Linear-Code Based Public-Key Cryptosystem

SeongHan Shin, Kazukuni Kobara
Research Center for Information Security (RCIS), AIST
LINEAR-CODE BASED PUBLIC-KEY CRYPTOSYSTEMS
Linear-Code based PKCs

• Examples
  – McEliece PKC, Niederreiter PKC
  – CFS signature

• One of the post-quantum cryptosystems
  – Shor’s alg.

• Encryption and decryption are faster
  – Especially, encryption is faster
  – Suitable for hardware implementation (xor operations in parallel)

• PK size is large
McEliece PKC [M78]

• Based on the fact
  – Generator and parity check matrices of binary Goppa codes are indistinguishable from those of random linear codes
    • Invisible structure due to a huge number of candidates for the same parameter \((n,k,t)\), a random permutation (and a secret matrix)
  – There exists an efficient decoding alg. [P75]
    • No such alg. exists for a general linear code
• Key generation \((PK=(G'=SGP,t), SK=(S,G,P))\)
  - \(G\): \((k \times n)\) generator matrix of a binary Goppa code
  - \(S\): \((k \times k)\) random binary non-singular matrix
  - \(P\): \((n \times n)\) random permutation matrix
• Encryption
  - \(C=M \cdot G' \oplus e\) where \(wt(e)=t\)
• Decryption
  - \(C \cdot P^{-1}=(M \cdot S)G \oplus e \cdot P^{-1}\)
  - \(M=(M \cdot S)S^{-1}\)
Security of McEliece PKC

- **Basic attacks**
  - Recover $G$ from $G'$ (Structural attack)
    - Secure if # of the candidate of Goppa polynomials is huge
      - E.g. $G$ should not be BCH code
    - And $n$ and $t$ are large
      - E.g., $(n,k,t)=(2048,1278,70)$
  - Recover $M$ from $C$ without learning $G$
    - General decoding problem is NP-complete [BMT78]
    - Nearest codeword problem (NCP)
    - Equivalent to Learning Parity with Noise (LPN) problem [R05]
• **OW-CPA**
  - Generalized information set decoding attack
  - Low weight codeword attack
  - Binary security workfactor for \((n,k,t)=(2048,1278,70)\) ≈ \(2^{106}\)

• **IND-CCA2**
  - With partial knowledge on target plaintexts, or decryption oracle
    • Partially-known plaintext attack
    • Related message attack
    • Reaction attack
    • Malleability attack
  - Specific conversions [KI01]
• Decryption alg. cannot be used for signatures
  – It will fail to produce any output unless its input is a vector within Hamming distance $t$ of some codeword
  – Only a very small fraction of $2^n$ possible binary vectors of length $n$ have this property
• Dual variant of McEliece PKC [LDW04]
• Encryption is faster than that of McEliece
  — Matrix operations
Niederreiter PKC

- **Key generation** ($PK=(H'=SHP,t), SK=(S,H,P))$
  - $H$: $(n-k) \times n$ parity check matrix of a binary Goppa code
  - $S$: $(n-k) \times (n-k)$ random binary non-singular matrix
  - $P$: $(n \times n)$ random permutation matrix

- **Encryption**
  - $C=H' \cdot M^T$ where $wt(M)=t$

- **Decryption**
  - Find $Z$ s.t. $H \cdot Z^T = S^{-1} \cdot C$
  - $M \cdot P^T$ by decoding alg.
  - $M=(M \cdot P^T)P$
• Complete decoding
  – Alg. to decode any syndrome (or good proportion)
  – Correct fixed additional $\delta$ errors
    • Add $\delta$ random columns from $H$ to $C$ and try to decode
    • Choose a random syndrome and try to decode
Signature: (D,M,i)

1. Hash D (to be signed) with a public hash function
2. Decrypt Hash(D,i) to get M
   - Usually, random syndrome has wt(M)>t
   - Decodable random syndrome with probability 1/9!
     - n=2^{16} and t=9 [CFS01]

Verification is straightforward

|sig|=81
- Binary security workfactor ≈ 2^{83.7}
OBLIVIOUS TRANSFER
Oblivious Transfer (OT)

- Fundamental primitive [R81]
  - Sender sends some information to receiver, but remains oblivious as to what is received
  - For secure two/multi-party computation
  - 1-out-of-2 OT [EGL82]
  - 1-out-of-n OT [EGL82]
    - Strengthened PIR (Private Information Retrieval)
    - From generic/specific computational computations

- Rabin OT (erasure channel)
  1. Sender sends \((N,e,M^e \mod N)\) to receiver
  2. Receiver sends \((X^2 \mod N)\) to sender
  3. Sender sends a square root of \(X^2\) to receiver
1-out-of-2 OT

