

Purple

# Purple

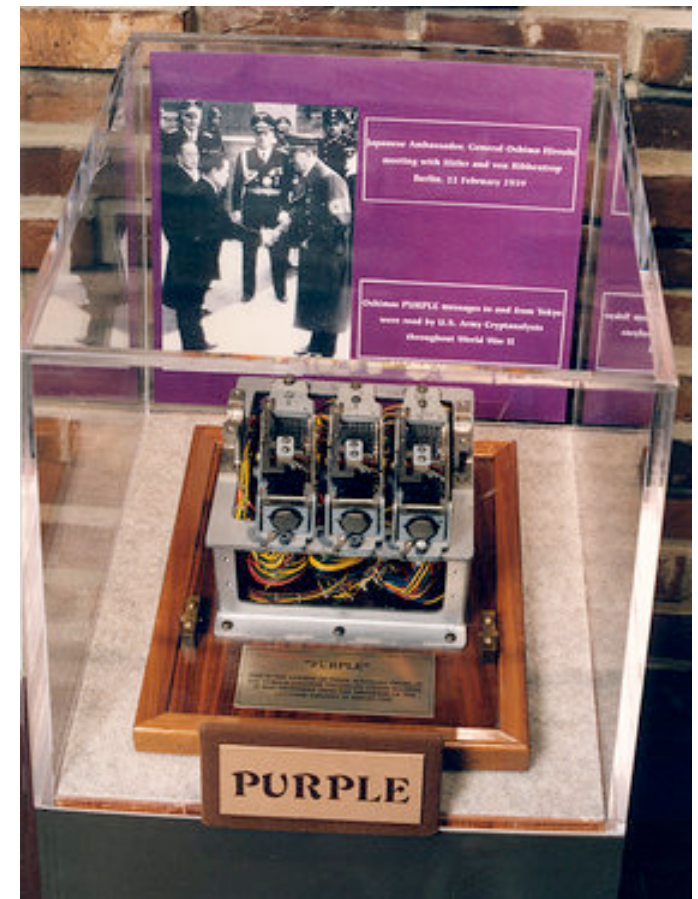
- ❑ Used by Japanese government
  - Diplomatic communications
  - Named for color of binder cryptanalysts used
  - Other Japanese ciphers: Red, Coral, Jade, etc.
- ❑ Not used for tactical military info
  - That was JN-25
- ❑ Used to send infamous "14-part" message
  - Broke off negotiations with U.S.
  - Supposed to be delivered in Washington immediately before attack at Pearl Harbor
  - Actually delivered after attack began

# Purple

- ❑ The “14-part” message
  - Decrypted by U.S. on December 6, 1941
  - No explicit warning of attack but...
  - ...Marshall sent warning to Hawaii
  - Warning arrived **after** attack was over
  - Endless fuel for conspiracy theorists
- ❑ Purple provided useful intelligence
  - For example, info on German D-day defenses
- ❑ Tactical military info was from JN-25
  - Midway/Coral Sea, Admiral Yamamoto, etc.

# Purple

- ❑ No intact Purple machine ever found
- ❑ This fragment from embassy in Berlin
  - Recovered from rubble at the end of war



