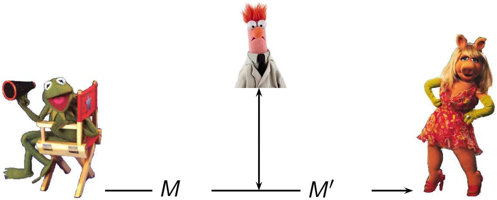
improving the security of MACs via randomized message preprocessing

Yevgeniy Dodis (New York University)
Krzysztof Pietrzak (CWI Amsterdam)

March 26, 2007

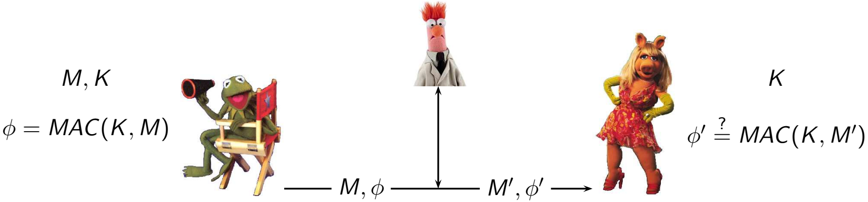
Symmetric Authentication: Message Authentication Codes

M



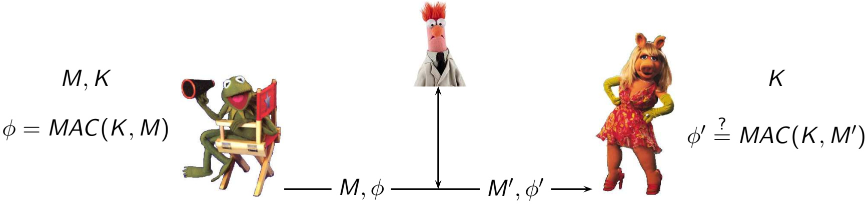
M'

Symmetric Authentication: Message Authentication Codes



- Kermit and Peggy share a secret key K.
- Kermit sends an authentication tag $\phi = MAC(K, M)$ together with message M.
- Peggy accepts M' iff $\phi' = MAC(K, M')$.

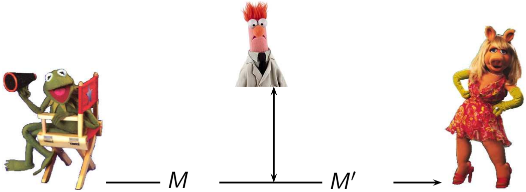
Symmetric Authentication: Message Authentication Codes



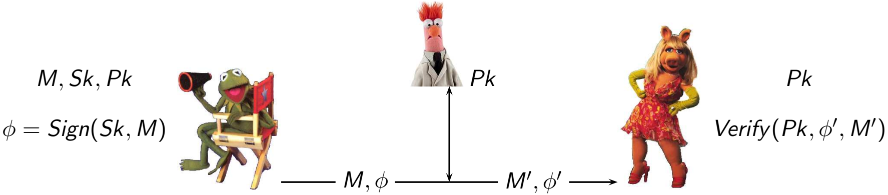
- Kermit and Peggy share a secret key K.
- Kermit sends an authentication tag $\phi = MAC(K, M)$ together with message M.
- Peggy accepts M' iff $\phi' = MAC(K, M')$.
- Security: It should be hard for Beeker (who does not know K) to come up with a pair (M', ϕ') where
  - $\phi' = MAC(K, M')$
  - Kermit did not already send (M', ϕ)