- Sender has two messages $M_0, M_1$
- Receiver chooses a bit $b$ and gets $M_b$
- Sender’s privacy
  - Receiver does not get $M_{1-b}$
- Receiver’s privacy
  - Sender does not know $b$
- Example
  1. Sender sends $(N, e, X_0, X_1)$ to receiver
  2. Receiver sends $(K^e + X_b \mod N)$ to sender
  3. Sender sends $(M_0 + K_0, M_1 + K_1)$ to receiver
1-out-of-2 OT [C87]
- From Rabin OT and hash function H
- Sender has two messages $M_0, M_1$
- Receiver chooses a bit $b$ and gets $M_b$
1. Sender sends $(R_1, R_2, ..., R_n)$ to receiver by Rabin OT
   - With erasure (receiving) probability $Q$ ($P$)
2. Receiver gets $R_i$, for $k \leq i \leq 2k-1$, and sends two disjoint sets $I, J$ of $k$ indices to sender
   - $k < Pn = (1-Q)n < 2k < n$
3. Sender sends $(C_0 = M_0 + H((R_i)_{i \in I}), C_1 = M_1 + H((R_i)_{i \in J}))$ to receiver
1-out-of-2 bit OT [DGQN08]

- Passively secure OT
  1. Sender sends a random matrix $Q$ to receiver
  2. Receiver sends $(G'_c, t)$ to Sender where $G'_c$ is either $G'$ or $G' \oplus Q$
  3. Sender sends $(C_0, C_1, R_0, R_1, B_0, B_1)$ to receiver where $C_0, C_1$ are encryptions with $G'$ and $G' \oplus Q$, respectively, and $B_0, B_1$ are $B_0 = b_0 \oplus <M_0, R_0>$ and $B_1 = b_1 \oplus <M_1, R_1>$, respectively

- Sender’s privacy: computationally secure
- Receiver’s privacy: unconditionally secure
McEliece-based OT

- Secure OT against malicious receiver
  - Random OT
    1. Receiver commits to $G0'_{c0}$ and $G1'_{c1}$ where $c0$ and $c1$ are randomly chosen bits
    2. Sender sends random matrices $(Q_0, Q_1)$ to receiver
    3. Receiver sends $(G0'_{0}, G1'_{0}, t)$ to Sender where $G0'_{1-c0} = G0'_{c0} \oplus Q_0$ and $G1'_{1-c1} = G1'_{c1} \oplus Q_1$
    4. Sender sends challenge $j$ (0/1) to receiver where sender computes $G0'_{1} = G0'_{0} \oplus Q_0$ and $G1'_{1} = G1'_{0} \oplus Q_1$
    5. Receiver opens commitment to $G(1-j)'_{c(1-j)}$
    6. Sender sends $(C_0, C_1, R_0, R_1, B_0, B_1)$ to receiver where $C_0, C_1$ are encryptions with $Gj'_{0}$ and $Gj'_{1}$, respectively, and $B_0, B_1$ are $B_0 = b_0 \oplus <M_0, R_0>$ and $B_1 = b_1 \oplus <M_1, R_1>$, respectively
  - Malicious receiver gets both bits with $\frac{1}{2} + \epsilon$
Secure OT against malicious receiver

1. Sender and receiver run random OT where the former has \((b_0, b_1)\) and the latter has \((d = c_j, b_d)\)
2. Receiver sends \(e = c \oplus d\) to sender where \(c\) is a random bit
3. Sender sends \((f_0, f_1)\) to receiver where \(f_0 = a_0 \oplus b_e\) and \(f_1 = a_1 \oplus b_e \oplus 1\), and \((a_0, a_1)\) are random bits
4. Receiver computes \(a_c = f_c \oplus b_d\)
Secure OT against malicious receiver

- \( \Pr[\text{malicious receiver}] \)
  1. Sender chooses \((a_0, a_1)\) s.t. \(a_0 = a_{0,1} \oplus a_{0,2} \oplus \ldots \oplus a_{0,s}\) and \(a_1 = a_{1,1} \oplus a_{1,2} \oplus \ldots \oplus a_{1,s}\), where all are random bits and \(s\) is security parameter
  2. Receiver chooses a random bit \(c\)
  3. Sender and receiver run OT \(s\) times, with inputs \((a_{0,i}, a_{1,i})\) of the former and \(c_i = c\) of the latter, for \(i = 1, \ldots, s\)
  4. Receiver computes \(a_c = a_{c,1} \oplus a_{c,2} \oplus \ldots \oplus a_{c,s}\)

- Malicious receiver gets both bits with \((\frac{3}{4})^s\)
McEliece-based OT

- Other constructions [KMO08]
  - Rabin string OT
    - McEliece PKC
    - ZKID (Zero-Knowledge Identification) protocols
    - Commitment schemes
  - 1-out-of-2 string OT
    - Generalization
  - Semi-honest receiver
  - Receiver’s privacy
    - Computationally secure
Open Problem

• Simple 1-out-of-n OT
  1. Sender sends \((G', t)\) to Receiver
  2. Receiver sends \(C_i = R \cdot G' \oplus e \oplus H(i)\) to sender where \(wt(e) = t\)
  3. Sender sends \((H(R_1) \oplus M_1, H(R_2) \oplus M_2, ..., H(R_n) \oplus M_n)\) to receiver
     – It might not work!
• Prove
     – For all \(i\), there is only one codeword which is efficiently decodable in \(C_i \oplus H(i)\) and its exhaustively-searchable range
References


References


References


THANK YOU FOR YOUR ATTENTION!