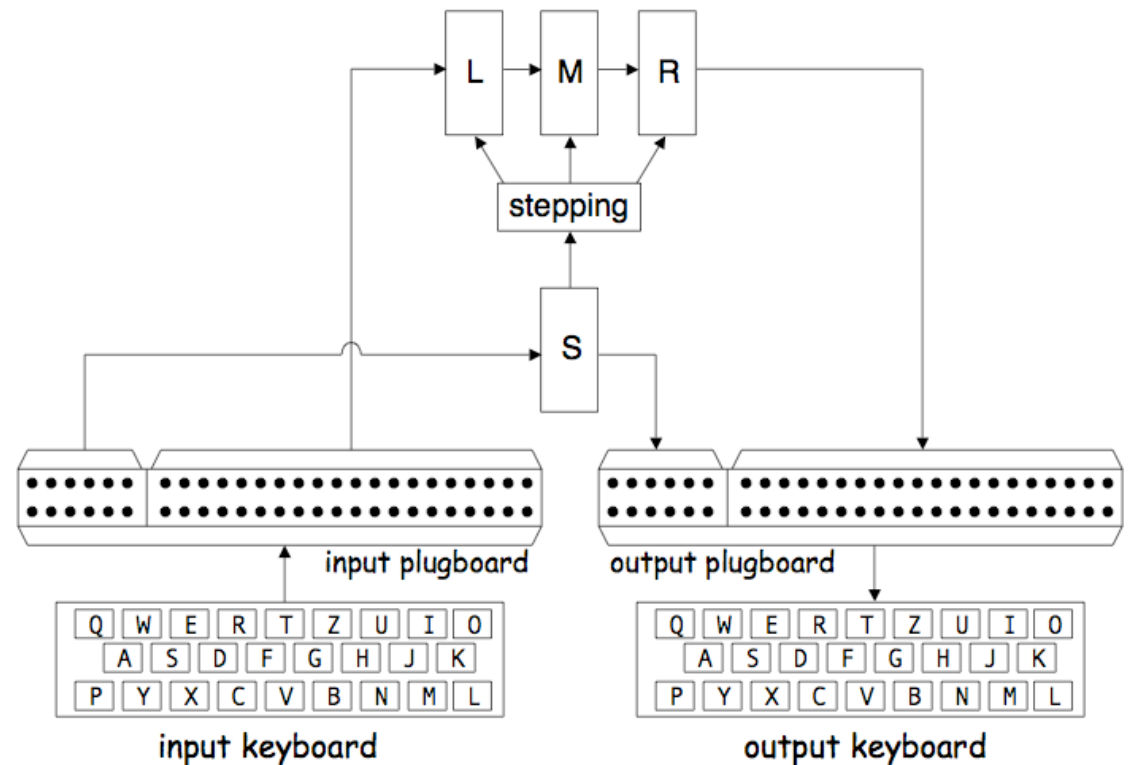
# Purple

- ❑ Simulator Constructed by American cryptanalysts
  - Rowlett gets most credit
  - Friedman, others involved
- ❑ Simulator based on intercepted ciphertext
  - Analysts never saw the Purple machine...
  - ...yet they built a functioning replica
  - Some say it was greatest crypto success of the war



