Design and analysis of pairing protocol for Bluetooth enabled devices using R-LWE Lattice-based cryptography

INTRODUCTION

* Up to Bluetooth 2.0 pairing : symmetric cryptography = both devices shared the identical PIN code
* Bluetooth 2.1 pairing : SSP - **Secure Simple Pairing** to improve security in piconet and scatter net

Uses *Elliptic Curve Diffie–Hellman* (ECDH) public-key cryptography

Improves BT pairing algorithm but has security issues - researchers observed and introduced many attacks

* Standard and classical computers employing elliptic curve cryptosystems are secure, because the inverse of the discrete logarithm problem on elliptic curves is liable to be unbreakable
* However, this is not true for future **quantum computers**.
* **Lattice-based cryptographic** constructions hold a great promise for post-quantum cryptography, as they enjoy very strong security proofs based on worst-case hardness
* Therefore, CRSSP is proposed over SSP
* **Classiquantum Resistance Secure Simple Pairing (CRSSP):** a quantum resistance pairing protocol for quantum computers using **Lattice-Based Ring Learning With Errors (R-LWE)**

Secure Simple Pairing in Bluetooth and attacks

1. Secure Simple Pairing in Bluetooth with numeric comparison mode

Phases

1. Unencrypted exchange of IO capability and device address (Phase 1)
2. i) Each device generates its own ECDH public-private key pair

ii) Each device computes the Diffie–Hellman shared secret key

*Phase 2: Public Key Exchange*

SecretKey = f1 ( ECPrivateKey, ECPublicKeyAlice, ECPublicKeyBob )

1. Authentication Stage 1: nonce are exchanged and then both devices use this nonce to validate each other.

*Phase 3: Authentication Stage 1*

validate = g ( ECPublicKeyAlice, ECPublicKeyBob, NonceAlice, NonceBob )

1. Authentication Stage 2: certifies both devices have efficiently completed exchange using Challenge-Response scheme

*Phase 4: Authentication Stage 2*

Substantiate ( Ea , Eb ) = f3 ( ECDHKey, AliceNonce, BobNonce, IOCap, AliceBDAddr, BobBDAddr )

1. On completion of authentication Stage 2, both parties compute link key with parameters DHKey, Nonce, public key, Bluetooth address and IO capability. *This link key is further used to generate an encryption key for encryption of messages or file. Here, g (…) f 3 (…) and f 4 (…) are HMAC-SHA256 functions and f 1 (…) is Diffie–Hellman algorithm*

*Phase 5: Link Key Calculation*

SessionKey = f4 ( ECDHKey, AliceNonce, BobNonce, “btlk”, AliceBDAddr, BobBDAddr )

1. Security threats in SSP

* Due to open nature of wireless media: transmission can be easily packed or cached

causes false and altered information to be inserted and delivered to piconet devices.

* Jakobsson devised "Off-Line PIN crunching attack" and **captured the link key** by observing key establishment protocol
* Based on this weakness, **Elliptic-Curve-Diffie–Hellman** (DH) key exchange pairing mechanism was proposed **to build up the security of the link key**
* To compute encryption key, Fluhrer and Lucks observed key stream and used public knowledge of the encryption mechanism in E0.
* Best-known attack so far against E0 devised by Luand and Vaudenay have made possible to recover the closest codeword for any linear code
* BT 2.1 + EDR increased the strength of security by providing the protection against both
  + **passive eavesdropping** attacks and
  + Man-In- The-Middle (MITM) attacks (**active eavesdropping** attacks)
* Even then, there were still ways to attack SSP
* Attackers get control over insecure channel in first phase of pairing and modify information to use Just Work as an association model
* During pairing process in passkey entry mode MITM mounts the attack and breaks the password
* Heart of the problem resides where the two devices pair for the first time or when they have to re-establish a link key
* Flaws in SSP is because:
  + Encryption key length is open to discussion,
  + Device authentication is performed but user verification is not done,
  + Link keys are stored in non-volatile memory,
  + Influence of the pseudo-random number generators (PRNG) is not known,
  + Peer-to-peer security is not realized instead only individual links are encrypted and authenticated,
  + Data may be decrypted in-between,
  + Security services are limited,
  + Low-Energy pairing do not provide eavesdropping protection.
* Better not to use Just Works association model for highly confidential data since it does not provide MITM protection

