

# Hw8: the Enigma Part II



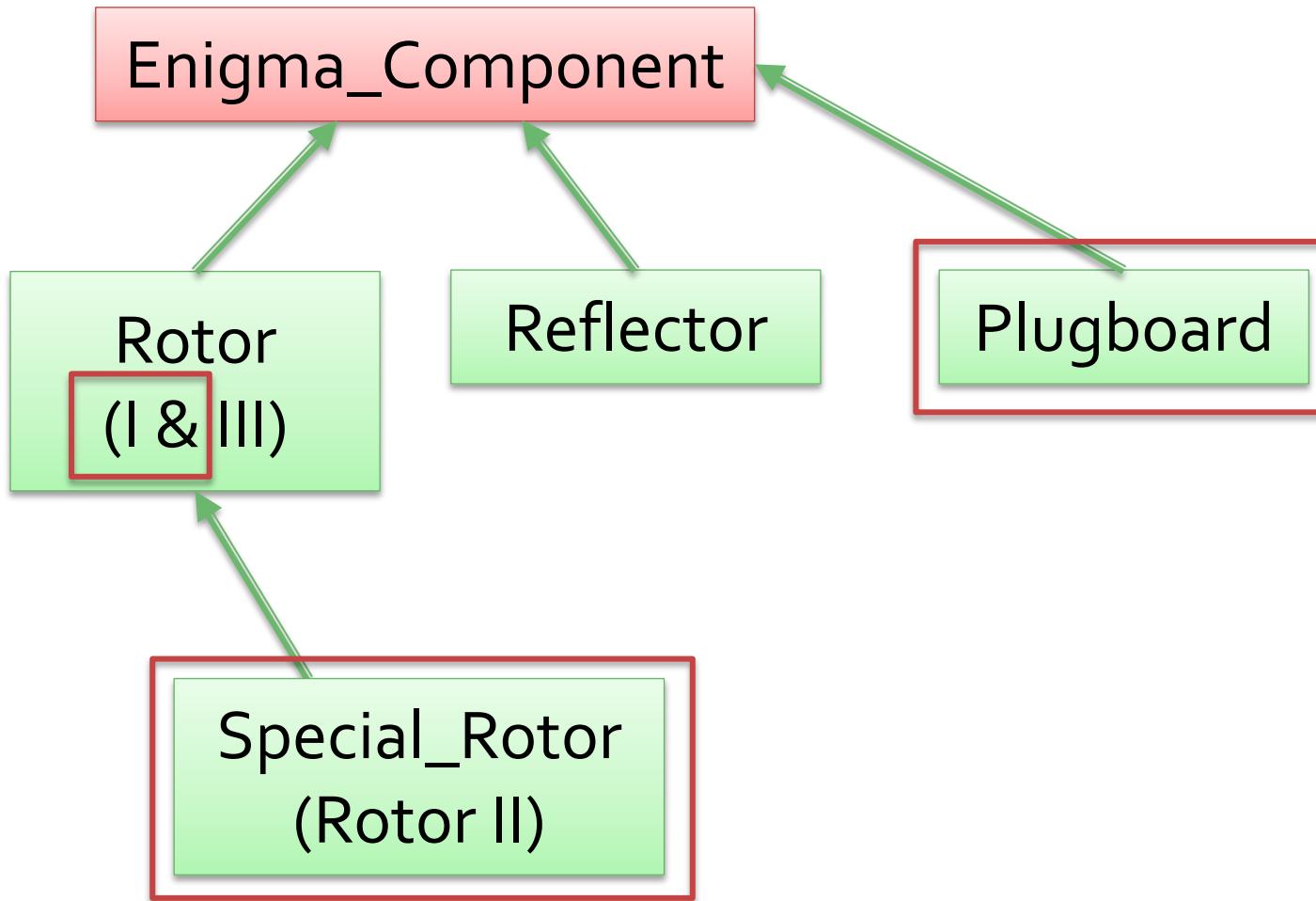
[https://en.wikipedia.org/wiki/Enigma\\_machine](https://en.wikipedia.org/wiki/Enigma_machine)

[https://en.wikipedia.org/wiki/Enigma\\_rotor\\_details](https://en.wikipedia.org/wiki/Enigma_rotor_details)

<http://www.enigmaco.de/enigma/> <- DEMO

Due date: 6/17 (Sun.)