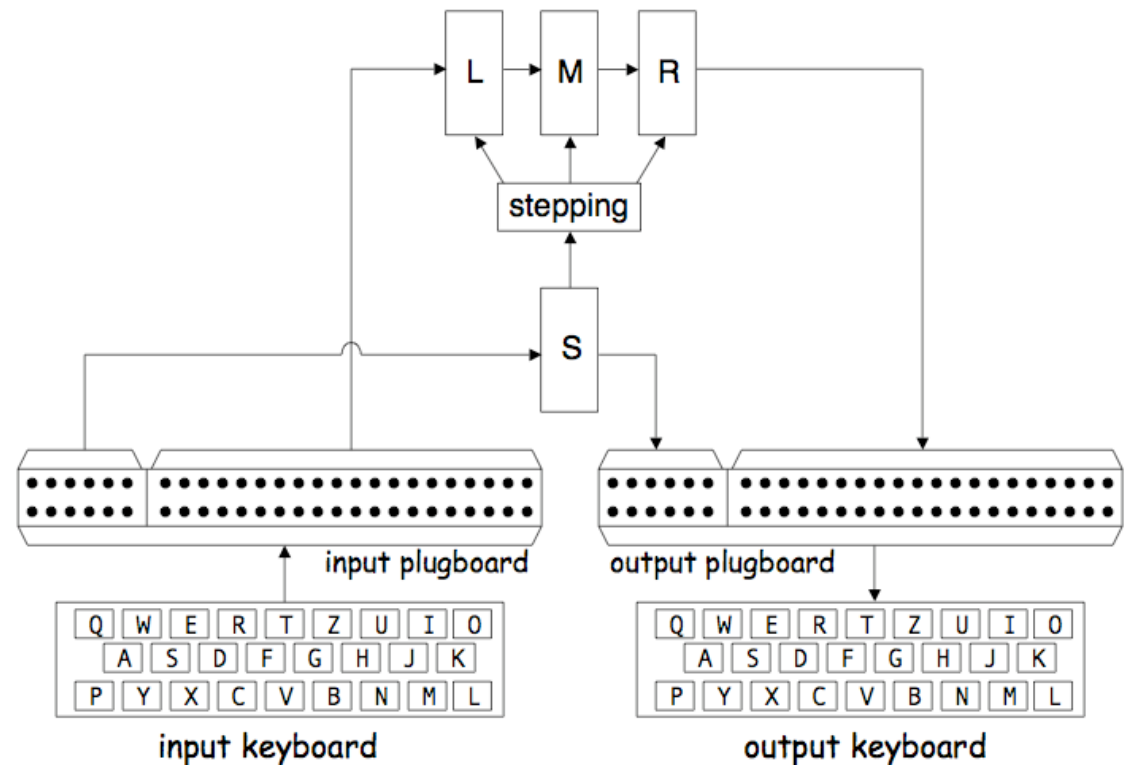
# Purple

- ❑ Switched permutations
  - Not rotors!!!
- ❑ S, L, M, and R are switches
  - Each step, one of the perms switches to a different perm



# Purple

- ❑ Input letter permuted by plugboard, then...
- ❑ Vowels and consonants sent thru different switches
- ❑ The "6-20 split"



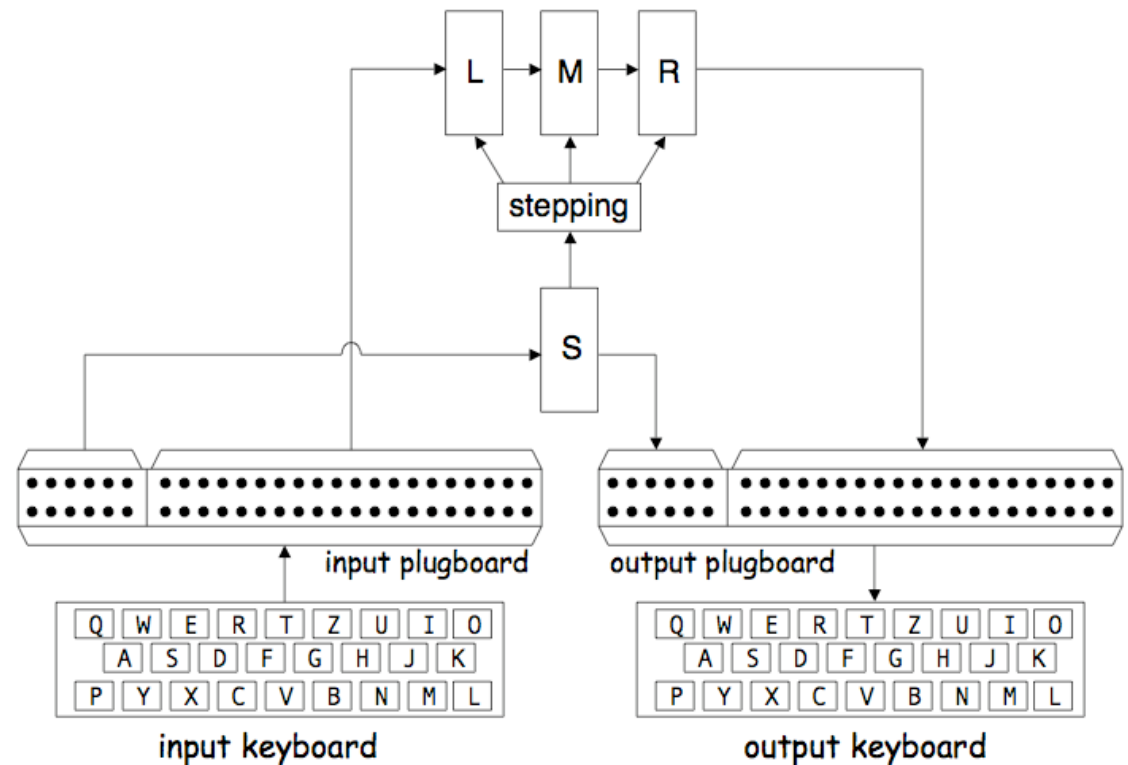
# Purple

## ❑ Switch S

- Steps once for each letter typed
- Permutes vowels

## ❑ Switches L,M,R

- One of these steps for each letter typed
- L,M,R stepping determined by S





# Purple Plugboard

- ❑ Purple Plugboard
  - Every letter plugged to another letter
  - **Not** the same as Enigma stecker
- ❑ Purple, plugboard
  - Could be any permutation of 26 letters
- ❑ Enigma stecker
  - Pairs connected (so stecker is its own inverse)
  - Only a limited set of perms are possible

