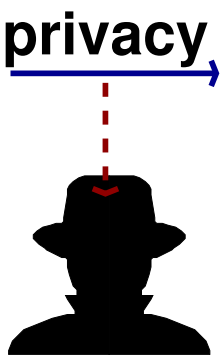
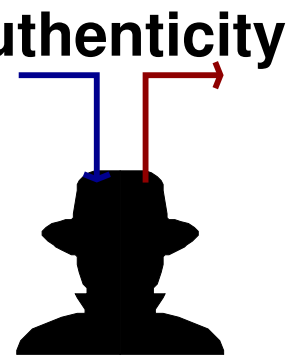
2: Authentication





Message From sender to receiver

Privacy: Eve is curious

Authenticity: Eve would like to modify a given message, or forge a fresh one.

CCA-Privacy: Eve is curious and can ask for the decryption of ciphertexts chosen by her
Is “(CPA-)Privacy + Authenticity = CCA-Privacy”?

MACs

Message Authentication Codes

- Does not consider message privacy
- Must detect forgeries/modifications of message
- For every message M_i, compute a key-dependent *message-authentication code* $A = \mathsf{MAC}_K(M_i)$
- For every pair (M, A), there is a key-dependent test to check the *validity* of (M, A)
  all pairs $(M_i, \mathsf{MAC}_K(M_i))$ are valid (others may also be)

Chosen-Message Existential Forgery

Attack Model for MACs

Chosen Message: For $i \in \{1, \ldots, q\}$:
chooses M_i, receive $A_i := \mathsf{MAC}_K(M_i)$.

Forgery: Given M, find valid (M, A).

Existential: An arbitrary $M \notin \{M_i\}$ will do.

As usual, Eve's resources are

- $\mathbf{q}$: #Chosen messages $M_1, M_2, \ldots, M_q$,
- σ: Total message length in #bits or #blocks
- $\mathbf{t}$: Eve's maximal runtime

Also important: Her success probability or rather advantage $\mathbf{a}$

UnForgeability under Chosen-Message Attack

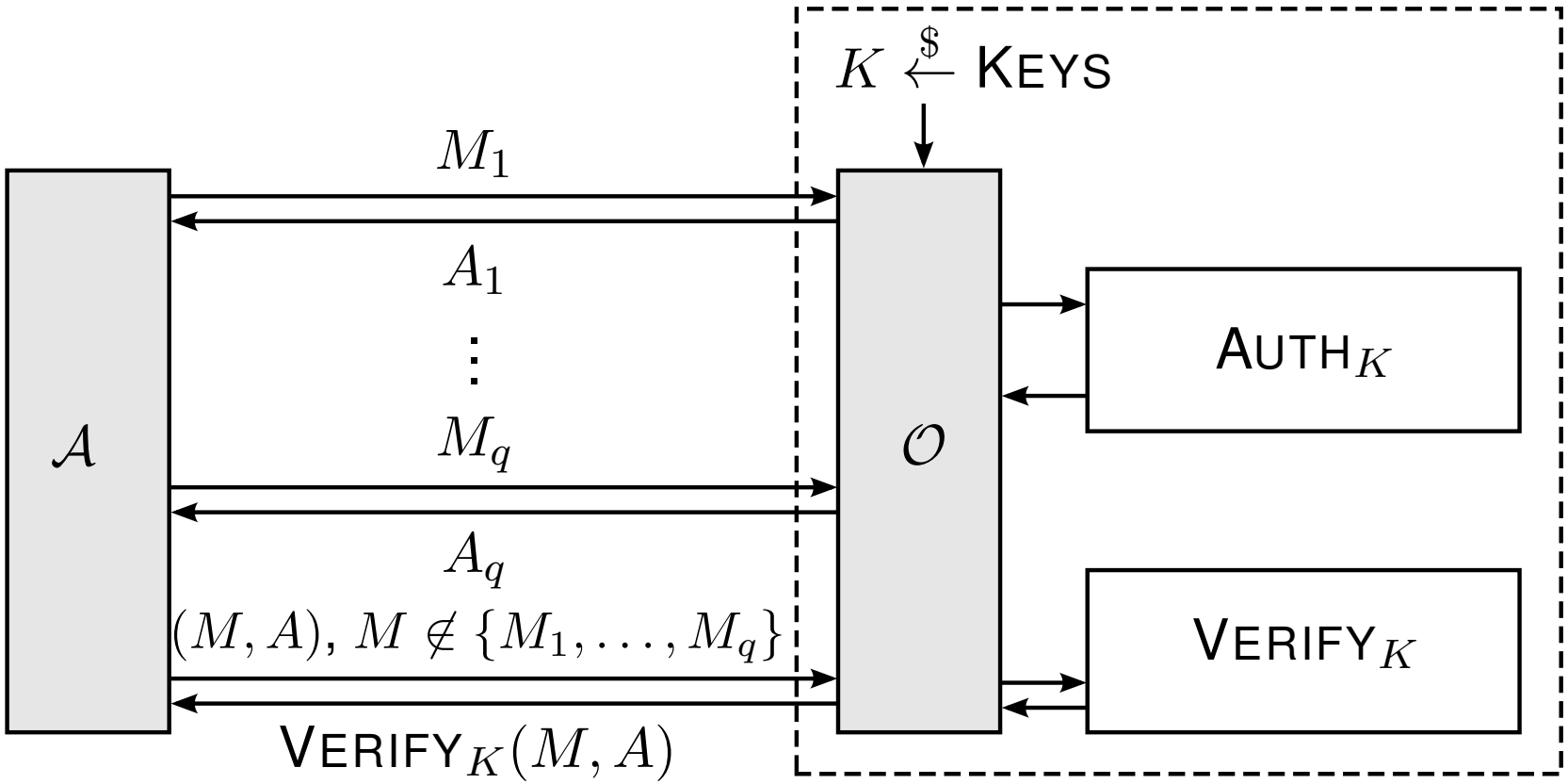
Definition 13 (UF-CMA Experiment)

1. **Querying Phase:** For $1 \leq i \leq q$:
   1. Eve chooses M_i.
   2. The oracle returns $A_i = \mathsf{MAC}_K(M_i)$.
2. **Guessing Phase:** Eve chooses (M, A), $M \notin \{M_1, \ldots, M_q\}$.
   Eve wins iff $\boxed{(M, A) \text{ is valid}}$.

Definition 14

A MAC is (t, q, σ, p)-UF-CMA-secure if any adversary that runs in time at most t and asks at most q queries of total length at most σ has a winning probability of at most p.