# Class Enigma\_Component (extended)



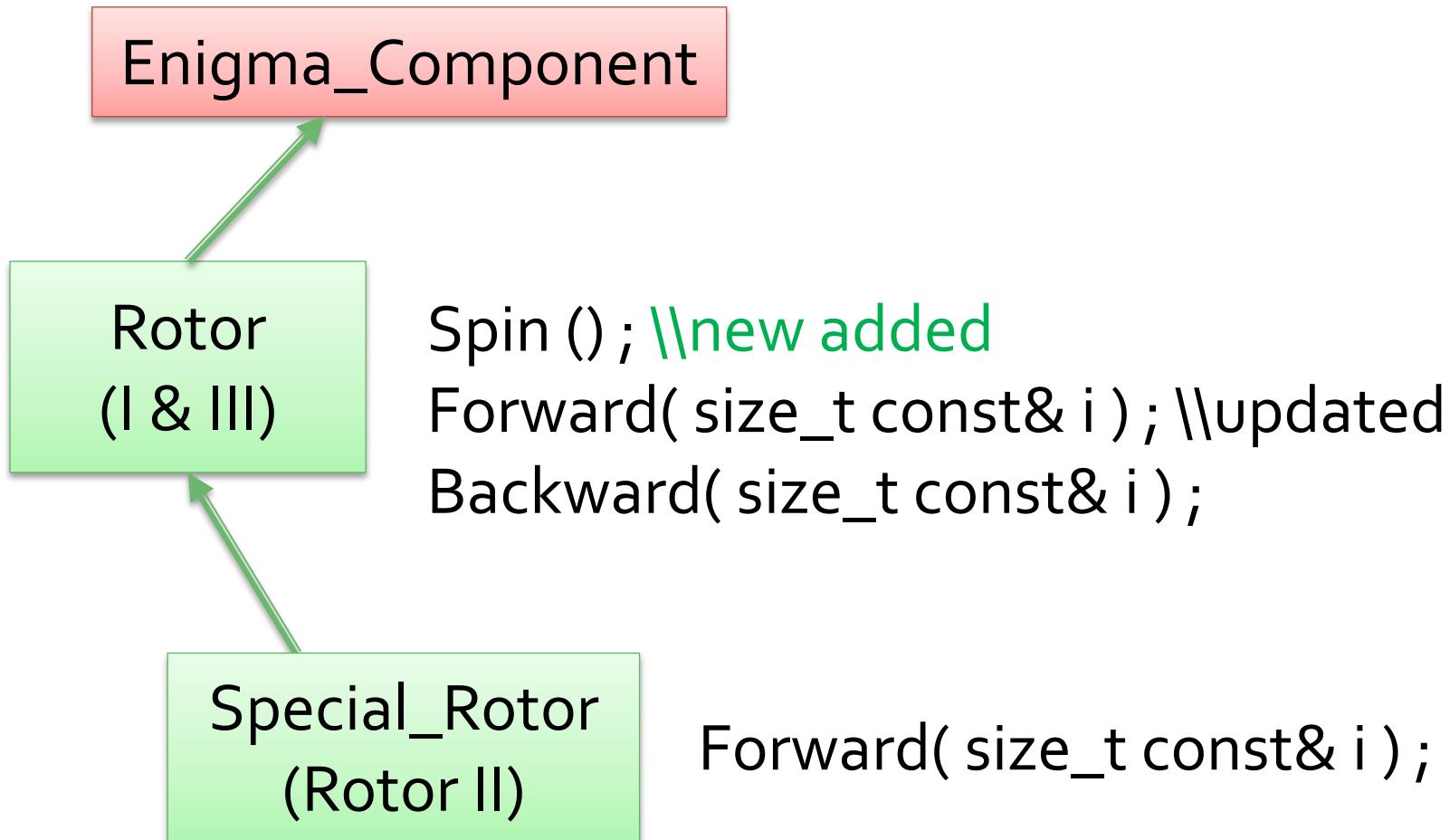
# Class Enigma\_Component (unchanged)