Asymmetric Authentication: Digital Signatures

M

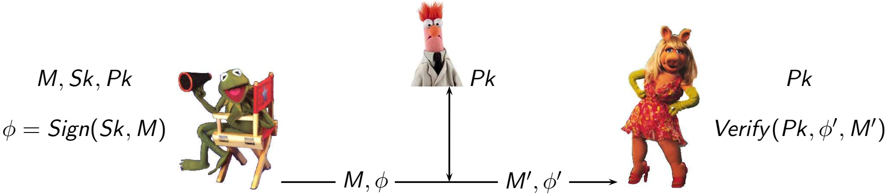


Asymmetric Authentication: Digital Signatures



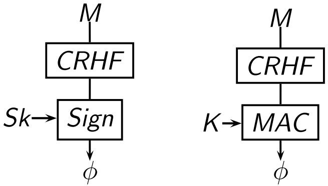
- Kermit generates a secret/public-key par Sk, Pk and send Pk to Peggy over an authentic chanell.
- Kermit sends Signature $\phi = Sign(Sk, M)$ together with message M.
- Peggy accepts M' iff $Verify(Pk, \phi', M') = accept$.

Asymmetric Authentication: Digital Signatures



- Kermit generates a secret/public-key par Sk, Pk and send Pk to Peggy over an authentic chanell.
- Kermit sends Signature $\phi = Sign(Sk, M)$ together with message M.
- Peggy accepts M' iff $Verify(Pk, \phi', M') = accept$.
- Security: It should be hard for Beeker (who does not know Sk) to come up with a pair (M', ϕ') where
  - $Verify(Pk, \phi', M') = accept$
  - Kermit did not already send (M', ϕ)

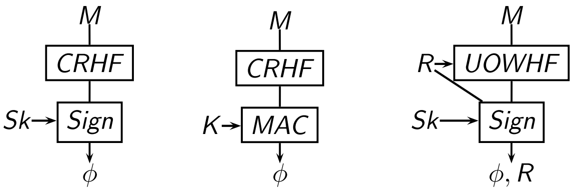
Hash then Sign/MAC/Encrypt



hash & Sign   hash & MAC

- CRHF: $Pr[A \rightarrow X, X' : H(X) = H(X')] = small$

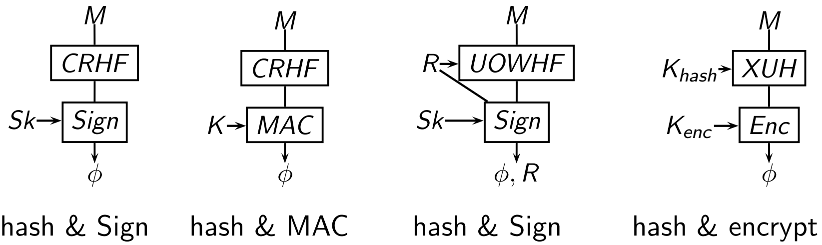
Hash then Sign/MAC/Encrypt



hash & Sign   hash & MAC   hash & Sign

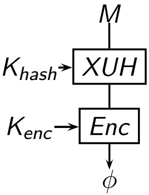
- CRHF: $Pr[A \rightarrow X, X' : H(X) = H(X')] = small$
- UOWHF: $\max_X Pr_R[A(R) \rightarrow X' : H_R(X) = H_R(X')] = small$

Hash then Sign/MAC/Encrypt

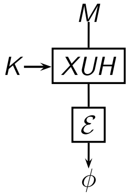


- CRHF: $Pr[A \rightarrow X, X' : H(X) = H(X')] = small$
- UOWHF: $\max_X Pr_R[A(R) \rightarrow X' : H_R(X) = H_R(X')] = small$
- ϵ-XUH: $\max_{X,X'} Pr_{K_{hash}}[H_{K_{hash}}(X) = H_{K_{hash}}(X')] \leq \epsilon$

Hash then Encrypt



Hash then Encrypt



To analyze the security we replace *Enc* with a uniformly random permutation $\mathcal{E} : \{0,1\}^k \rightarrow \{0,1\}^k$.

Sample K and $\mathcal{E}$ at random

MAC queries

Forgery queries

Beeker wins if for some j, $\phi''_j = \phi'_j$.

Theorem (security of hash then encrypt)

If H is ϵ-universal then

$$\Pr[\mathit{Beeker\ wins}] \leq \epsilon \cdot q_{mac}^2 + \epsilon \cdot q_{forge}$$

where q_{mac}/q_{forge} is the number of MAC/forgery queries.