UnForgeability under Chosen-Message Attack



A (Pseudo-)Random Function Is a Good MAC

Theorem 15

Let $F : \{0,1\}^* \to \{0,1\}^n$ be a random function, i.e. for all $X \in \{0,1\}^*$, the values $F(X)$ are independent and uniformly distributed random values. Then, it holds that F is an $(\infty, \infty, \infty, 1/2^n)$-secure MAC.

Theorem 16

Let $P : \{0,1\}^* \to \{0,1\}^n$ be a (t, q, a)-secure pseudorandom function, i.e. a t-time adversary asking q queries cannot distinguish P from a random function with an advantage exceeding a. Then, P is a $(t, q, \infty, a + 1/2^n)$-secure MAC.

Replay-Attacks

A Very Simple Example for a Limitations of UF-CMA Security!

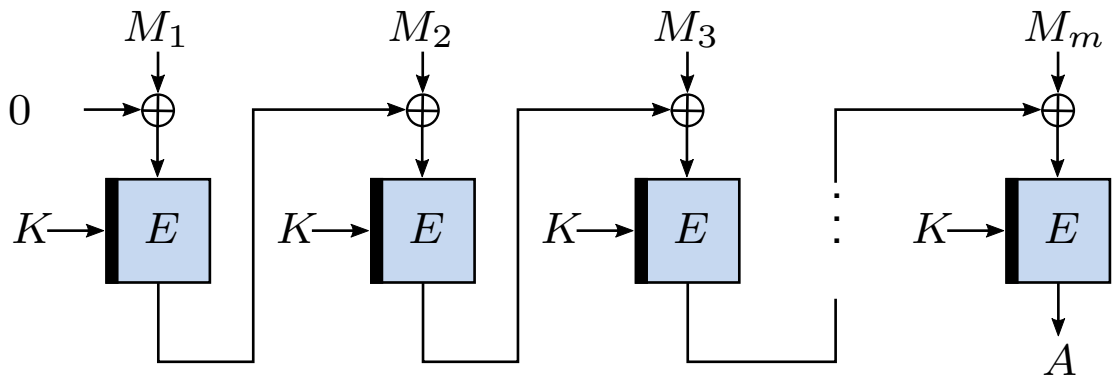
1. Sara Rich sends (M, A) to her bank.
   $M =$ *Transfer 1000$ to Anton Poor!*;
   $A = \mathsf{MAC}_K(M)$.
2. Anton's friend Eve eavesdrops (M, A).
3. The bank compares $\mathsf{MAC}_K(M)$ and A.
4. If $A = \mathsf{MAC}_K(M)$ and Sara has the money,
   then the bank transfers the money
   else **goto 6!**
5. Eve sends (M, A) to the bank; **goto 3!**
6. Now, Sara is poor and Anton is rich.

So is UF-CMA a Practical Security Definition?

- Replay attacks are a serious problem
- UF-CMA security does not deal with replay attacks
- Reason: UF-CMA does not deal with the message context
- The relevant context depends on the application, e.g., Sara might actually make the same transfer more than once (e.g., every month), and the MAC should not prevent that
- Replay attacks must be considered . . . at higher protocol levels

2.1: The CBC-MAC

- Let $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$
- Encrypt $M = (M_1, \ldots, M_m) \in (\{0,1\}^n)^m$ in CBC mode with a fixed $C_0 = 0$
- Ignore $C_0, \ldots, C_{n-1}$ and output $A = C_m$ as authentication tag



CBC-MAC$_K(M_1, \ldots, M_m)$

The Security of the CBC MAC

When all messages have the same length, the CBC-MAC is a *secure* MAC – and even a good PRF:

Theorem 17

Consider the CBC-MAC*, using a random n-bit permutation E. Let σ be the total number of blocks queried.*

Assume that each message consists of exactly m blocks (i.e., $q = \sigma/m$ queries are made).

If $\sigma \ll 2^{n/2}$, the advantage of distinguishing the CBC-MAC from a random function is negligible.

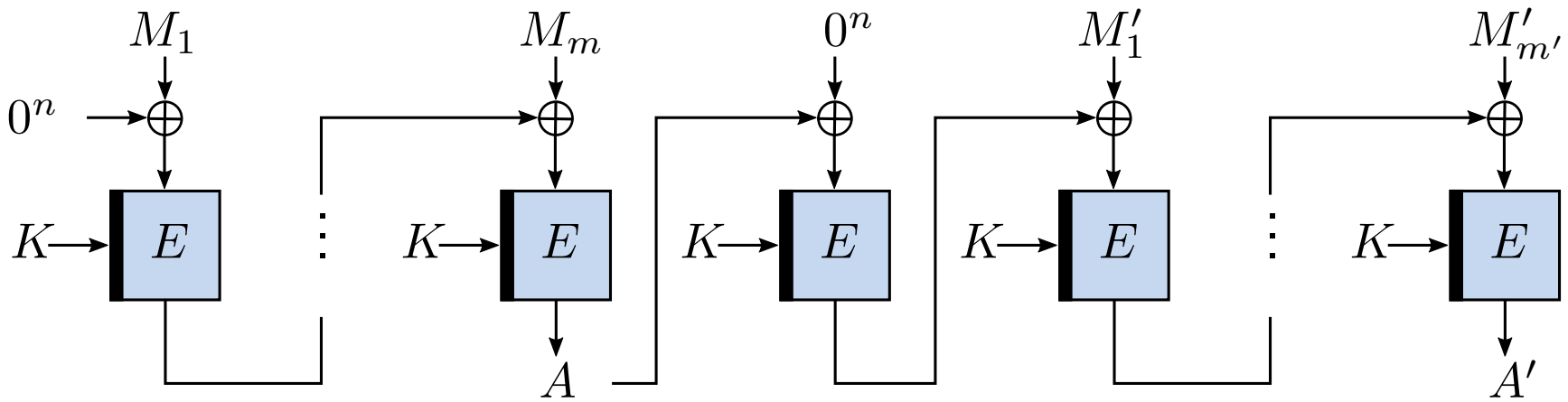
Remark

The theorem actually holds for all **prefix-free** sets of messages, that means that for all pairs of messages $(M_1, \ldots M_m)$ and $(M'_1, \ldots, M'_{m'})$, an $i \leq \min\{m, m'\}$ exists with $M_i \neq M'_i$.

Some Attacks on the CBC-MAC

Recall the Problem Session?

