TLS_Handshake_Agent

- session: std::shared_ptr<Session>
- current_state: State
- prime_group: int
- G: std::shared_ptr<BigInt>
- P: std::shared_ptr<BigInt>
- s: std::shared_ptr<BigInt>
- S: std::shared_ptr<BigInt>
- c: std::shared_ptr<BigInt>
- C: std::shared_ptr<BigInt>
- key: std::shared_ptr<BigInt>
- local_protocol: std::string
- partner_protocol: std::string
- partner_encrypted: bool

- check_protocols(): void
- handle_message(tls::Message_Wrapper): void
- receive_client_hello(): void
- receive_server_hello(tls::Message_Wrapper): void
- receive_certificate(tls::Message_Wrapper): void
- receive_server_hello_done(): void
- receive_client_key_exchange(tls::Message_Wrapper): void
- receive_finished(tls::Message_Wrapper): void

+ TLS_Handshake_Agent(std::shared_ptr<Session>)
+ notify(tls::Message_Wrapper, unsigned int): void
+ initiate_handshake(): void
+ is_secure(): bool
+ is_establishing(): bool
+ reconnect(): void
+ get_key(): std::string

+ generate_random_number(BigInt, BigInt): BigInt
+ red_primes_json(std::string, int, BigInt&, BigInt&): void
+ encrypt(const std::string&, unsigned long&, const string&): void
+ decrypt(const std::string&, unsigned long&, const string&): void
+ encode_base64(const std::string&): void
+ decode_base64(const std::string&): void
+ send_message(std::string, unsigned long&, std::string): void
+ receive_message(std::string, unsigned long, std::string): void