

# More TWAMP – proposed features

Al Morton, Kaynam Hedayat  
March 2008

# 2008, The year of TWAMP

- Lots of people implementing now
- New readers = clarific. to base spec
- New readers = New Ideas
- Need a place to collect the ideas that have merit as extensions
- [draft-morton-more-twamp-00.txt](http://draft-morton-more-twamp-00.txt)

# Security Modes Extension

- RFC4656 OWAMP requires TEST to match the CONTROL protocol.
- “All OWAMP-Test sessions that are spawned by an OWAMP-Control session inherit its mode.”
- Seems that a MUST is missing there...

# Security Modes: Mandatory to Implement

- Secure modes may see more widespread use on the Control Protocol IF
  - Test protocol can run Unauthenticated
  - Priority on accuracy and scale when resources are limited

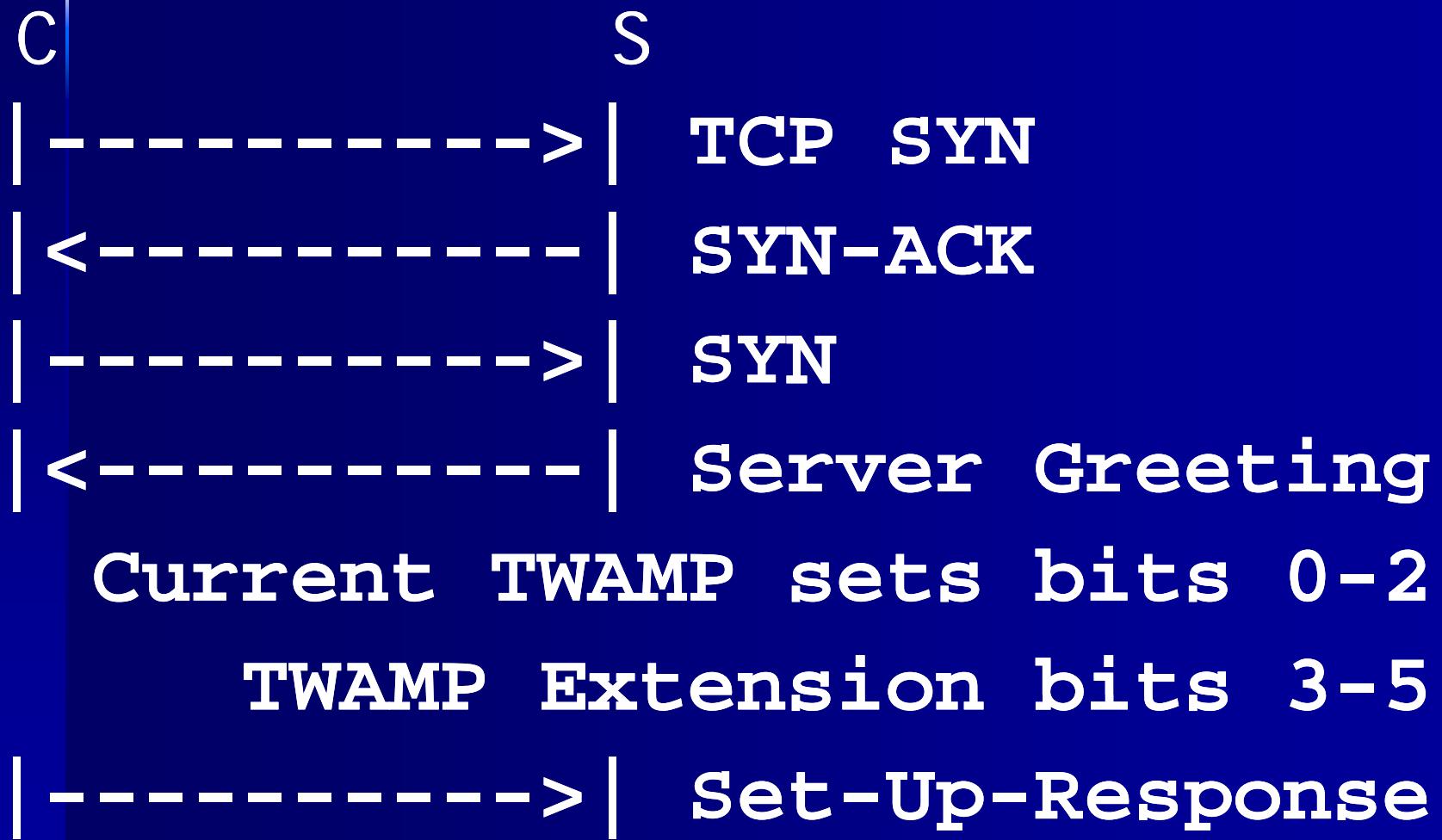
# Mode Field Extension (new registry needed)