1. Lattice-based cryptography

* Relying only on asymmetric cryptography based on the stability of factoring or ECDLP is not enough
* This is because of the fact that factoring, DLP and ECDLP could be easily attacked with quantum computers
* Lattice-based cryptographic constructions clutch great assurance for post-quantum cryptography.
* Studies show that quantum factoring algorithms cannot crack lattice problems
* LWE : Learning with Errors
* Using “ring-based” variant of LWE reduced the key size to 2–5 kilobits and proved that the cryptosystems are faster on modern hardware.
* Peikert proved the hardness of inverting the LWE function even when the error vectors have very small entries
* Joppe demonstrated usage of post-quantum key exchange based on Ring-Learning with Errors (R-LWE) by constructing cipher suites for the Transport Layer Security (TLS).
* Their cryptographically secure implementation, aimed at the 128-bit security level, revealed that the performance price when switching from the non- quantum-safe key exchange is not too high and provided forward secrecy against future quantum attackers.

Improved design of Bluetooth pairing protocol: Classiquantum Resistance Secure Simple Pairing (CRSSP)

1. Working of CRSSP

* Simulates numeric comparison mode with five phases of pairing named as
  + Phase 1 - Input-Output capability exchange,
  + Phase 2 - R-LWE Public Key Exchange,
  + Phase 3 - Signature Generation and Verification,
  + Phase 4 - Second phase of R-LWE Public Key Exchange
  + Phase 5 - Link/Session Key generation
* Single authentication, then device verification
* As Diffie–Hellman shared secret key computed using Lattice Based Cryptography is quantum resistance, so to strengthen the security of link key, it is wrapped with two secret keys computed in phase 2 and in phase 4 using technique R-LWE.

Phases

1. Devices exchange unencrypted IO capability and Bluetooth Device Address (BDAddr)
2. Alice and Bob computes R-LWE-Diffie–Hellman (R-LWE-DH) public key and swap the public keys

*Phase 2: Public Key Exchange 1*

*SecretKey1 = f1 ( RLWEPrivateKey1 , RLWEPublicKeyAlice1 , RLWEPublicKeyBob1 , RLWEMaskingBits1 )*

*Phase 4 : Second phase of Public Key Exchange*

*SecretKey2 = f1 ( RLWEPrivateKey2 , RLWEPublicKeyAlice2 , RLWEPublicKeyBob2 , RLWEMaskingBits2 )*

1. Both calculate R-LWE shared secret key from its own private key and others public key
2. Each device randomly selects a pseudo-random number of 128-bit and communicates the same to each other
3. Based on the secret key, nonce(random number) and public keys, pairing devices computes their digital signature

