( \sim \) collisions and preimages in time \( 2^{n(1-r)} \)

Algebraic Structures: modular arithmetic

schemes with security reduction:

[Goldwasser-Micali-Rivest 84 ]

public: \( N = pq \), two random squares \( \bmod N \), \( a_0 \) and \( a_1 \)

\[
h : \{0,1\} \times Z_N^* \to Z_N^*: \ y \mapsto y^2 \cdot a_0 \cdot a_1^{-b} \mod N
\]

collision gives \((x, x')\) with \( x' \neq x \) but \( x^2 = x'^2 \mod N \), which allows for factoring \( N \)

[Damgård 87]: improved efficiency

[Gibson 91]: discrete logarithm modulo a composite

public: \( N = pq \), \( \alpha \) of order \( \lambda(n) = \lcm(p-1, q-1) \)

\[
h : \{0,1\}^* \to Z_N^*: \ x \mapsto \alpha^x \mod N
\]

if \((x, x')\) collide under \( h \), then \( x-x' = k \cdot \lambda(n) \), which allows for factoring \( N \).
Algebraic Structures: modular arithmetic

how to generate the RSA modulus?
answer: secure multi-party computation
[Boneh-Franklin 97], [Frankel-MacKenzie-Yung 98], [Gilboa 99]

security reduction to discrete log:
[Chaum+ 91], [Brands], [Bellare+ 94]

• public: prime $p$ and $t$ random elements $\alpha_i$ from $G_p$ ($\alpha_i \neq 1$), $q = (p-1)/2$
  
  $$h : \mathbb{Z}_q^t \rightarrow G_p : (x_1 \parallel x_2 \parallel \ldots \parallel x_t) \mapsto \prod_{i=1}^{t} \alpha_i^{\bar{x}_i} \quad \text{with} \quad \bar{x}_i = 1\|x_i$$

• finding a collision for $h$ results in a discrete log in $G_p$

Algebraic Structures: schemes without security reduction:

• many broken proposals, including CCITT X.509 Annex D
• most promising: ISO/IEC 10118-4:1998

MASH-1 (Modular Arithmetic Secure Hash)

$$H_i = ((x_i \oplus H_{i-1}) \mathbin{\vee} A)^2 \pmod N \oplus H_{i-1}$$

$A = 0xF00...00$

$x_i$: first 4 bits in every byte equal to 1111 (1010 in last byte)
output transformation that reduces output size to at most $n/2$

MASH-2: replace exponent 2 by $2^8 + 1$

security for $n$-bit RSA modulus:

• best known attacks: preimage in $2^{n/2}$, collision in $2^{n/4}$
• feedforward of $H_{i-1}$ essential

Algebraic Structures: Very Smooth Hash (VSH)

[Contini-Lenstra-Steinfeld 05]

public: $N = pq$, $p_1, \ldots, p_k$ primes such that $\prod_{j=1}^{k} p_j < N$

consider message $x$ of $tk$ bits $x_1 \parallel x_2 \parallel \ldots \parallel x_{tk}$
block $i$ ($k$ bits): $\bar{x}_i = x_{(i-1)k+1} \parallel x_{(i-1)k+2} \parallel \ldots \parallel x_{ik}$

$H_0 = 1$

$$f(H_{i-1}, x_i) = H_{i-1}^2 \cdot \prod_{j=1}^{k} p_j^{x_{(i-1)k+j}} \pmod N$$

collision results in smooth relation that helps to factor $N$

Algebraic Structures: Very Smooth Hash (VSH)

[Saarinen 07]
problems with partial preimage resistance

multiplicative property: $h(x) \cdot h(y) = h(x \mathbin{\vee} y) \cdot h(z) \pmod N$
if $z = x \mathbin{\wedge} y$ and $z = 0$.