Theorem (security of hash then encrypt)

If H is ϵ-universal then

$$\Pr[Beeker\ wins] \leq \epsilon \cdot q_{mac}^2 + \epsilon \cdot q_{forge}$$

where q_{mac}/q_{forge} is the number of MAC/forgery queries.

Proof.

$$\begin{aligned} \Pr[\text{Beeker wins}] &\leq \Pr[\text{collision}] + \Pr[\text{forgery}|\text{no collision}] \\ &\leq \epsilon \cdot q_{mac}^2 + \epsilon \cdot q_{forge} \end{aligned}$$

□

Theorem (security of hash then encrypt)

If H is ϵ-universal then

$$\Pr[Beeker\ wins] \leq \epsilon \cdot q_{mac}^2 + \epsilon \cdot q_{forge}$$

where q_{mac}/q_{forge} is the number of MAC/forgery queries.

Corollary

$q = q_{mac} + q_{forge}$

If H is $O(1/2^k)$ universal, then the security is $O(q^2/2^k)$.

If H is $O(|M|/2^k)$ universal, then the security is $O(|M|q^2/2^k)$.

Theorem (security of hash then encrypt)

If H is ϵ-universal then

$$\Pr[Beeker\ wins] \le \epsilon \cdot q_{mac}^2 + \epsilon \cdot q_{forge}$$

where q_{mac}/q_{forge} is the number of MAC/forgery queries.

Corollary

$q = q_{mac} + q_{forge}$
If H is $O(1/2^k)$ universal, then the security is $O(q^2/2^k)$.
If H is $O(|M|/2^k)$ universal, then the security is $O(|M|q^2/2^k)$.

Can we get $O(q^2/2^k)$ security using $O(|M|/2^k)$ universal hashing?

Theorem (security of hash then encrypt)

If H is ϵ-universal then

$$\Pr[\mathit{Beeker\ wins}] \leq \epsilon \cdot q_{mac}^2 + \epsilon \cdot q_{forge}$$

where q_{mac}/q_{forge} is the number of MAC/forgery queries.

Corollary

$q = q_{mac} + q_{forge}$
If H is $O(1/2^k)$ universal, then the security is $O(q^2/2^k)$.
If H is $O(|M|/2^k)$ universal, then the security is $O(|M|q^2/2^k)$.

Can we get $O(q^2/2^k)$ security using $O(|M|/2^k)$ universal hashing?
Yes, by randomizing the message

Theorem (security of hash then encrypt)

If H is ϵ-universal then

$$\Pr[Beeker\ wins] \leq \epsilon \cdot q_{mac}^2 + \epsilon \cdot q_{forge}$$

where q_{mac}/q_{forge} is the number of MAC/forgery queries.

Corollary

$q = q_{mac} + q_{forge}$
If H is $O(1/2^k)$ universal, then the security is $O(q^2/2^k)$.
If H is $O(|M|/2^k)$ universal, then the security is $O(|M|q^2/2^k)$.

Can we get $O(q^2/2^k)$ security using $O(|M|/2^k)$ universal hashing?
Yes, by randomizing the message using only $O(\log(|M|))$ random bits.

almost universal hash-functions

Definition (ϵ-universal hash function)

$H : \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is ϵ universal if

$$\forall M \neq M' \in \mathcal{M} : \Pr_{K \in \mathcal{K}}[H(K, M) = H(K, M')] \leq \epsilon$$

- $H : \mathbb{Z}_L^2 \times \mathbb{Z}_L \to \mathbb{Z}_\ell$ where $H_{x,y}(M) = (x \cdot M + y \bmod L) \bmod \ell$ is $1/\ell$ universal.
- $H : \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where $H_x(M_1, \ldots, M_d) = x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is d/ℓ-universal

the salted hash-function paradigm

A salted hash function H is $(\epsilon_{forge}, \epsilon_{mac})$ universal if

- Inputs collide with probability $\leq \epsilon_{forge}$ if salt is not random.
- Inputs collide with probability $\leq \epsilon_{mac}$ if salt is random.