*Phase 3: Signature Verification*

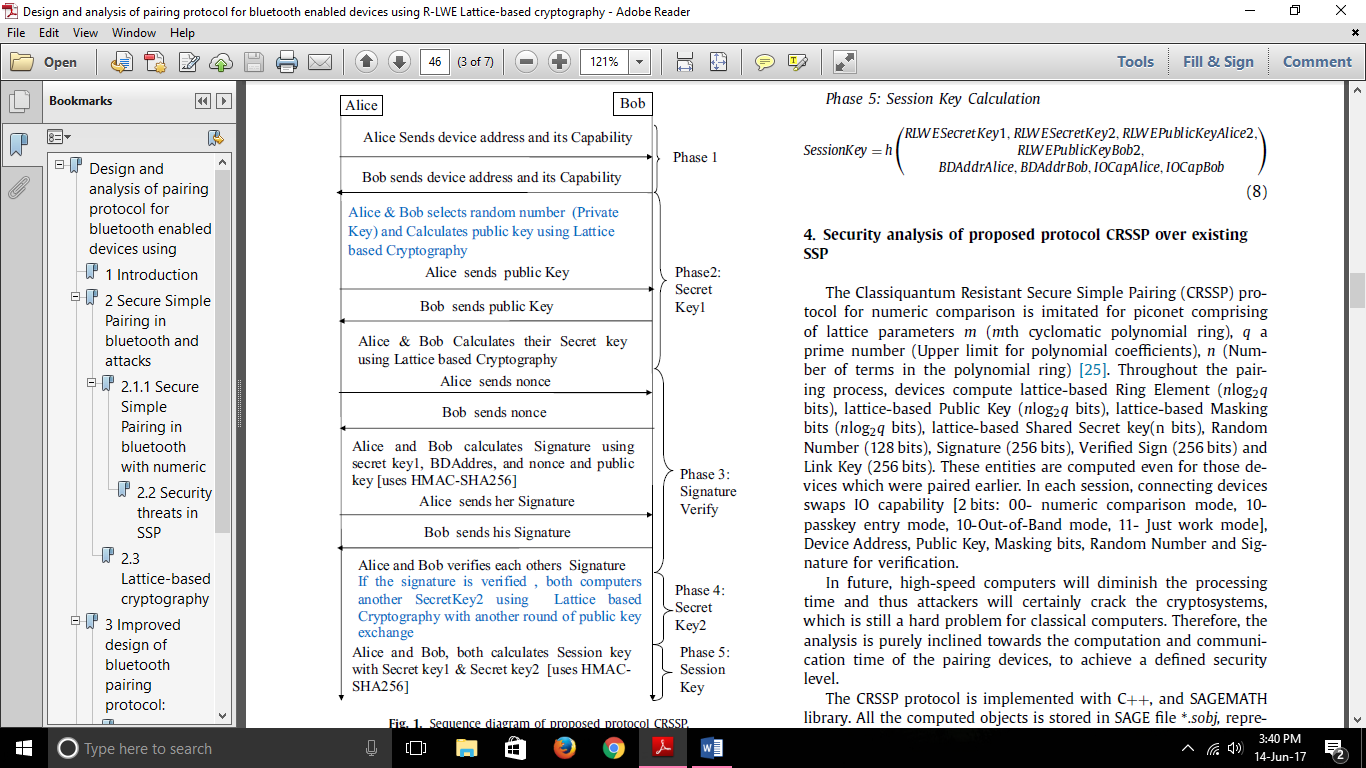
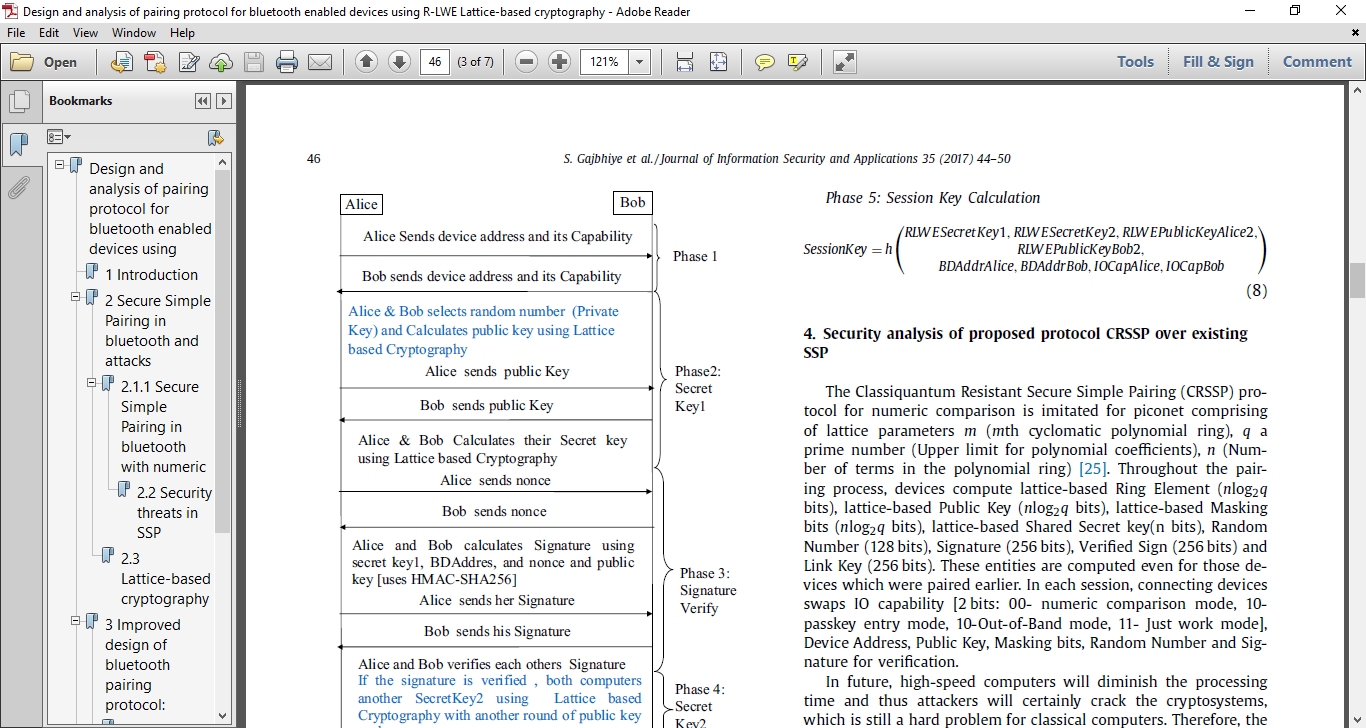