Value	Description	Reference/Explanation
0	Reserved	
1	Unauthenticated	RFC4656, Section 3.1
2	Authenticated	RFC4656, Section 3.1
4	Encrypted	RFC4656, Section 3.1
8	Unauth. TEST protocol, Auth. CONTROL	new bit position (3)
16	Unauth. TEST protocol, Encrypted CONTROL	new bit position (4)
32	Auth. TEST protocol, Encrypted CONTROL	new bit position (5)

# Mode Field Extension

Protocol	Permissible Mode Combinations		
Control	Unauth.	Auth.	Encrypted
	Unauth.	<u>Unauth.</u>	<u>Unauth.</u>
Test		Auth.	<u>Auth.</u>
			Encrypted

# Mode Field Extension Compatibility (easy!)



# Question for IPPM WG

- Does this sound like a reasonable extension?

# Another Suggestion: Reflect Packet Padding (Len Ciavattone)

- Applies to both Control and Test
- Session Reflector needs to move the first X octets of the Padding to the reflected or reply packet
- Session-Sender knows what was put in, the format used, etc.
- Adds to possibilities for extensions
  - Special processing key in some packets

# Reflect Packet Padding: Mode Field Extension

Value	Description	Reference/Explanation
...		
8	Unauth. TEST protocol, new bit position (3) Auth. CONTROL	
16	Unauth. TEST protocol, new bit position (4) Encrypted CONTROL	
32	Auth. TEST protocol, new bit position (5) Encrypted CONTROL	
-----		
64	Reflect Padding Capability	new bit position (6)
128	Reflect & Operate on Padding Bits	new bit position (7)

## Reflect Packet Padding: TWAMP Test Sender packet

The diagram illustrates a 3D vector packet structure. The vertical axis (y-axis) shows the sequence of fields from top to bottom: Sequence Number, Timestamp, Error Estimate, MBZ, Length (2 oct), Ext ID, Packet Padding (to be reflected), and Additional Packet Padding. The horizontal axis (x-axis) shows the sequence of bytes across three dimensions: 0, 1, 2, and 3. Dimension 0 contains the first byte of each field. Dimension 1 contains the second byte of each field. Dimension 2 contains the third byte of each field. Dimension 3 contains the fourth byte of each field. The length of each field is indicated by a vertical bar: Sequence Number (4 bytes), Timestamp (4 bytes), Error Estimate (1 byte), MBZ (1 byte), Length (2 oct) (2 bytes), Ext ID (1 byte), Packet Padding (to be reflected) (variable length), and Additional Packet Padding (variable length).

## Reflect Packet Padding: TWAMP Test Reflector packet

The diagram illustrates a 32-bit packet structure divided into several fields:

- Sequence Number**: 16 bits, starting at index 0.
- Timestamp**: 16 bits, starting at index 1.
- Error Estimate**: 16 bits, starting at index 2.
- MBZ**: 16 bits, starting at index 3.
- Receive Timestamp**: 16 bits, starting at index 4.
- Sender Sequence Number**: 16 bits, starting at index 5.
- Sender Timestamp**: 16 bits, starting at index 6.
- Sender Error Estimate**: 16 bits, starting at index 7.
- MBZ**: 16 bits, starting at index 8.
- Sender TTL**: 16 bits, starting at index 9.
- Length (2 oct)**: 16 bits, starting at index 10.
- Ext ID**: 16 bits, starting at index 11.
- Packet Padding (from Session-Sender)**: 16 bits, starting at index 12.
- Additional Packet Padding**: 16 bits, starting at index 13.

# Reflect Packet Padding: TWAMP Control