Definition ($(\epsilon_{forge}, \epsilon_{mac})$-universal salted hash function)

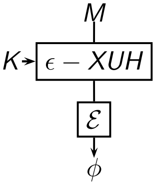
$H : \mathcal{P} \times \mathcal{K} \times \mathcal{M} \to \mathcal{T}$ is $(\epsilon_{forge}, \epsilon_{mac})$ universal if
$\forall (M, P) \neq (M', P')$:

$$\Pr_{K \in \mathcal{K},}[H(K, P, M) \neq H(K, P', M')] \leq \epsilon_{forge}$$

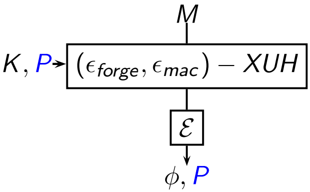
$\forall (M, M', P)$:

$$\Pr_{K \in \mathcal{K}, P' \in \mathcal{P}}[H(K, P, M) \neq H(K, P', M')] \leq \epsilon_{mac}$$

salted hash then encrypt



hash then encrypt

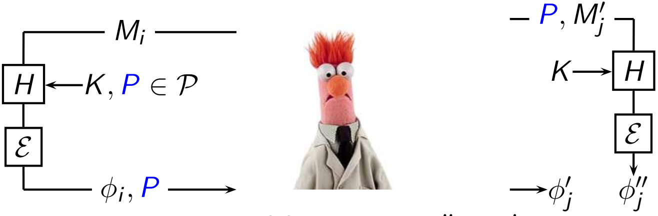


salted hash then encrypt

on each invocation a random salt P is chosen by the MAC

Sample K and $\mathcal{E}$ at random

MAC queries | Forgery queries



Beeker wins if for some j, $\phi''_j = \phi'_j$.

Theorem (security of salted hash then encrypt)

If H is $(\epsilon_{forge}, \epsilon_{mac})$-universal then

$$\Pr[Beeker\ wins] \leq \epsilon_{mac} \cdot q_{mac}^2 + \epsilon_{forge} \cdot q_{forge}$$

where q_{mac}/q_{forge} is the number of MAC/forgery queries.

Theorem (security of salted hash then encrypt)

If H is $(\epsilon_{forge}, \epsilon_{mac})$*-universal then*

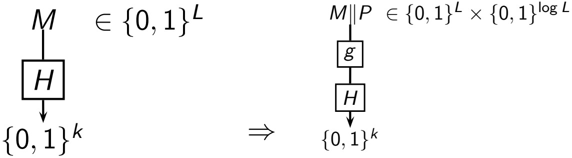
$$\Pr[Beeker\ wins] \leq \epsilon_{mac} \cdot q_{mac}^2 + \epsilon_{forge} \cdot q_{forge}$$

where q_{mac}/q_{forge} *is the number of MAC/forgery queries.*

To achieve optimal $O(q^2/2^k)$ security ($q = q_{mac} + q_{forge}$), we just need $\epsilon_{mac} \in \Theta(1/2^k)$ but ϵ_{forge} can be much bigger.

As the salt is part of the output, we want the domain $\mathcal{P}$ for the salt to be small.

the generic result, proof of concept [1]



Theorem (generic construction)

Let $H : \{0,1\}^L \rightarrow \{0,1\}^k$ *be* $L/2^k$ *universal & balanced*
$\exists$ *permutation over* $g : \{0,1\}^{L+\log(L)}$ *such that with* $P \in \{0,1\}^{\log L}$

$$H'(K, P, M) := H(K, g(M\|P))$$

is $(\epsilon_{forge}, \epsilon_{mac})$ *universal with*

$$\epsilon_{forge} = (L + \log(L))/2^k \qquad \epsilon_{mac} = 2/2^k$$

the generic result, proof of concept [2]