a\_Component

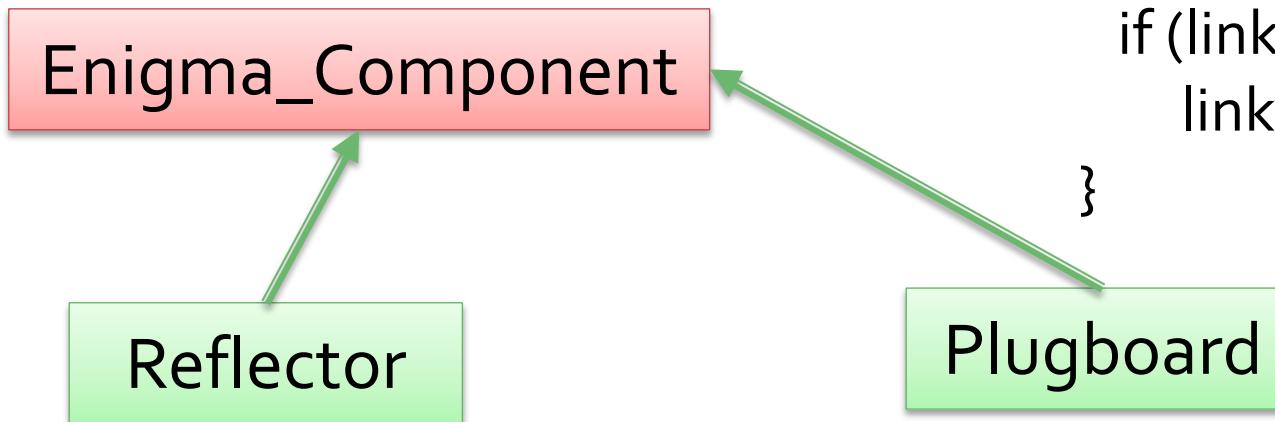
```
Enigma_Component ( std::string file )
char Input_signal ( char c )
void Link ( Enigma_Component & next )
virtual void Reset ()
```

```
Enigma_Component* link_next ;
Enigma_Component* link_previous ;
//Mapping of forward/backward Encoding
std::vector<size_t> encode_table;
std::vector<size_t> reverse_table;
virtual size_t Forward( size_t const& i ) = 0 ;
virtual size_t Backward( size_t const& i ) = 0 ;
void Read_table ( std::string file );
virtual void Spin () {}
```

# Class Rotor and Special Rotor (extended)



# Class Reflector and Plugboard (extended)



Forward( size\_t const& i );  
Backward( size\_t const& i );  
\unchanged

```
void Plugboard::Spin(){  
    if (link_next)  
        link_next->Spin();  
}  
  
Spin();  
Forward( size_t const& i );  
Backward( size_t const& i );
```

# main\_105hw7.cpp (Loading)

```
Enigma_Files Rotor_start_file, ..., encoded_message_file;  
...  
original_message_file.Read_file("original_data.txt");  
Rotor_arrow_file.Read_file("Rotor_arrow_web.txt");  
Rotor_start_file.Read_file("Rotor_start_web.txt");
```

```
Plugboard plugboard("Plugboard_web.txt");  
Rotor Rotor_III("Rotor_III_web.txt", Rotor_start_file[0], Rotor_arrow_file[0]);  
Special_Rotor Rotor_II("Rotor_II_web.txt", Rotor_start_file[1], Rotor_arrow_file[1]);  
Rotor Rotor_I("Rotor_I_web.txt", Rotor_start_file[2], Rotor_arrow_file[2]);  
Reflector reflector("Reflector_web.txt");
```

```
plugboard.Link(Rotor_III);  
Rotor_III.Link(Rotor_II);  
Rotor_II.Link(Rotor_I);  
Rotor_I.Link(reflector);
```