1. **Birthday Attack** with $\sigma \approx 2^{n/2}$.
2. **Splicing Attack** for variable-length messages:



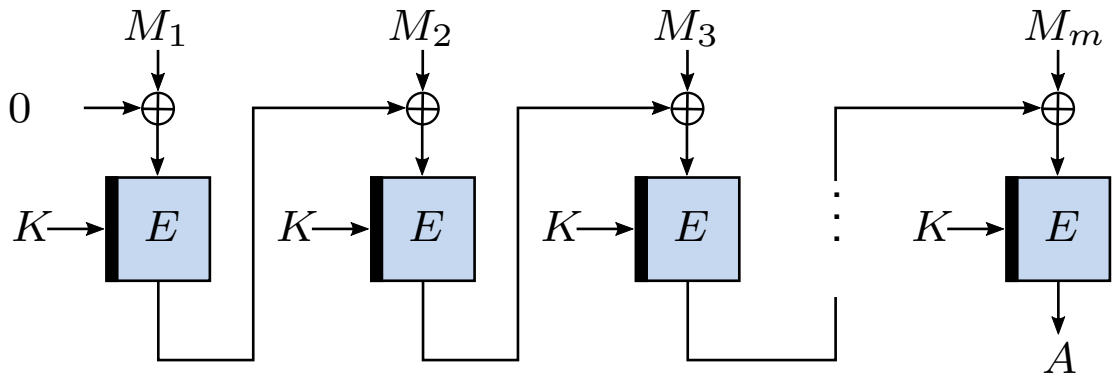
Protection for Variable-Length Messages

Either truncate tag A (e.g., use first $n/2$ bits of A, only),

or prepend the length $|M|$ to message ($M_0 := |M|$),

or encrypt tag A (under an independent key),

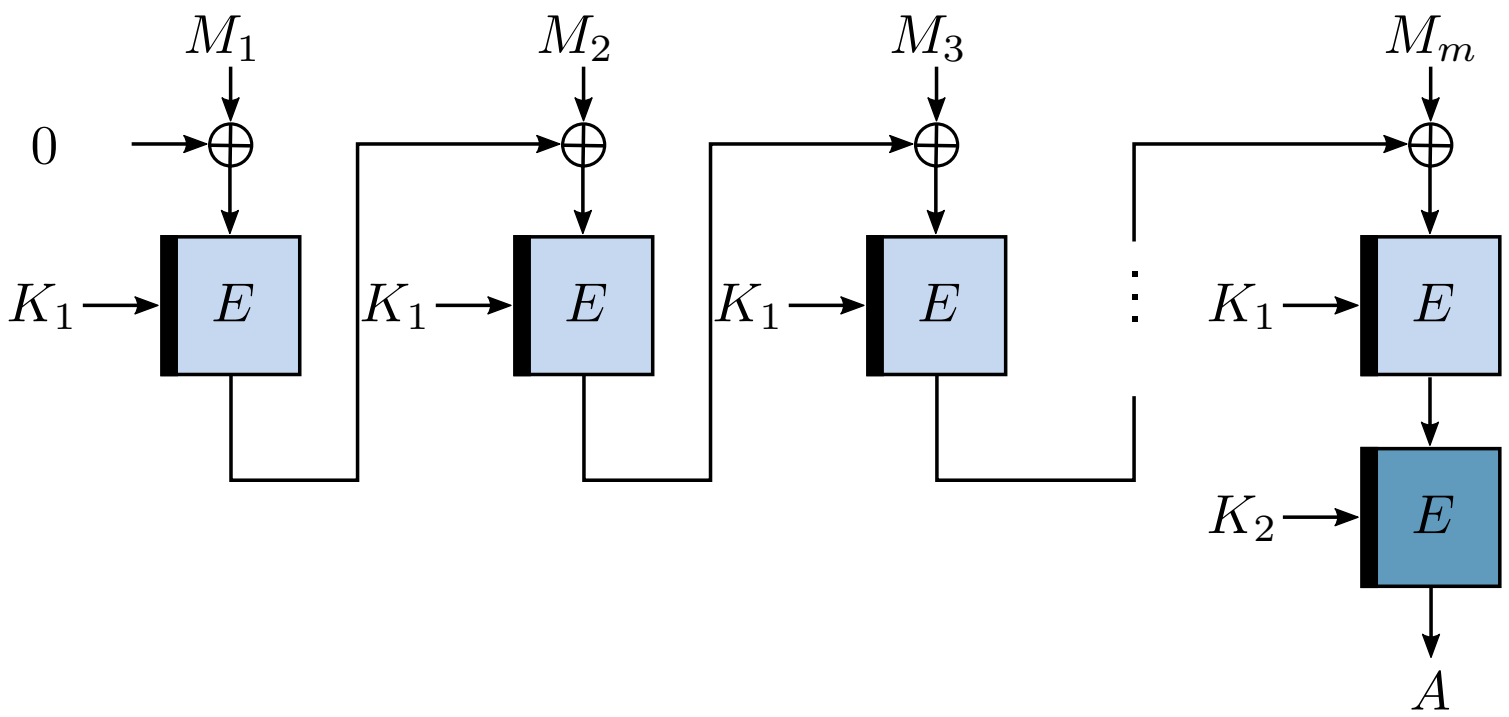
or encrypt last block M_m.



CBC-MAC$_K(M_1, \ldots, M_m)$

Encrypted CBC-MAC

Security for Variable-Length Message



$$\text{ECBC-MAC}_{K_1,K_2}(M_1,\ldots,M_m) = E_{K_2}(\text{CBC-MAC}_{K_1}(M_1,\ldots,M_m))$$

The Security of the Encrypted CBC-MAC

Theorem 18

Consider the encrypted CBC-MAC*, using a random permutation* $E_K : \{0,1\}^n \to \{0,1\}^n$*. Let* σ *denote the number of queried blocks of an adversary. If* $\sigma \ll 2^{n/2}$*, then the advantage of distinguishing* ECBC-MAC *from a random function is negligible for any adversary.*

How Secure is the Encrypted CBC-MAC Really?

- **Good:** Without either
  - recovering the key K (making $\approx 2^k$ queries to E or D)
  - or querying the MAC with $\sigma \approx O(2^{n/2})$ blocks in total,

  Eve cannot distinguish "real" from "random" with significant advantage.
- **Bad:** On the other hand, $\approx 2^k$ queries to E or $\approx 2^{n/2}$ blocks to the MAC are sufficient to gain a significant advantage.