Generic Construction

- Optimal $\epsilon_{mac} = 2/2^k$.
- Salt of length $\log(L)$ if H is $L/2^k$ universal.
  In general: If H is $L^c/2^k$-universal, then salt will be $c \cdot \log(L)$
- Non-constructive.

a concrete example: polynomial evaluation [1]

$H : \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where
$H_x(M_1, \ldots, M_d) = x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is d/ℓ-universal

Theorem (set constant coefficient completely random)

*$H' : \mathbb{Z}_\ell \times \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where
$H'_x(P, M_1, \ldots, M_d) = P + x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is
$(\epsilon_{forge}, \epsilon_{mac})$ universal $\epsilon_{forge} = d/\ell$ and optimal $\epsilon_{mac} = 1/\ell$.*

Proof.

$H'_x(P, M) = H'_x(P', M')$ for exactly one possible $P \in \mathbb{Z}_\ell$, thus
$\epsilon_{mac} = 1/\ell$. □

a concrete example: polynomial evaluation [1]

$H : \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where
$H_x(M_1, \ldots, M_d) = x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is d/ℓ-universal

Theorem (set constant coefficient completely random)

$H' : \mathbb{Z}_\ell \times \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where
$H'_x(P, M_1, \ldots, M_d) = P + x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is
$(\epsilon_{forge}, \epsilon_{mac})$ universal $\epsilon_{forge} = d/\ell$ and optimal $\epsilon_{mac} = 1/\ell$.

Proof.

$H'_x(P, M) = H'_x(P', M')$ for exactly one possible $P \in \mathbb{Z}_\ell$, thus
$\epsilon_{mac} = 1/\ell$. □

Trivial, optimal ϵ_{mac} but $|P| = \log(\ell)$ is large.

a concrete example: polynomial evaluation [2]

$H : \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where
$H_x(M_1, \ldots, M_d) = x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is d/ℓ-universal

Theorem (choose constant coefficient from a small set $\mathcal{P}$)

*$\exists \mathcal{P} \subset \mathbb{Z}_\ell, |\mathcal{P}| = d^3$ s.t. $H' : \mathcal{P} \times \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where
$H'_x(P, M_1, \ldots, M_d) = P + x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is
$(\epsilon_{forge}, \epsilon_{mac})$ universal $\epsilon_{forge} = d/\ell$ and optimal $\epsilon_{mac} = 2/\ell$.*

a concrete example: polynomial evaluation [2]

$H : \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where
$H_x(M_1, \ldots, M_d) = x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is d/ℓ-universal

Theorem (choose constant coefficient from a small set $\mathcal{P}$)

$\exists \mathcal{P} \subset \mathbb{Z}_\ell, |\mathcal{P}| = d^3$ s.t. $H' : \mathcal{P} \times \mathbb{Z}_\ell \times \mathbb{Z}_\ell^d \to \mathbb{Z}_\ell$ where $H'_x(P, M_1, \ldots, M_d) = P + x \cdot M_1 + x^2 \cdot M_2 + \cdots + x^d \cdot M_d$ is $(\epsilon_{forge}, \epsilon_{mac})$ universal $\epsilon_{forge} = d/\ell$ and optimal $\epsilon_{mac} = 2/\ell$.

Optimal ϵ_{mac}, small $|P| = 3 \cdot \log(d)$.
No constructive way to choose $\mathcal{P}$, but choosing it at random will do with high probability.

Conclusions

- Introduced the concept of *salted* almost universal hash functions.
- Show their usefulness for hash then encrypt.
- Generic result: any XUH can be turned into a salted XUH where
  - The random salt is very short.
  - The collision probability with random salt (ϵ_{mac}) is optimal.

  Give concrete such transformations for polynomial evaluation.
- Moreover in the paper: transformation for Merkle-Damgård. Generic result for *XOR*-universal hash functions.