# main\_106hw8.cpp (Encoding)

```
std::cout << "*Original message: " << original_message_file.Data() << std::endl;  
  
for (size_t i = 0; i < original_message_file.Length(); i++)  
    encoded_message_file.Push(plugboard.Input_signal(original_message_file[i]));  
  
encoded_message_file.Write_file("encoded_data.txt");  
std::cout << "*Encoded message: " << encoded_message_file.Data() << std::endl;  
  
std::cout << "*Decode it back..." << std::endl;  
plugboard.Reset(); //(make a chain reaction to reset all encoder)  
  
for (size_t i = 0; i < encoded_message_file.Length(); i++)  
    std::cout << plugboard.Input_signal(encoded_message_file[i]);
```

# Rules of Assignment

- Encode string from file (original\_data.txt) and output encoded string to encoded\_data.txt
  - Implement several additional class for different encoder
    - **Rotor** (hw6) for Rotor III & Rotor I (**with spin()**)
    - **Special\_Rotor** for Rotor\_II
    - **Plugboard**
  - Use **Linked List** to construct the encoding chain and Reset() chain

# Plugboard of Enigma

- 接線板 Plugboard
  - 接線板的連線



Keyboard	A	B	C	D	E	F	G	H	I	J	K	L	M
Plugboard	F	B	C	D	K	A	G	H	I	J	E	L	T
Keyboard	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Plugboard	N	O	P	W	R	S	M	U	V	Q	X	Y	Z

# Ring Settings of Enigma

- 字母環 Ring settings (Ringstellung)
  - 字母環與旋轉盤線路的相對位置

Rotor III (外圈)	A	B	C	D	E	F	G	H	I	J	K	L	M
Rotor III (内圈)	B	D	F	H	J	L	C	P	R	T	X	V	Z
Rotor III (外圈)	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Rotor III (内圈)	N	Y	E	I	W	G	A	K	M	U	S	Q	O

# Ring Settings of Enigma

- 字母環 Ring settings (Ringstellung)
  - 字母環與旋轉盤線路的相對位置

Rotor II (外圈)	A	B	C	D	E	F	G	H	I	J	K	L	M
Rotor II (内圈)	A	J	D	K	S	I	R	U	X	B	L	H	W
Rotor II (外圈)	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Rotor II (内圈)	T	M	C	Q	G	Z	N	P	Y	F	V	O	E

# Ring Settings of Enigma

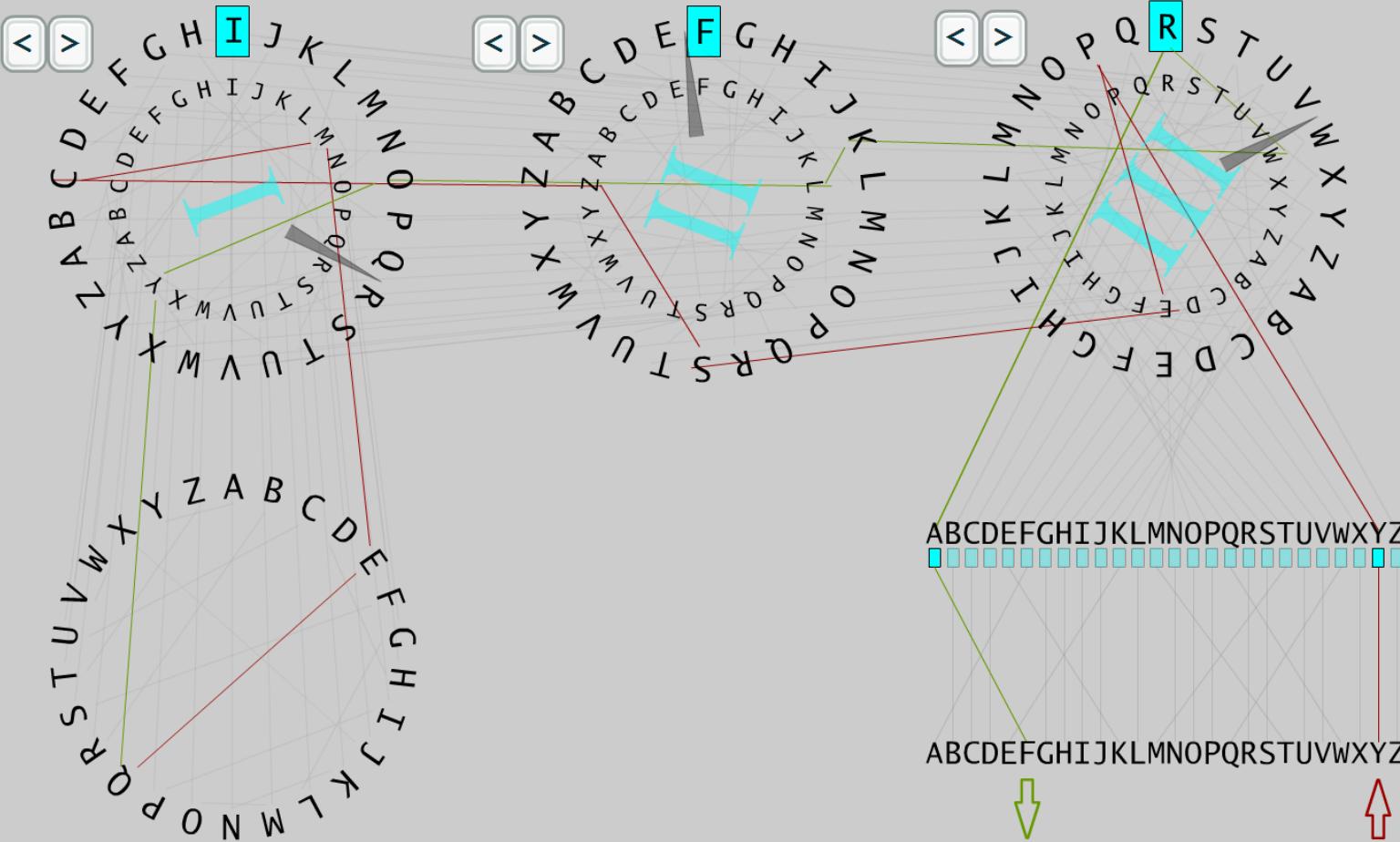
- 字母環 Ring settings (Ringstellung)
  - 字母環與旋轉盤線路的相對位置