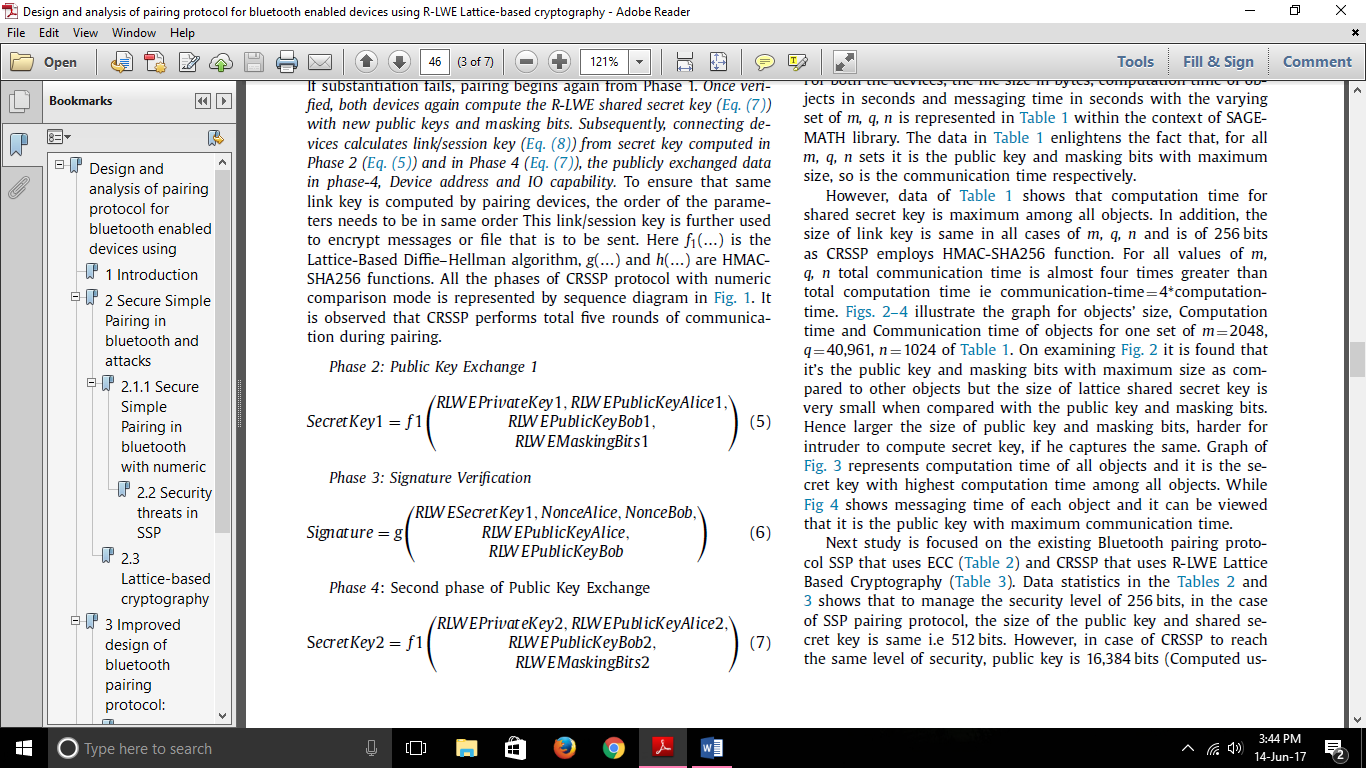
*Signature = g ( RLWESecretKey1 , NonceAlice, NonceBob, RLWEPublicKeyAlice, RLWEPublicKeyBob )*