CBC-MAC with one Single Key: CMAC

2003: Published by Iwata and Kurosawa as *One-Key MAC* (OMAC)

2005: Standardized by the NIST as *Cipher-Based MAC* (CMAC)

Approach: Encrypt the last block M_m, actually, XOR M_m to a key-dependent secret constant

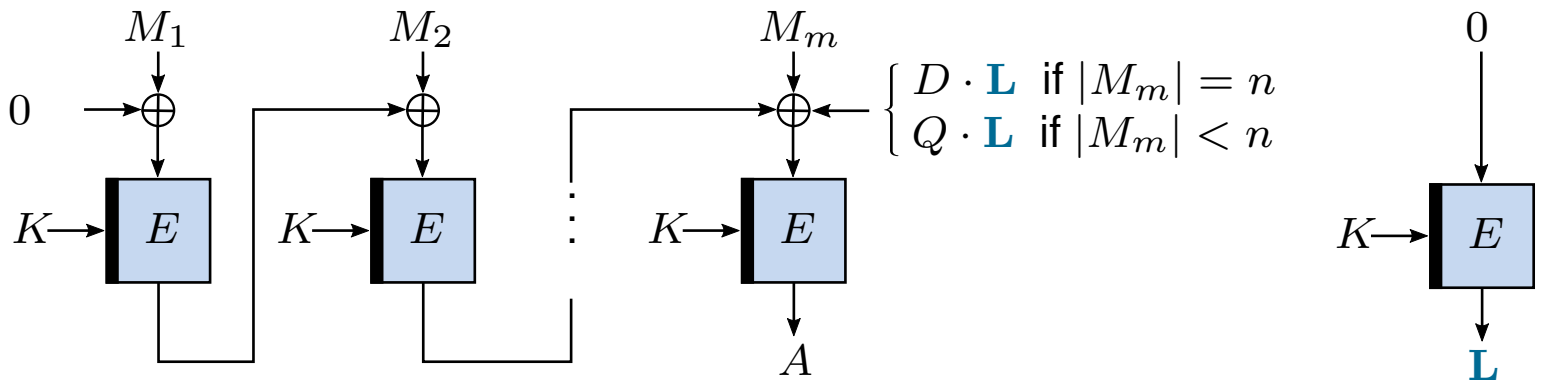
CMAC

Set $\mathbf{L} := E_K(0)$, $\mathbf{D} := 2 \cdot \mathbf{L}$; $\mathbf{Q} := 2^2 \cdot \mathbf{L}$ (multiplication in GF(2^n)), pad the final message block and XOR it to a key-dependent constant:

$$M'_m := \begin{cases} M_m \oplus D & \text{if } |M_m| = n \\ (M_m \,\|\, 10^{n-c-1}) \oplus Q & \text{if } |M_m| = c < n. \end{cases}$$

and apply the CBC-MAC:

$$\text{CMAC}_K(M_1, \ldots, M_m) := \text{CBC-MAC}_K(M_1, \ldots, M_{m-1}, M'_m).$$



Security of CMAC

Theorem 19

Consider the CMAC*, using a random permutation* $E_K : \{0,1\}^n \to \{0,1\}^n$*. If the total message length (in blocks) is* $\sigma \ll 2^{n/2}$*, the advantage of distiguishing* CMAC *from a random function* $\{0,1\}^* \to \{0,1\}^n$ *is negligible.*

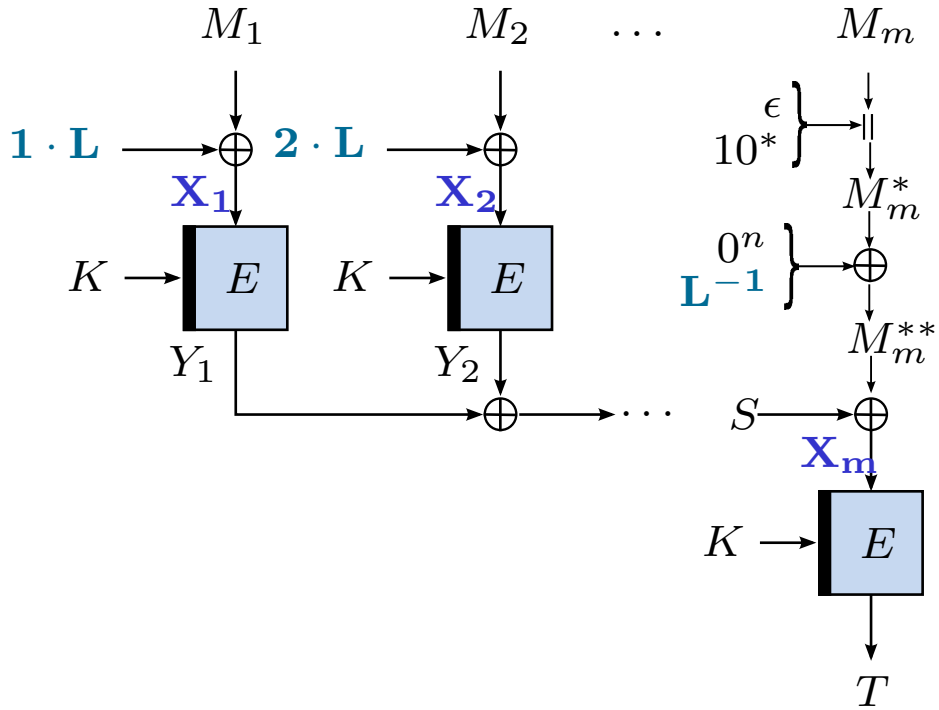
Corollary 20 (Tweakable CMAC)

Let $t \in \{0,1\}^n$*. Define* $\mathsf{CMAC}^t_K(M) := \mathsf{CMAC}_K(t \,\|\, M)$*. If* $\sigma \ll 2^{n/2}$*,* CMAC^0*,* CMAC^1*, . . . , are good independent PRFs.*

2.2: The PMAC – a Parallelizable MAC

Black and Rogaway, 2002

1: $S \leftarrow 0^n$
2: $\mathbf{L} \leftarrow E_K(0^n)$
3: $R \leftarrow L^{-1}$
4: **for** $i \leftarrow 1..m-1$ **do**
5: $\quad \mathbf{X_i} \leftarrow M_i \oplus (\mathbf{i} \cdot \mathbf{L})$
6: $\quad Y_i \leftarrow E_K(\mathbf{X_i})$
7: $\quad S \leftarrow S \oplus Y_i$
8: **if** $|M_m| < n$ **then**
9: $\quad M_m^* \leftarrow M_m \,\|\, 10^*$
10: $\quad M_m^{**} \leftarrow M_m^* \oplus \mathbf{R}$
11: **else**
12: $\quad M_m^* \leftarrow M_m$
13: $\quad M_m^{**} \leftarrow M_m^*$
14: $\mathbf{X_m} \leftarrow S \oplus M_m^*$
15: **return** $T \leftarrow E_K(\mathbf{X_m})$