Rotor I (外圈)	A	B	C	D	E	F	G	H	I	J	K	L	M
Rotor I (内圈)	E	K	M	F	L	G	D	Q	V	Z	N	T	O
Rotor I (外圈)	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Rotor I (内圈)	W	Y	H	X	U	S	P	A	I	B	R	C	J

# Reflector of Enigma

## ■ 反射器 Reflector

Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Reflector	Y	R	U	H	Q	S	L	D	P	X	N	G	O
Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Reflector	K	M	I	E	B	F	Z	C	W	V	J	A	T



Input:

KTWREEOSTUVDOPLLBCOSVXVOLPBZSQKWENMRQHXJWRZDCY

Output:

PKRFLGFFYFFFFFFFFFFFFFFXTPMFBFRFFFIFFFFF

Status: Highlighted wires show steps of encryption.



```
Prof. Yeh's version (hw7)
Read original message: "original_data.txt"
Read arrows position of [Rotor]: "Rotor_arrow_web.txt"
Read start position of [Rotor]: "Rotor_start_web.txt"
Read [Plugboard]: "Plugboard_web.txt"
Read [Rotor III]: "Rotor_III_web.txt"
Read [Rotor II]: "Rotor_II_web.txt"
Read [Rotor I]: "Rotor_I_web.txt"
Read [Reflector]: "Reflector_web.txt"
*Original message:
KTWREEOSTUVDOPLLBCOSUXUOLPBZSQKWENMRQHXJWRZDCY
(Output encoded data to "encoded_data.txt")
*Encoded message:
PKRFLGFFYFFFFFFFFFFFFFFXTPMFBFRFFFIFFFFF
*Decode it back...
KTWREEOSTUVDOPLLBCOSUXUOLPBZSQKWENMRQHXJWRZDCY
Press any key to exit...
```

