ASEAN IVO

Title :

Handshake-Free Secure Device-to-Device Communications for Disaster Relief Operations

Full name of Speaker :

Sye Loong Keoh (UGS)
Raymond Ching Bon Chan (SIT)

Institutions :

University of Glasgow (Singapore)
Singapore Institute of Technology

Background :

- ASEAN is one of the most disaster-prone regions globally.
- Frequent natural disasters include earthquakes / tsunamis, floods and landslides, forest fire and haze.
- Over 200 million people affected in the past decade.
- Rapid urbanization and climate change increase vulnerability.

Source: ASEAN Disaster Information Network

Background :

- Disasters often disable traditional communication infrastructures.
- Lack of connectivity impedes search and rescue operations.
- Survivors trapped in rubble rely on timely rescue operations.
- Strong need for direct Device-to-Device (D2D) communication that can operate without network infrastructure.

- **5G New Radio (NR) Sidelink** enables D2D communication without requiring base stations.
- It provides ultra-reliable low-latency and high data rates.
- Secure D2D communication is challenging.

Source: ChatGPT and copilot generated images

Targets:

- Secure D2D Communication in a disaster-relief operation is challenging:
  - No Internet access.
  - No central key management service.
  - Devices and equipment belong to different administrative domains.
  - Handshake protocols introduce delay and overhead.

- Design a security protocol to enable fast, efficient key management for:
  - Session key establishment.
  - Session key renewal (Rekey).
  - Authenticated and encrypted communication session.
  - Supports D2D and Broadcast communications.
- Integrate with 5G NR Sidelink hardware.

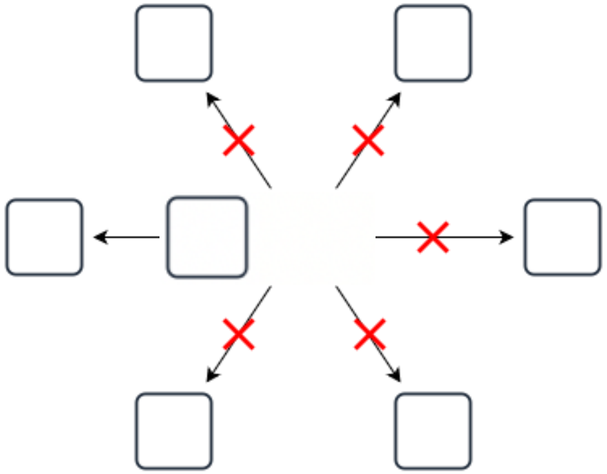
Source: ChatGPT and copilot generated images

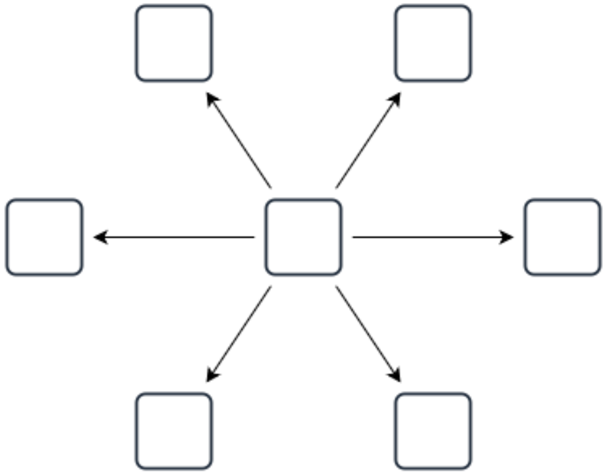
Proposed Method: Keychain-based re-Keying Function (KKF Protocol)

- Generates unbounded keychain using Chameleon Hash Function.

  $$\mathbf{H}_0 \rightarrow \mathbf{H}_1 \rightarrow \mathbf{H}_2 \rightarrow \mathbf{H}_3 \rightarrow \mathbf{H}_4 \rightarrow \ldots$$

- The resulting hash key is used as the ephemeral session keys for secure communication.
- A **Pre-Shared Master Key (MK)** is required.
- Rekey or key update is efficient by advancing the keychain to the next key.
- Do not require a centralized infrastructure and expensive handshakes.