Security of PMAC

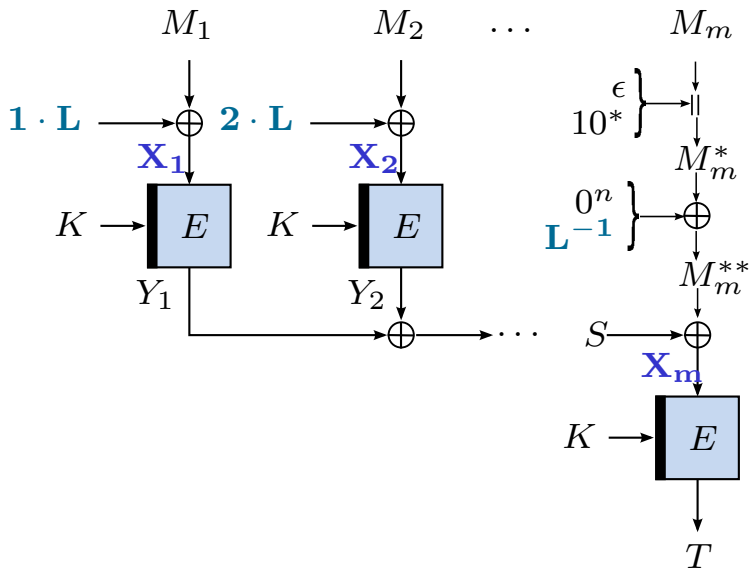
Theorem 21

Consider PMAC, using a random permutation $E_K : \{0,1\}^n \to \{0,1\}^n$. If the total message length (in blocks) is $\sigma \ll 2^{n/2}$, the advantage of distiguishing the PMAC from a random function $\{0,1\}^* \to \{0,1\}^n$ is negligible.

A Very Sketchy Security Proof for PMAC

Two Cases

- Assume $\sigma \ll 2^{n/2}$
- Replace E_K by random function (PRP-PRF switching)
- Consider two different messages $(M_1, \ldots, M_{m-1}, M_m)$ and $(M'_1, \ldots, M'_{m'-1}, M'_{m'})$
- **Case 1:** $m = m'$ and $M_i = M'_i$ for $1 \le i \le m-1$ thus $S = S'$ and $M_m \neq M'_m$
- **Case 2:** All other message pairs.
- **Prove:** $\Pr[\mathbf{X_m} = \mathbf{X'_{m'}}] \approx 0$



A (Very Sketchy) Security Proof for PMAC (2)

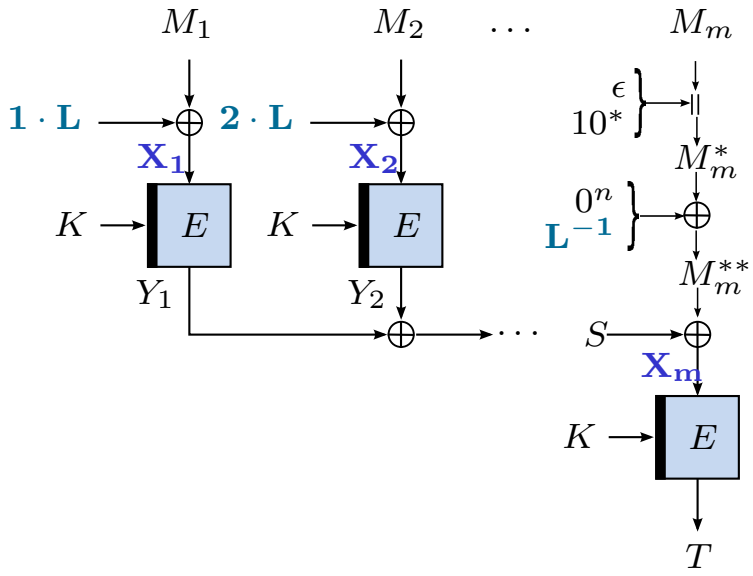
Case 1

- Remember: $S = S'$
- Both or neither padded:

$$M_m^{**} \neq M'^{**}_m$$

and thus $\mathbf{X_m \neq X'_m}$.

- One padded:



$$\mathbf{X_m = X'_m} \quad \Rightarrow \quad M_m^{**} = M'^{**}_m$$
$$\Rightarrow \quad M_m^* \oplus \mathbf{R} \neq M'^*_m$$

Exactly one $\mathbf{L = R^{-1}}$ matches.

A (Very Sketchy) Security Proof for PMAC (3)

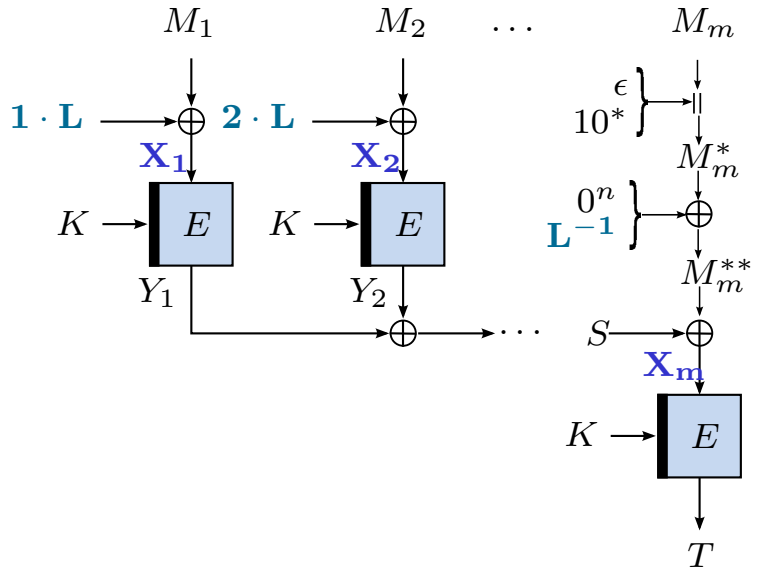
Case 2

We expect $\mathbf{X_i} \neq \mathbf{X'_j}$, except when $i = j$ and $M_i = M_j$.

- For a pair of case-2 messages, we expect

$$\{\mathbf{X_i} \mid 1 \leq i < m\} \neq \{\mathbf{X'_j} \mid 1 \leq j < m'\}$$

- Thus, there is at least one random value Y_i or Y'_j, such that either S or S' depends on that random value, but the other one doesn't.