# Purple

- ❑ Each switch S,L,M,R has 25 different, unrelated, hardwired permutations
  - Each L,M,R permutes 20 consonants
  - Each S permutes 6 vowels
- ❑ Period for 6s perms is 25
- ❑ Period for 20s perms is  $25^3 = 15,625$
- ❑ Set fast, medium, slow of L,M,R

# Purple

- ❑ Each switch S,L,M,R has 25 different, unrelated, hardwired permutations
  - Each L,M,R permutes 20 consonants
  - Each S permutes 6 vowels
- ❑ Purple is **not** its own inverse
- ❑ To decrypt? Reverse the flow thru diagram
- ❑ In WWII, input and output plugboard settings were apparently always the same
  - Why?

# Purple Encryption

- Let  $P_I, P_O, P_S, P_R, P_M, P_L$  be input and output plugboards, "sixes" perm, R, M, L "twenties" perms, respectively
- Note:  $P_S, P_R, P_M, P_L$  vary with step
- Then encryption formula is

$$y = \begin{cases} P_O^{-1} P_R P_M P_L P_I(x) & \text{if } P_I(x) \text{ is one of the twenties} \\ P_O^{-1} P_S P_I(x) & \text{if } P_I(x) \text{ is one of the sixes} \end{cases}$$

# Purple Encryption

- And decryption is

$$x = \begin{cases} P_I^{-1}P_L^{-1}P_M^{-1}P_R^{-1}P_O(y) & \text{if } P_O(y) \text{ is one of the twenties} \\ P_I^{-1}P_S^{-1}P_O(y) & \text{if } P_O(y) \text{ is one of the sixes} \end{cases}$$

- If  $P_I \neq P_O$  then decryption is complex
- Requires inverse plugboard perms
  - Or tricky wiring

# Purple Keyspace

- If switch perms unknown, then

$$(6!)^{25} \cdot (20!)^{75} = 2^{237} \cdot 2^{4581} = 2^{4818} \text{ switches}$$

$$25^4 = 2^{18.6} \text{ switch initial settings}$$

$$6 = 2^{2.6} \text{ choices for fast, medium, slow}$$

$$(26!)^2 = 2^{176.8} \text{ plugboards}$$

- Implies keyspace is about  $2^{5016}$

# Purple Keyspace

- ❑ If switch perms are **known**, then
  - $25^4 = 2^{18.6}$  switch initial settings
  - $6 = 2^{2.6}$  choices for fast, medium, slow
  - $(26!)^2 = 2^{176.8}$  plugboard (assuming only 1)
- ❑ Keyspace is “only” about  $2^{198}$
- ❑ Note that most of this is due to plugboard
  - But plugboard is cryptographically very weak

# Purple

- ❑ Purple message included “message indicator”
  - “Code” to specify initial switch settings
  - MI different for each message
- ❑ “Daily key” was plugboard setting
- ❑ Cryptanalysts needed to
  - Determine inner workings of machine, that is, **diagnose** the machine
  - Break indicator system (easier)



# Purple

- ❑ Only about 1000 daily keys used
- ❑ So once the machine was known
  - After a number of successful attacks...
  - ...cryptanalysts could decrypt messages as fast as (or faster than) the Japanese
- ❑ But, how to diagnose the machine?
- ❑ Only ciphertext is available!

# Purple Diagnosis

- ❑ From cryptanalysts perspective
  - Know Purple is poly-alphabetic substitution
  - But how are permutations generated?
- ❑ The 6-20 split is a weakness
  - Suppose D,E,H,Q,W,X are plugged to vowels A,E,I,O,U,Y, respectively, by input plugboard
  - Assume input/output plugboards are the same
  - Then output D,E,H,Q,W,X go thru S perms
  - All other output letters go thru L,M,R perms
  - So what?