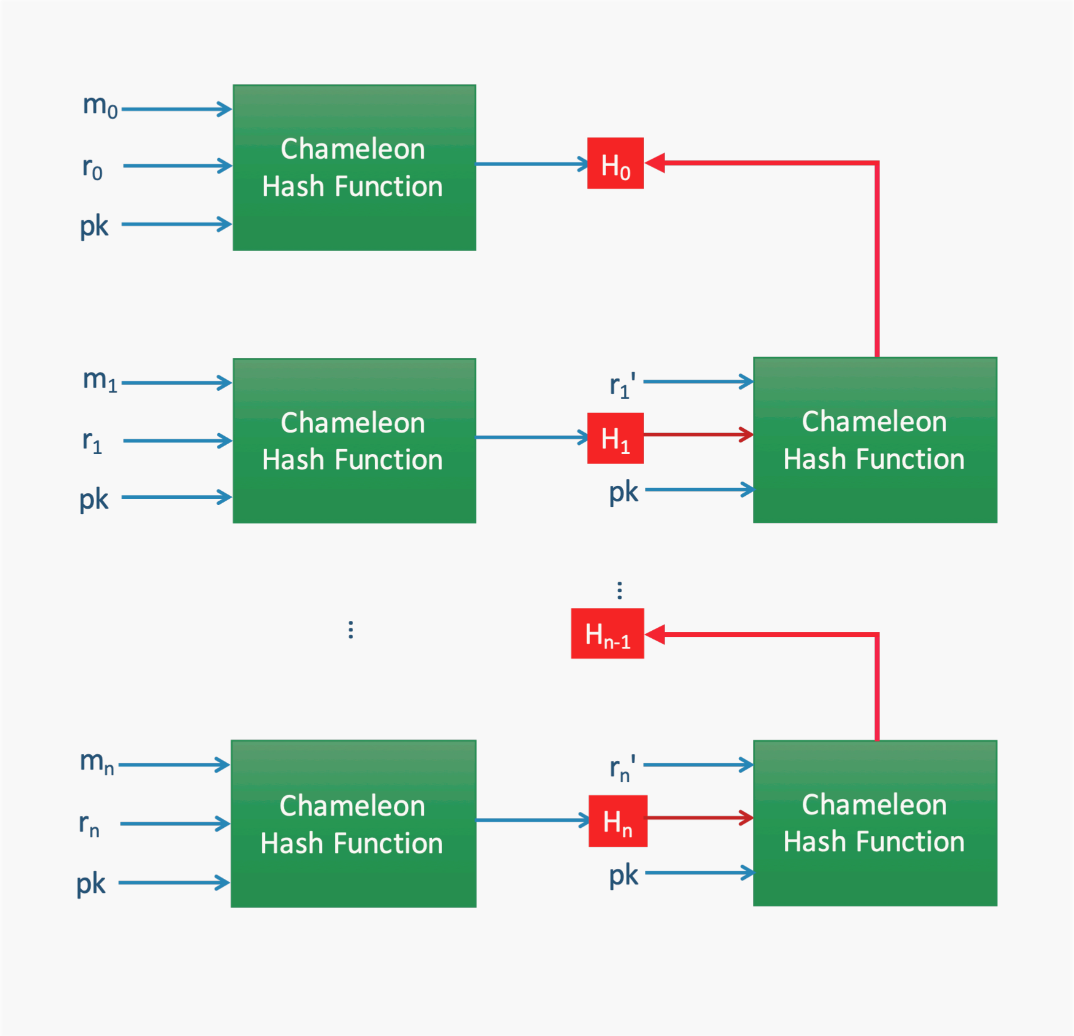




Proposed Method: Keychain-based re-Keying Function (KKF Protocol)