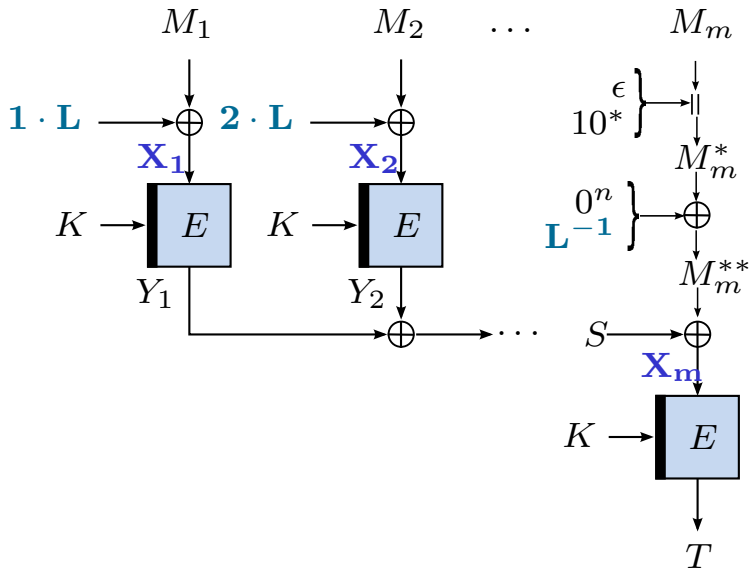
If so, we can treat S and S' as independent random values, and so are $\mathbf{X_m}$ and $\mathbf{X'_m}$. Thus

$$\Pr[\mathbf{X_m} = \mathbf{X'_{m'}}] \approx 0.$$

A (Very Sketchy) Security Proof for PMAC (4)

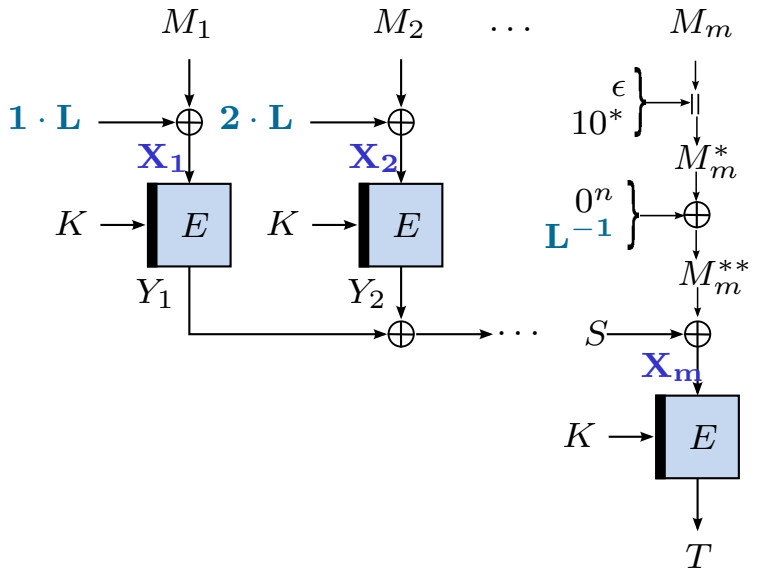
Conclusion

For any two messages M and M', we have $\mathbf{X_m} \neq \mathbf{X'_{m'}}$ with high probability. Then, their authentication tags are independent random values since we model E_K as a random function. □



PMAC in Practice

- We cheated a little when describing PMAC
- Where I wrote $\mathbf{i} \cdot \mathbf{L}$, PMAC actually computes $\mathbf{g_i} \cdot \mathbf{L}$.
- $\mathbf{g_i}$ is the i-th Gray-code word.
- For all i, the Hamming distance between g_i and g_{i-1} is 1.
- This simplifies the computation of $\mathbf{g_i} \cdot \mathbf{L}$ from $\mathbf{g_{i-1}} \cdot \mathbf{L}$ and significantly improves the efficiency
- Nevertheless, a sequential implementation of PMAC is slightly slower than CMAC.
- But PMAC can be arbitrarily parallelized, much unlike CMAC.



2.3: MACs Based on Universal Hash Functions

Consider the following scheme:

- Public prime p
- Secret key (b, c) with $b \in \mathbb{Z}_p^*$ and $c \in \mathbb{Z}_p$
- Message $x \in \mathbb{Z}_p$
- Authentication tag $a \in \mathbb{Z}_p$:

$$a = \mathsf{MAC}_{b,c}(x) = h(x) = bx + c \bmod p.$$

How secure is this scheme in the UF-CMA sense?

(Consider $q \in \{0, 1, 2, 3, \ldots\}$ oracle queries.)

Why Are Our Computations Modulo p?

Recall the computation

$$a = \mathsf{MAC}_{b,c}(x) = h_{b,c}(x) = bx + c \bmod p.$$

1. Why must p be a prime? Wouldn't $p = 2^w$ be much nicer?
2. Or could we replace arithmetic in the finite field $\mathbb{Z}_p$ by computations in any other finite field? Maybe even GF(2^w)?

Authenticating $\leq q$ Messages

- Public prime p
- Secret key $K = (b, c_1, \ldots c_q) \in (\mathbb{Z}_p^*)^{q+1}$
- Message counter i
- Messages $x_i \in \mathbb{Z}_p$
- Authentication tag $a_i \in \mathbb{Z}_p$:

$$a_i = \mathsf{MAC}_K(x_i, i) = h_{b,c_i}(x_i) = bx_i + c_i \bmod p.$$

Definition 22 (UF-CMA with Message Counter)

1. **Query Phase:** For $1 \leq i \leq q$:
   1. Eve sends x_i.
   2. The oracle returns $a_i = \mathsf{MAC}_K(x_i, i)$.
2. **Guessing Phase:** Eve chooses (x, a, j) with $x \notin \{x_1, \ldots, x_q\}$ and $j \in \{1, \ldots, q\}$.
   Eve wins iff $\boxed{(x, a, j) \text{ is valid}}$.

Theorem 23 ($bx_i + c_i \bmod p$)

Eve's success probability is at most

$$\frac{1}{p-1}.$$

Authenticating "Almost Any Number" of Messages

- Public prime p
- Secret key $b \in \mathbb{Z}_p$
- Message counter i
- Secret random function $F : \mathbb{N} \to \mathbb{Z}_p$
  (e.g., instantiated using block cipher E and secret K)
- Messages $x_i \in \mathbb{Z}_p$
- Authentication tags (a_i, b_i):

$$a_i = bx_i + F(i).$$

We will discuss the security of this approach later.

2.4: Universal Hash Functions

Definition 24 (Universal Hash Functions)

Consider the finite set of inputs M, of outputs A, and a set H of hash functions $M \to A$. Randomly choose $h \in H$.