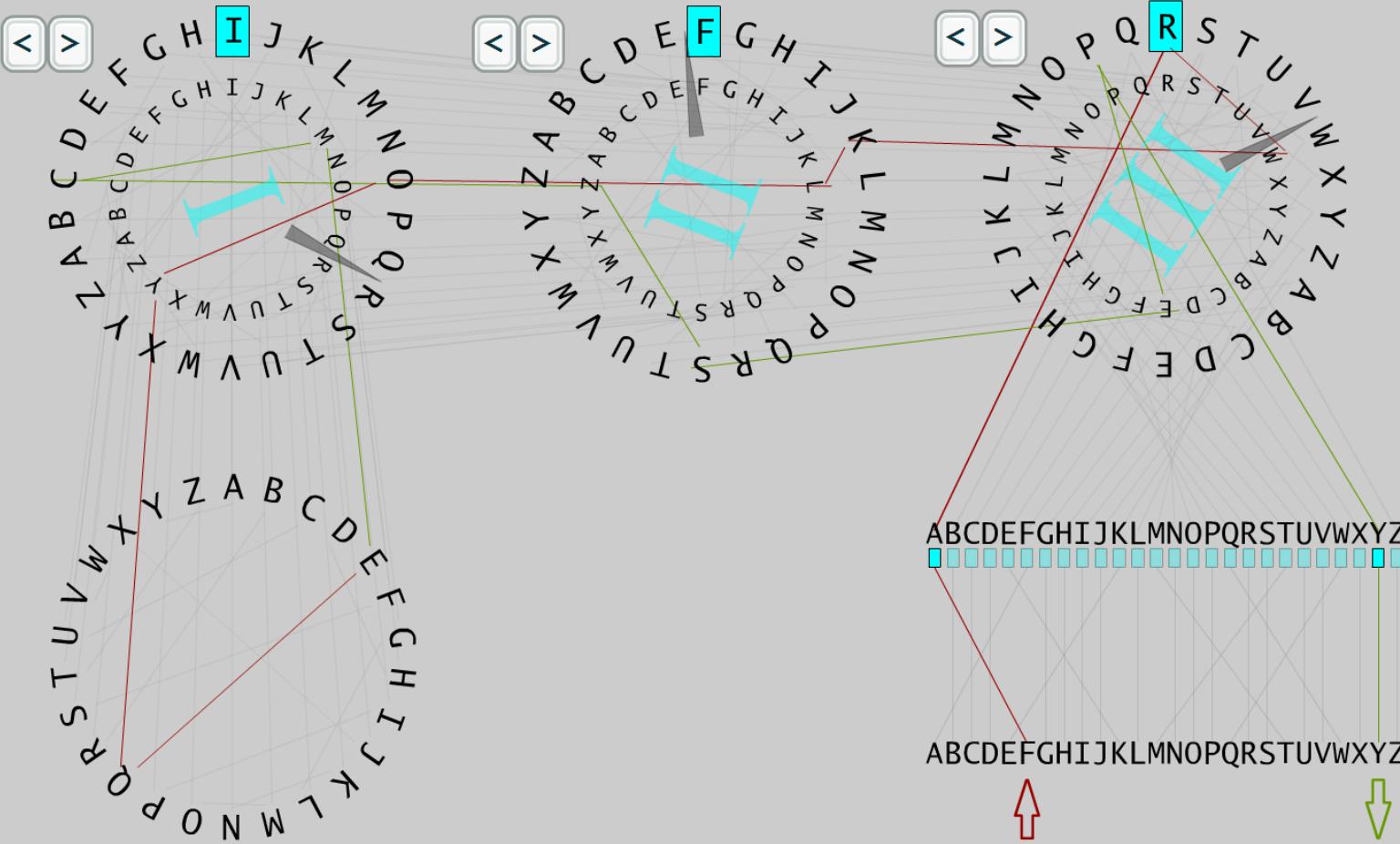
Input:

KTWREEOSTUVDOPLLBCOSVXVOLPBZSQKWENMRQHXJWRZDCY

Output:

PKRFLGFFYFFFFFFFFFFFFFFXTPMFBFRFFFIFFFFF

Status: Highlighted wires show steps of encryption.



## Reverse

Input:

|PKRFLGFFYFFFFFFFFFFFFFFFFFFXTPMFBFRFFFIFFFFF

Output:

KTWREEOSTUVDOPLLBCOSVXVOLPBZSQKWENMRQHXJWRZDCY

Status: Highlighted wires show steps of encryption.