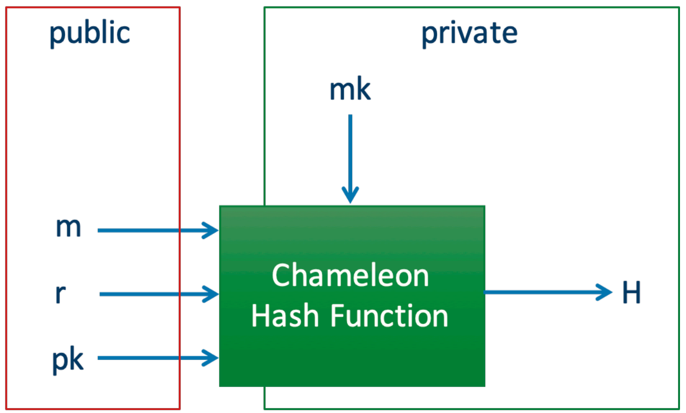
Chameleon Hash Keychain

- One-way hash function $H_n = \mathbb{CH}(m_n, r_n, pk)$
- Trapdoor function to find collision such that
  $r'_n = \mathbb{CH}(m_n, r_n, m'_n, td)$
- Verify the hash chain such that
  $\mathbb{CH}(m_n, r_n, pk) = \mathbb{CH}(H_{n+1}, r'_{n+1}, pk)$
- Series of hash keys forming a chain:
  $H_0 \rightarrow H_1 \rightarrow H_2 \rightarrow H_3 \rightarrow H_4 \rightarrow \ldots$
- Verify the sender authenticity:
  $H_{n-1} = (H_n, r'_n, pk)$

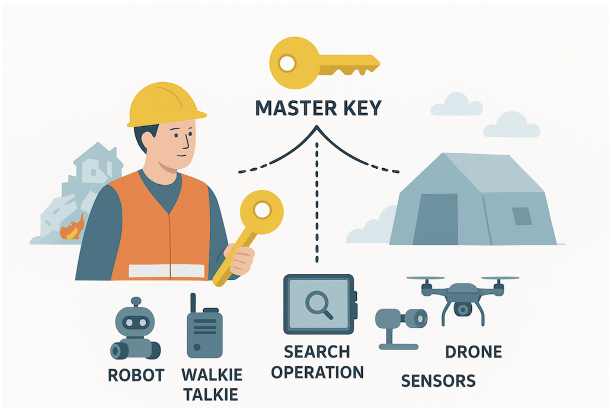


Proposed Method: KKF Security Protocols

Pre-distribution of Master Key (MK)



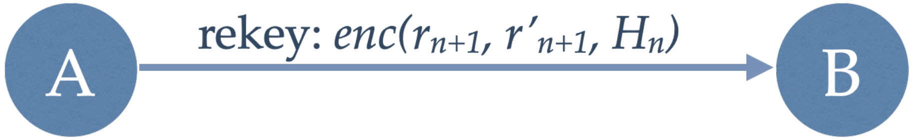
- mk: master key
- m: payload
- r: nonce
- pk: public key
- H: symmetric hash key



One-to-One Secure D2D Communication:

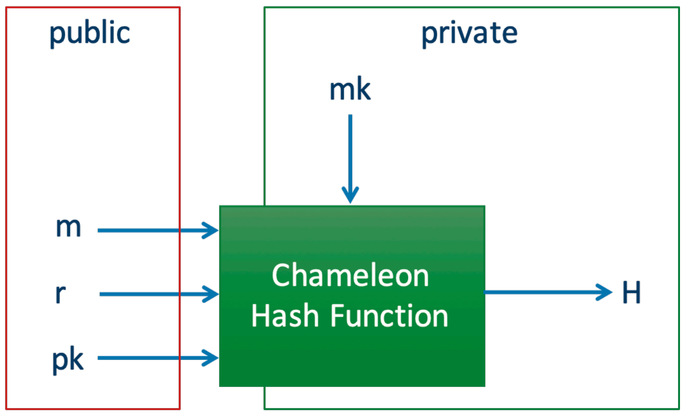
- Generate an initial key H_0 using ***m = (sid || MK)*** and a random nonce, *r.*
- Distribute *m, r* to recipient to generate H_0.
- No handshake is required.