H is **ϵ-almost-universal**, (ϵ-AU) if for all $m_1 \neq m_2 \in M$ holds

$$\Pr[h(m_1) = h(m_2)] \leq \epsilon.$$

H is **ϵ-almost-Δ-universal** (ϵ-AΔU) if for all $m_1 \neq m_2 \in M$, $d \in A$ holds

$$\Pr[h(m_1) - h(m_2) = d] \leq \epsilon.$$

H is **ϵ-almost-strongly-universal** (ϵ-ASU), if for all $m_1 \neq m_2 \in M$ and $a_1, a_2 \in A$ holds

$$\Pr[h(m_1) = a_1 \text{ und } h(m_2) = a_2] \leq \epsilon.$$

Universal: Carter, Wegman 1979; Stinson 1992.
Almost-Δ-Universal: Krawczyk 1994; Stinson 1995.
Almost-Strongly-Universal: Wegman, Carter 1981; Stinson 1992.

Fundamental Properties of Universal Hash Functions

Theorem 25

Let H be a nonempty set of functions $h : M \to A$.

1. *H is 1-AU, 1-AΔU and 1-ASU.*
2. *If H is $(\epsilon/|A|)$-ASU then it is also ϵ-AΔU.*
3. *If H is ϵ-AΔU, then it is also ϵ-AU.*
4. *H is at most $(1/|A|)$-AΔU.*
5. *H is at most $(1/|A|^2)$-ASU.*

Examples

Let p be a prime, and $b \xleftarrow{\$} \mathbb{Z}_p^*$ and $c \xleftarrow{\$} \mathbb{Z}_p$ independent secret keys.

1. H_1 is a set of functions $\mathbf{h(x) = c \bmod p}$
   H_1 is not ϵ-AU for any $\epsilon < 1$

2. H_2 is a set of functions $\mathbf{h(x) = bx \bmod p}$
   H_2 is ϵ-AΔU with
   $$\epsilon \leq \frac{1}{p-1}.$$
   H_2 is not ϵ'-ASU for any $\epsilon' < \frac{1}{p-1}$

3. H_3 is a set of functions $\mathbf{h(x) = bx + c \bmod p}$
   H_3 is ϵ-ASU with
   $$\epsilon \leq \frac{1}{p(p-1)}.$$

Combination of Almost-Universal Hash Functions

Theorem 26 (Combination of Almost-Universal Sets of Functions)

Let $F = \{f : M \to A_f\}$ be an ϵ_f-AU set of functions.
Let $G = \{g : M \to A_g\}$ be an ϵ_g-AU set of functions.

Then there exists a set H_1 of functions $h : M \to (A_f \times A_g)$ such that

- *H_1 is $(\epsilon_f \cdot \epsilon_g)$-AU*
- *and $|H_1| = |F| \cdot |G|$.*

Let $M = A_f$. Then, there exists also a set H_2 of functions $h : M \to A_g$ s. t.

- *H_2 is $(\epsilon_f + \epsilon_g)$-AU*
- *and $|H_2| = |F| \cdot |G|$.*

From Universal Hashing to Information-Theoretically Secure MACs

- An $(\epsilon/|A|)$-ASU set of functions H can directly be used as a MAC, **if one chooses a new key $h \in$H for each new message.** Eve succeeds with probability at most ϵ.
- Alternative: **Wegman-Carter Construction (W-C)**:
  - H: set of functions $h : M \to A$
  - Random values $c_1, \ldots, c_q \in A$
  - Random $h \xleftarrow{\$} H$
  - The authentication tag for the i-th message m_i is computed as

$$a_i = h(m_i) + c_i$$

(here "$+$" represents a group operation over A)

We will study this construction in the UF-CMA scenario with a message counter.

Information-Theoretical Security of the Wegman-Carter Construction

Theorem 27 (ϵ-AΔU Suffices for the Security of W-C)

If H is ϵ-AΔU, an adversary can succeed with probability at most ϵ.

Theorem 28 (ϵ-AU Is Insufficient)

Assume that H is only ϵ-AU and p-AΔU for some $p > \epsilon$. This means, there exist $m_1 \neq m_2 \in M$ and $d \in A$ with $\Pr[h(m_1) - h(m_2) = d] \geq p$. Then, an adversary can succeed with probability $p > \epsilon$.

Complexity-Theoretically Secure MACs

- The Wegman-Carter construction is essentially the "encryption of $h(m_i)$ using a one-time-pad" $\Longrightarrow$ each new message m_i needs a new key c_i.
- We can just as well apply "conventional" encryption, at the cost of abandoning information-theoretical security.
- Frequently used (m_i, i, a_i) with $a_i = h(m_i) + F_K(i)$, where
  - h is from an ϵ-AΔU set of functions H
  - F_K is a pseudorandom function depending on a secret key K.
- Note that ϵ-AU is insufficient for H!

Why Use Such Complexity-Theory-Secure MACs at All?

- A random $h \in H$ possesses properties that do not depend on any (unproven) assumption
- On many platforms, $h \in H$ can run much faster (or with much less chip space) than "classical" cryptographic functions.
- We no longer need a key whose length is proportional to the number of messages we authenticate.
- Since F_K is called only once for each (potentially long) message, one may employ a function with a very high security margin.

2.5: Polynomial Hashes

- Finite field $\mathbb{F}$
- Set of keys: $K \subseteq \mathbb{F}$
- Secret key $k \in K$
- Fixed message length ℓ blocks
- Message $m_1, \ldots, m_\ell$

$$H_k(m_1, \ldots, m_\ell) := m_1 \cdot k^1 + m_2 \cdot k^2 + \cdots + m_\ell \cdot k^\ell = \sum_{1 \le i \le \ell} m_i \cdot k^i.$$

- "Horner Scheme": (n multiplications and $n-1$ additions in $\mathbb{F}$)

1: $y \leftarrow m_\ell \cdot k$
2: **for** $i \leftarrow \ell - 1$ down to 1 **do**
3: $\quad y \leftarrow k \cdot (y + m_i)$
4: (Invariant: $y = km_i + k^2 m_{i+1} + \ldots + k^{\ell-i+1} m_\ell$)
5: **return** y

The Fundamental Theorem of Polynomial Hashing

Theorem 29

A polynomial hash H is ϵ-AΔU with

$$\epsilon \leq \frac{n}{|K|}.$$

- Note: H is completely insecure if messages of different lengths are allowed:

$$H(m_1, \ldots, m_n) := H(m_1 \ldots, m_n, 0).$$

- Handling messages of different lengths needs additional precautions.
- Even if done so, the *maximum* message length ℓ is a crucial security parameter. As it will turn out soon, the $\leq$-bound above is actually tight ($\rightarrow$ "polynomial attack").