## 6-20 Split

- ❑ Suppose D,E,H,Q,W,X are the sixes
- ❑ Input plugboard
  - Connects D,E,H,Q,W,X to vowels
- ❑ Output plugboard
  - Connects A,E,I,O,U,Y to D,E,H,Q,W,X
- ❑ Can cryptanalyst determine the sixes?

## 6-20 Split

- Average letter frequency of D,E,H,Q,W,X is about 4.3%
- Average letter frequency of remaining 20 letters is about 3.7%
- **Each** of the sixes letters should appear about 4.3% of the time
- Each "20" should appear about 3.7%

# 6-20 Split

- ❑ For any ciphertext of reasonable length, usually relatively easy to find 6s
  - 6 high frequency or 6 low frequency letters
- ❑ Then easy to find 6s permutations
  - Hardwired, so never change (only 25 of them)
- ❑ With this info, can decrypt some messages
  - Especially if 6s were actually vowels...
  - As was the case with Red (predecessor to Purple)

# 6-20 Split

- ❑ Can solve for the 6s...
- ❑ But what about 20s?
- ❑ WWII cryptanalysts familiar with rotors
  - From ciphertext, easy to see that Purple not a rotor machine...
  - But what was it?
- ❑ Suppose, many messages collected, many of these broken, so **known plaintext** available

# 20s

- ❑ Suppose that many messages encrypted with same key
  - Said to be in **depth**
- ❑ Suppose the plaintext is also known
- ❑ Then analyst knows lots of 20s perms...
- ❑ But this is only a small part of key space
- ❑ So how can this help?
  - Consider a scaled-down example

# 20s

- ❑ Consider "7s" instead of "20s"
  - Numbers instead of letters
  - Then perms of 0,1,2,3,4,5,6
- ❑ Known plaintext so encryption perms known
  - Msg 1, first position, plaintext 4 → ciphertext 1
  - Msg 2, first position, plaintext 0 → ciphertext 5
  - Msg 3 first position, plaintext 5 → ciphertext 2
  - Msg 4, first position, plaintext 6 → ciphertext 4
  - Msg 5, first position, plaintext 3 → ciphertext 3
  - Msg 6, first position, plaintext 1 → ciphertext 6
  - Msg 7, first position, plaintext 2 → ciphertext 0
- ❑ Then 1st perm is 5,6,0,3,1,2,4



# 20s (actually, 7s)

permutations

key position		0	1	2	3	4	5	6
	0	5	6	0	3	1	2	4
	1	6	1	0	4	5	2	3
	2	4	2	1	6	5	0	3
	3	6	0	4	1	2	5	3
	4	6	1	5	2	3	0	4
	5	6	2	1	5	0	3	4
	6	5	3	2	6	0	1	4
7	3	4	6	5	1	0	2	

Purple

# 20s

- ❑ Pattern on previous slide occurs if same sequence of permutations applied
  - But **input** is different (permuted)
  - Consistent with “switched” permutations
- ❑ Looks easy here, but not so easy when
  - Period of 25 for fast 20s switch
  - Only partial permutations available
  - Do not know what you are looking for!

## 20s

- ❑ Analysts determined three switches
  - Each with 25 perms
- ❑ Can then solve equations to peel apart perms
- ❑ Had to construct a working Purple simulator
  - How to do so?

## 20s

- ❑ How to implement switched perms?
- ❑ Used six 4-level telephone switches
- ❑ Discovered after the war that this is exactly what Japanese had used
- ❑ That's what you call ironic...

# Hill Climb Attack

- ❑ In modern symmetric ciphers
  - If key is incorrect by one bit, then putative decrypt unrelated to plaintext
- ❑ Purple cipher
  - "Nearby" plugboard settings yield approximate plaintext
  - A so-called hill climb attack is possible

# Cryptanalysts

- ❑ Purple broken by Frank Rowlett's team
- ❑ Rowlett among designers of Sigaba
  - Sigaba was never broken during war
  - Today, Sigaba not trivial to break
- ❑ We talk about Sigaba next...

# Purple: The Bottom Line

- ❑ As with Enigma, designers confused physical security and statistical security
  - Even worse for Purple than with Enigma
  - Physical security of Purple was protected
- ❑ Once Purple machine diagnosed
  - And message indicator system broken
- ❑ Then a very small number of “keys”
  - Only about 1000 plugboard settings