- To rekey the session key, send the new r_{n+1} and collision nonce r'_{n+1} only.
- The recipient computes the new session key $H_1 = \mathbb{CH}(m, r_1, pk)$ and verify that $H_0 = \mathbb{CH}(H_1, r'_1, pk)$.

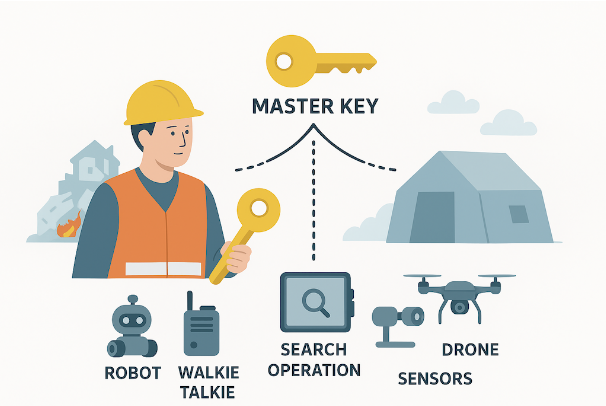


Proposed Method: KKF Security Protocols

Pre-distribution of Master Key (MK)

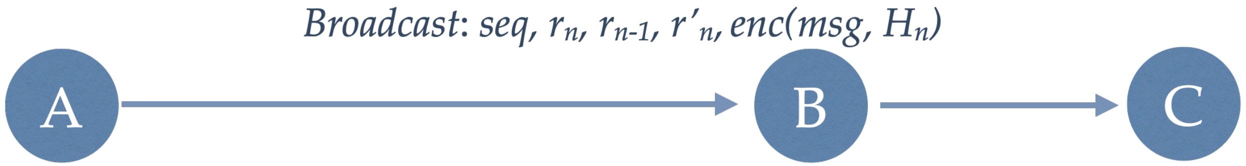


- mk: master key
- m: payload
- r: nonce
- pk: public key
- H: symmetric hash key



One-to-Many Secure Broadcast Communication:

- Every broadcast message is encrypted with a new hash key on the chain.
- Each message is encrypted with a new key.
- Sender broadcasts parameters to enable recipients to generate two consecutive hash keys. The parameters are r_n, r_{n-1} and r'_n.
- The recipient derives the latest hash key $H_n = \mathbb{CH}(m, r_n, pk)$.
- The recipient verifies the new session key's authenticity: $H_{n-1} = \mathbb{CH}(H_n, r'_n, pk)$.



Implementation: KKF Security Protocols

- The KKF security protocols were implemented in C using ECC, deployed on Raspberry Pi 4 running Bluetooth Low Energy (BLE) Generic Attribute Profile (GATT).
- Key distribution and updates took slightly longer time, approximately 1.7ms, while message protection remains relatively fast.
- Main overhead of the protocols comes from the inherent Bluetooth communication itself, i.e., round trip communication via BLE is 2s.
- The additional security protocol overhead introduced by KKF is minimal (12-13 ms).
- Integrated with 5G NR Sidelink in Matlab.