GHASH

McGrew, Viega, 2004

- Field GF(2^n)
- Define: A family of hash functions H is called
  ϵ-almost-XOR-universal (ϵ-AXU) if for $h \xleftarrow{\$} H$ and all m_1, m_2, d

$$\Pr[h(m_1) \oplus h(m_2) = d] \leq \epsilon$$

  (renaming ϵ-AΔU for the special case of GF(2^n))
- Secret key $H \in$ GF(2^n), fixed message length ℓ:

$$\text{GHASH}_H(m_1, \ldots, m_\ell) = \bigoplus_{1 \leq i \leq \ell} m_i H^i$$

- Fundamental Theorem: GHASH is ϵ-AXU with $\epsilon \leq \ell / 2^n$

PolyP

Krovetz, Rogaway, 2000

- Finite field $\mathbb{Z}_p$
- Subset $K \subseteq \mathbb{Z}_p$.
- For random $k \xleftarrow{\$} K$, PolyP$_k$ is defined as

$$\text{PolyP}_k(m_1, \ldots, m_\ell) = m_1 k^1 + m_2 k^2 + \cdots + m_\ell k^\ell = \sum_{1 \le i \le \ell} m_i k^i.$$

- Fundamental Theorem: PolyP is ϵ-AΔU with

$$\epsilon \le \frac{\ell}{|K|}.$$

Messages of Different Lengths

PolyQ

- Fix a maximum message length ℓ.
- Choose a key $k \xleftarrow{\$} K \subseteq \mathbb{Z}_p$.
- For a message of length $m \leq \ell' \leq \ell$, PolyQ_k is defined as follows:

$$\begin{aligned} \mathsf{PolyQ}_k(m_1, \ldots, m_{\ell'}) &= \mathsf{PolyP}_k(m_1, \ldots, m_{\ell'}, \mathbf{1}) \\ &= \mathbf{b}^{\ell+1} + \mathsf{PolyP}_k(m_1, \ldots, m_{\ell'}) \end{aligned}$$

Theorem 30

If messages consist of at most ℓ values from $\mathbb{Z}_p$, PolyQ is ϵ-AΔU with

$$\epsilon \leq \frac{\ell + 1}{|K|}.$$

"Good" Finite Fields for PolyQ

PolyQ w; p = Largest w-Bit Prime

- PolyQ32: $p = 2^{32} - 5$; observe

$$\begin{aligned} 2^{32} &\equiv 5 && \pmod p \\ 2^{33} &\equiv 10 && \pmod p \\ &\vdots \\ D \cdot 2^{32} &\equiv 5D && \pmod p \end{aligned}$$

  - Restrict key k by $k \in K = \{0, \ldots 2^{29} - 1\}$.
    If the message is at most n values long, then

$$\epsilon = (\ell + 1)/2^{29}$$

    (not too good, but OK for some applications)
  - Efficient on most 32-bit and 64-bit machines
- PolyQ64 with $p = 2^{64} - 59$: less efficient, but smaller ϵ

Attacks on Polynomial Hashes

$$H_k(m_1, \ldots, m_n) = m_1 k^1 + m_2 k^2 + \cdots + m_n k^n = \sum_{1 \le i \le n} m_i k^i.$$

- We know, this is ϵ-AΔU with $\epsilon \le n/|K|$.
- But can we "guess" messages m_1, m_2 with a likely difference $c = h(m_1) - h(m_2)$?
- Or is there anything else, we can do to break a Wegman-Carter construction, using H?

The Nonce-Reuse Attack

Attacking Wegman-Carter MACs with a “Two-Time-Pad”

- Consider a polynomial hash

$$H_k(m_1, \ldots, m_\ell) = \sum_{1 \le i \le \ell} m_i k^i = m(k).$$

- Eve does not know the secrets k and r, but she receives
  - $A = H_K(m) + r$ and
  - $A' = H_K(m') + r$

  for $m = (m_1, \ldots, m_\ell)$ and $m' = (m_1, \ldots, m_\ell)$ under her control
- Eve would like to find K. How should she choose m and m'?

The Polynomial Attack

Choose m and m', and c such that $\Pr[H_k(m) - H_k(m') = c] = \ell/|K|$

- Choose an arbitrary message $m = (m_1, \ldots, m_\ell)$. Any such message defines a polynomial $m(x) = \sum_{1 \le i \le \ell} m_i x^i$ with

$$H_k(m_1, \ldots, m_\ell) = \sum_{1 \le i \le \ell} m_i k^i = m(k).$$

- Guess ℓ keys $k_1, k_2, \ldots, k_\ell$ from K.
  We will succeed, if $k \in \{k_1, k_2, \ldots, k_n\}$.
- Compute the polynomial p with $p(x) = (k_1 - x)(k_2 - x) \cdots (k_\ell - x)$.
  This has degree ℓ, and $p(x) = 0 \Leftrightarrow x \in \{k_1, \ldots, k_\ell\}$.
- Determine $p_0, \ldots, p_\ell$, with $p(x) = \sum_{1 \le i \le \ell} p_i x^i$.

The Polynomial Attack (2)

- Choose $m' = (m_1 + p_1, m_2 + p_2, \ldots, m_\ell + p_\ell)$ and $c = p_0$.
- Now

$$\begin{aligned} H_k(M) - H_k(m') &= m(k) - m'(k) \\ &= m(k) - (m(k) + p(k) - p_0) \\ &= -p(k) + p_0. \end{aligned}$$

- If $k \in \{k_1, k_2, \ldots, k_\ell\}$, then $p(k) = 0$ and thus

$$H_k(M) - H_k(m') = p_0.$$

Another Practical Example: Poly-1305-AES

Bernstein, 2005

- Universal hash function PolyP130 with prime $p = 2^{130} - 5$:

$$\text{PolyP130}_k(m_0, \ldots, m_{\ell-1}) = \sum_{0 \le i < \ell} m_i k^{\ell-i-1} \pmod{p}.$$

  - Input blocks m_i are 128-bit values $m_i \in \{2^{128}, \ldots, 2^{129} - 1\}$.
  - Since $m_i \ge 128 \neq 0$, messages of different lengths are not a security issue (no need for PolyQ130)
- Nonce N ("number used once"), e.g., a counter
- Poly-1305-AES$((k, k'), m, N)$
  = $\text{PolyP130}_k(m) + \text{AES}_{k'}(N) \bmod 2^{128}$.
- To improve efficiency: set 22 bits of k to zero
- Conclusion from Theorem 29:
  If messages consist of at most n 128-bit blocks, then PolyP130 is ϵ-AΔU with $\epsilon \le \frac{n}{2^{106}}$.