$\sim$ preimage of size $\ell$ bits can be recovered in time $2^{\ell/2}$
for small input no mod $N$ reduction with high probability
if output is truncated to $\ell$ LSBs, collisions in time $2^{\ell/3}$

[Lenstra 06]: “VSH is not a hash function”
Algebraic Structures: additive knapsacks

Given a set of $n$ $l$-bit integers $\{a_1, a_2, \ldots, a_n\}$, and an $l$-bit integer $S$ find a vector $X$ with components $x_i$ equal to 0 or 1 such that

$$\sum_{i=1}^{n} a_i \cdot x_i = S \mod 2^{\ell(n)}.$$

For hashing, one needs $n > \ell(n)$.

The **good** news:
- [Impagliazzo-Naor 96]: UOWH as secure as knapsack
- [Ajtai 96], [Goldreich+ 96]: one-way and collision-resistant function if approximating the shortest vector in a lattice to polynomial factors is hard
- [Sendrier+ 07]: random matrix + structured input: syndrome decoding is hard problem

Algebraic Structures: additive knapsacks (2)

The **bad** news:
- The knapsack problem seems to be ‘too easy’ for realistic parameters (1000 vectors of 500 bits).
- LLL for $\ell(n) > 1.0629n$
- [Camion-Patarin 91] and [Patarin 93] for $n \gg \ell(n)$
- [Wagner 02] generalized birthday attack
- LASH: knapsack based on circulant matrix $C$ [Bentahar-Page-Saarinen-Smart 06]
  No security reduction, but inspired by [Goldreich+ 96]
  $$f(H_{i-1}||x_i) = (H_{i-1} \oplus x_i) + C \cdot (H_{i-1}||x_i)^T$$
  attack [Contini+08]: collisions in $2^{4n/11}$ and preimages in $2^{4n/7}$
  (for improved version with IV $\neq 0$: $2^{7n/8}$)
  Note: attacks require large memory

Algebraic Structures: multiplicative knapsacks (1)

[Tillich-Zémor 94]

Matrix product in group $SL_2(F_{2^n})$

$$A = \begin{pmatrix} X & 1 \\ 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} X & X + 1 \\ 1 & 1 \end{pmatrix}$$

$$\pi\{0,1\} \rightarrow \{A,B\}; 0 \leftrightarrow A, 1 \leftrightarrow B$$

$$h(x_1x_2\ldots x_n) = \pi(x_1) \cdot \pi(x_2) \ldots \pi(x_n)$$

Evaluation:
- Proof that two colliding messages have ‘large’ Hamming distance
- Parallelism
- New attacks using algebraic structure for specific field polynomials (but not general attacks yet)

Algebraic Structures: multiplicative knapsacks (2)

[Lauter-Charles-Goren 07]

Matrix product in group $PSL_2(F_p)$ (identify $A$ with $-A$)

Generator set $S$ of size $t = \ell + 1$ (say 6)

Theory based on expander graphs
- No small cycles
- Close to uniform distribution when length of random walk is logarithmic in the number of vertices

[Tillich-Zémor 08]

Collisions in seconds for 1024-bit prime by lifting problem to $\mathbb{Z}[i]$
Incremental hashing

**Incrementality** [Bellare+ 94]
Given $x$ and $h(x)$, if a small modification is made to $x$, resulting in $x'$, one can update $h(x)$ in time proportional to the amount of modification between $x$ and $x'$, rather than having to recompute $h(x')$ from scratch.

[Bellare-Micciancio 97]
- hash individual blocks of message
- combine hash values with a group operation, e.g., multiplication in a group of prime order in which the discrete logarithm problem is hard

proof based on ‘random oracle’ assumption

Concluding Remarks

- we understand very little about the security of compression functions
- we know a little more about building hash functions from compression functions
- do we need a ‘small’ collision resistant compression function or should we use a ‘sponge’-type construction?
- designers have been too optimistic (over and over again. . .)
- more work should be done on other security properties: (2nd) preimage resistance, partial preimage resistance, pseudo-randomness, security with iterated applications, . . .