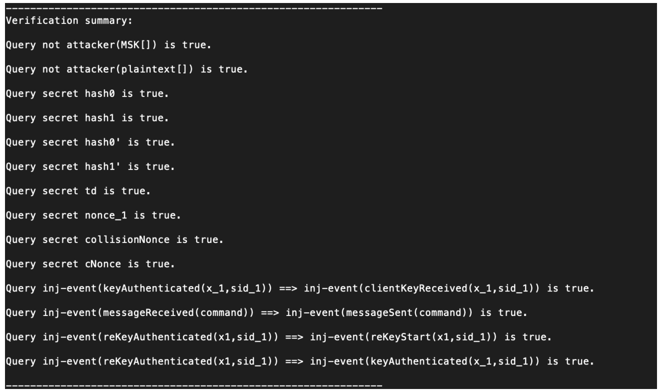
Operations	Average Time (ms)
Initial key distribution	1.713
Initial key verification	0.229
One-to-One session key update	1.723
One-to-One session key verification	1.106
One-to-One message creation	0.817
One-to-One message verification	1.358
One-to-Many message creation	0.707
One-to-Many message verification	1.134

Operations	**Average Round-trip time (s)**
One-to-One message	0.209
One-to-Many message	0.210
Plaintext message	0.197

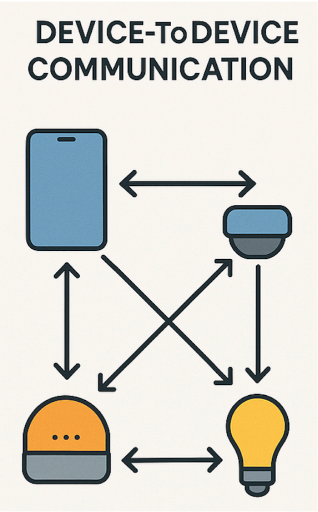
Output/Outcome: KKF Security Protocols

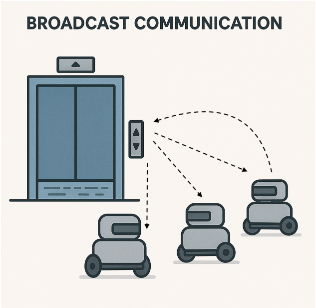
- A suite of cryptographic libraries that can be used for many applications involving authentication and secure communications:
  - Smart meter data aggregation (*Tan et. al. WF-IoT 2018*).
  - GNSS signal spoofing (*Chu et. al. IFIP ICCIP 2021).*
  - IoT security (*Wang et. al. IFIP ICCIP 2024*).

- Further conducted formal security analysis of KKF protocols using *Proverif*, ensuring the following security properties:
  - Secrecy of Master key, message, hash key (ephemeral session key), nonce and collision nonce.
  - Resistant to replay attacks of “key distribution”, “rekey” messages.
  - Ability to detect spoofing of messages.



```
--------------------------------------------------------------
Verification summary:

Query not attacker(MSK[]) is true.

Query not attacker(plaintext[]) is true.

Query secret hash0 is true.

Query secret hash1 is true.

Query secret hash0' is true.

Query secret hash1' is true.

Query secret td is true.

Query secret nonce_1 is true.

Query secret collisionNonce is true.

Query secret cNonce is true.

Query inj-event(keyAuthenticated(x_1,sid_1)) ==> inj-event(clientKeyReceived(x_1,sid_1)) is true.

Query inj-event(messageReceived(command)) ==> inj-event(messageSent(command)) is true.

Query inj-event(reKeyAuthenticated(x1,sid_1)) ==> inj-event(reKeyStart(x1,sid_1)) is true.

Query inj-event(reKeyAuthenticated(x1,sid_1)) ==> inj-event(keyAuthenticated(x_1,sid_1)) is true.

--------------------------------------------------------------
```





Conclusions and Future Works

- Proposed a handshake-free KKF security protocols for securing both D2D and broadcast communication.
  - No handshake overhead.
  - Fast and secure authentication and rekey.
  - Verified its security formally using *Proverif.*
  - Easily integrated with any communication medium.
- Next steps include implementing the KKF protocols for 5G NR sidelink actual devices.
- Work with partners in ASEAN to integrate the KKF protocols on their disaster relief operations, particularly the establishment of communication using 5G NR on drones, robots, mobile devices.
- Real-world testing with ASEAN & Japan partners.


University of Glasgow
Thank you
University of Glasgow (Singapore)