1. Passes DS to connecting device in piconet and substantiates digital signatures of each other. The protocol proceeds further, only when signatures are verified.
2. If substantiation fails, pairing begins again from Phase 1
3. Once verified, both devices again compute the R-LWE shared secret key with new public keys and masking bits.
4. Subsequently, connecting devices calculates link/session key from secret key computed in Phase 2 and in Phase 4, the publicly exchanged data in phase-4, Device address and IO capability.

*Phase 5: Session Key Calculation*

*SessionKey = h ( RLWESecretKey1 , RLWESecretKey2 , RLWEPublicKeyAlice2 , RLWEPublicKeyBob2 , BDAddrAlice, BDAddrBob, IOCapAlice, IOCapBob )*

1. To ensure that same link key is computed by pairing devices, the order of parameters needs to be in same order
2. This link/session key is further used to encrypt messages or file that is to be sent.
3. Here f1 (…) is the Lattice-Based Diffie–Hellman algorithm, g (…) and h (…) are HMAC- SHA256 functions.



Security analysis of proposed protocol CRSSP over existing SSP

* Computation time, communication time, pairing time, size of public key in CRSSP is very large when compared to SSP.
* For the classical computers, these time and size appears to be huge.
* But for quantum computers -> increased speed,

increased memory,

decreased processing,

computation time,

therefore, it will happen in the blink of an eye.

* If MITM captures public key,
* for quantum computers -> inverse of R-LWE will be difficult equation for the attacker,
* for classical computers -> computing secret key with the inverse of ECDLP is easy.
* CRSSP resists replay attacks in conventional and quantum computers even for those devices which have been paired earlier.

*Reason -> previously generated public key, masking bits, secret key, nonce and link key are discarded immediately after being used.*

* Moreover, in each session, protocol computes new value for these objects.
* Since digital signature and its verification is realized in authentication stage of Phase 3 with challenge response mechanism, the CRSSP protocol
* resists passive eavesdropping attack
* can discover positive MITM attack.
* HMAC-SHA-256 functions and two secret keys wrap the reliability of digital signatures and link key.
* It is suggested that manufactures must change the technique used in SSP to handle the pairing in quantum computers.

Link: [sec protcols\2017 = 1-s2.0-S2214212617300911-main.pdf](sec%20protcols/2017%20=%201-s2.0-S2214212617300911-main.pdf)